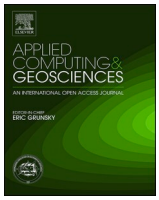
Applied Computing and Geosciences 16 (2022) 100091

Contents lists available at ScienceDirect

Applied Computing and Geosciences

journal homepage: www.sciencedirect.com/journal/applied-computing-and-geosciences



A parsimonious parametrization of the Direct Sampling algorithm for multiple-point statistical simulations

Przemysław Juda [a,1], Philippe Renard [a,b,*], Julien Straubhaar [a]

[a] Stochastic Hydrogeology and Geostatistics Group, Centre for Hydrogeology and Geothermics, University of Neuchâtel, Rue Emile-Argand 11, 2000, Neuchâtel, Switzerland
[b] Department of Geosciences, University of Oslo, Oslo, Norway

ARTICLE INFO

Keywords:
Geostatistics
Multiple-point statistics
Hydrogeology
Stochastic simulation
Direct sampling

ABSTRACT

Multiple-point statistics algorithms allow modeling spatial variability from training images. Among these techniques, the Direct Sampling (DS) algorithm has advanced capabilities, such as multivariate simulations, treatment of non-stationarity, multi-resolution capabilities, conditioning by inequality or connectivity data. However, finding the right trade-off between computing time and simulation quality requires tuning three main parameters, which can be complicated since simulation time and quality are affected by these parameters in a complex manner. To facilitate the parameter selection, we propose the Direct Sampling Best Candidate (DSBC) parametrization approach. It consists in setting the distance threshold to 0. The two other parameters are kept (the number of neighbors and the scan fraction) as well as all the advantages of DS. We present three test cases that prove that the DSBC approach allows to identify efficiently parameters leading to comparable or better quality and computational time than the standard DS parametrization. We conclude that the DSBC approach could be used as a default mode when using DS, and that the standard parametrization should only be used when the DSBC approach is not sufficient.

1. Introduction

Many different Multiple-Point Statistics (MPS) methods exist and are used to model discrete or continuous fields for a broad range of applications (Mariethoz and Caers, 2015). These methods are able to represent complex spatial or temporal patterns and use analog data (in the form of training images) to learn the patterns that should be simulated for a given problem. Among the MPS methods, the Direct Sampling (DS) is often employed since it is very flexible and computationally efficient (Mariethoz et al., 2010). DS is capable of simulating non-stationary fields, accounting for continuous maps of rotations or affinity ratios (Mariethoz and Kelly, 2011), accounting for trends expressed as secondary variables to guide the patterns in the simulation grid. It has been recently extended for the simulation of multi-resolution patterns (Straubhaar et al., 2020), or for handling complex conditioning data such as inequality constraints (Straubhaar and Renard, 2021). DS is parallelized (Mariethoz, 2010) to simulate efficiently large simulation grids. DS has been used for example for hydrogeological applications (Jäggli et al., 2018; Dall'Alba et al., 2020; Lam et al., 2020), ore reserve estimation (Dagasan et al., 2018), or geomorphological simulations (Neven et al., 2021).

As with any geostatistical algorithm, DS has several computational parameters which govern the simulation quality and computation time. The three main parameters requiring tuning are: the distance threshold (t), the maximal scan fraction (f), and the number of nearest neighbors (n). The distance threshold controls the acceptable level of similarity between patterns when searching for good patterns, the maximal scan fraction limits the search in the TI, and the number of nearest neighbors controls the size of the patterns. The quality of the simulations and the computation time depend on the choice of these parameters in a complex manner. While some rules of thumb have been proposed for choosing the parameters (Meerschman et al., 2013), the choice of an optimal set can be difficult and not intuitive for new DS users. Therefore an extensive search in parameter space is often the only solution to avoid a trial-and-error search but the systematic search can have a significant computational cost (Dagasan et al., 2018).

* Corresponding author. Stochastic Hydrogeology and Geostatistics Group, Centre for Hydrogeology and Geothermics, University of Neuchâtel, Rue Emile-Argand 11, 2000, Neuchâtel, Switzerland.
E-mail address: philippe.renard@unine.ch (P. Renard).
[1] Current address: MeteoSwiss, Via ai Monti 146, Locarno Monti, Switzerland.

https://doi.org/10.1016/j.acags.2022.100091
Received 17 January 2022; Received in revised form 11 June 2022; Accepted 21 July 2022
Available online 18 August 2022

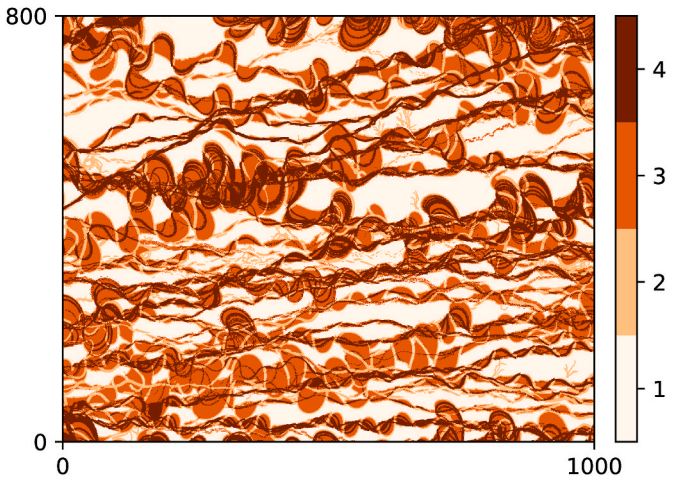


Fig. 1. Categorical fluvial training image (Jäggli et al., 2018). The dimensions are indicated in pixels, and different categories (facies) are labeled with 1, 2, 3, 4.

Table 1
DSBC parameters for Case 1. The Cartesian product of the two lists gives the complete parameter sets.

n	f
8, 16, 24, 32, 64	1/256, 1/128, 1/64, 1/32, 1/16, 1/8

Table 2
DS parameters for Case 1. The union of the Cartesian products of the two lists in each row gives all the parameter sets.

n	t	f
8	1/8, 2/8	0.25
16	1/16, 2/16, 3/16, 4/16	0.25
32	1/32, 2/32, ..., 8/32	0.25
64	1/64, 2/64, ..., 16/64	0.25

To address these points, we show in this paper that a simplified parametrization of DS could facilitate its use. The key idea is to set the distance threshold t to 0. This way to parameterize the DS algorithm will be denoted DSBC in the following of this paper, it stands for Direct Sampling Best Candidate (DSBC). Three test cases are presented to test and illustrate the DSBC approach: a categorical unconditional simulation, a continuous simulation with conditioning data and a categorical simulation with trend and orientations.

The paper is structured as follows. First, we review the Direct Sampling method and explain the DSBC special case. Second, quality metrics are introduced to compare the performance of DS and DSBC. Third, the test cases are presented. Fourth, results and comparison of the DS vs DSBC are given. Finally, we wrap up with conclusions.

2. Direct Sampling and new parametrization

This section reviews first the key elements of the Direct Sampling (DS) algorithm. More details are available in the original publication (Mariethoz et al., 2010). We then introduce the new Direct Sampling Best Candidate (DSBC) parametrization. Note that we use different notations as compared to the original paper.

The aim of DS is to simulate a random field $Z(\mathbf{x})$ on a simulation grid (SG). The different locations (pixels, nodes) on the grid will be denoted with $\mathbf{x}$. All the non-informed locations $\mathbf{x}$ in SG will be iteratively visited and simulated by the algorithm. The SG can be partially filled with conditioning data. DS uses a training image (TI) as an analog random field $Z(\mathbf{y})$ to model $Z(\mathbf{x})$. The TI is one of the input parameters. We denote by $\mathbf{y}$ the nodes of the TI. The conditioning data are a set of pairs $\{(\mathbf{x}_1^{HD}, z_1), (\mathbf{x}_2^{HD}, z_2), \ldots, (\mathbf{x}_{N_{HD}}^{HD}, z_{N_{HD}})\}$, where $\mathbf{x}_i^{HD}$ denotes the position of the ith conditioning data, z_i its corresponding value, and N_{HD} is the number of conditioning data points.

After assigning the conditioning data to SG, the remaining points are ordered $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where $N = N_{SG} - N_{HD}$, and N_{SG} is the number of nodes in the simulation grid. This ordering is called a simulation path, and it is usually random. If no conditioning was used, the value of the first simulated point in SG is taken randomly from the TI: $Z(\mathbf{x}_1) = Z(\mathbf{y})$, where $\mathbf{y}$ is a random point in the TI.

DS fills the SG node by node. Let i be the index of the next node to be filled. First, the n closest neighbors of $\mathbf{x}_i$ are found. The neighbors are the nodes which are informed, already simulated or in the conditioning data set. In the early iterations (i small), it might happen that fewer than n neighbors can be found. Then, all of them are considered, and for simplicity the number of neighbors is still noted n in this situation and they are denoted with:

$$\mathcal{X}_i = \{\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^n\}. \tag{1}$$

Second, the set $\mathcal{L}_i$ of lag vectors is found:

$$\mathcal{L}_i = \left\{\mathbf{h}_j : \mathbf{h}_j = \mathbf{x}_i^j - \mathbf{x}_i, \quad j = 1, \ldots, n\right\}. \tag{2}$$

Before we proceed to the third step, we need definitions of the neighborhood, data event, the search window, and the distance function. The neighborhood $\mathcal{N}$ of the node $\mathbf{x}$ given the lag vectors $\mathcal{L}$ is defined by:

$$\mathcal{N}(\mathbf{x}; \mathcal{L}) = \{\mathbf{x} + \mathbf{h}_1, \mathbf{x} + \mathbf{h}_2, \ldots \mathbf{x} + \mathbf{h}_n\}, \tag{3}$$

and data event at the node $\mathbf{x}$ given the neighborhood $\mathcal{L}$:

$$\mathcal{D}(\mathbf{x}; \mathcal{L}) = \{Z(\mathbf{x} + \mathbf{h}_1), Z(\mathbf{x} + \mathbf{h}_2), \ldots, Z(\mathbf{x} + \mathbf{h}_n)\}. \tag{4}$$

The search window $\mathcal{Y}$ is the set of points in the TI, which can be visited to look for the matching pattern:

$$\mathcal{Y}(\mathcal{L}) = \{\mathbf{y} \in TI : \mathcal{N}(\mathbf{y}, \mathcal{L}) \subset TI\}. \tag{5}$$

The distance between two data events can be defined in various manners. Different metrics are used for categorical and continuous variables. The distance only makes sense, when the same lag vectors are used. The default categorical distance reads:

$$d(\mathcal{D}(\mathbf{x}), \mathcal{D}(\mathbf{y}); \mathcal{L}) = d(\mathcal{D}(\mathbf{x}; \mathcal{L}), \mathcal{D}(\mathbf{y}; \mathcal{L})) = \frac{1}{n} \sum_{i=1}^{n} \left[1 - \mathbb{1}_{Z(\mathbf{x}+\mathbf{h}_i)}(Z(\mathbf{y} + \mathbf{h}_i))\right] \tag{6}$$

with the indicator variable: $\mathbb{1}_x(y) = 1$ if $x = y$ and 0 otherwise. The default continuous distance reads:

$$d(\mathcal{D}(\mathbf{x}), \mathcal{D}(\mathbf{y}); \mathcal{L}) = \frac{1}{n} \sum_{i=1}^{n} \sqrt{\frac{[Z(\mathbf{x} + \mathbf{h}_i) - (Z(\mathbf{y} + \mathbf{h}_i))]^2}{d_{max}^2}}, \tag{7}$$

with $d_{\max} = \max_{\mathbf{y} \in TI} Z(y) - \min_{\mathbf{y} \in TI} Z(y)$.

The third step of the algorithm consists in finding the search window $\mathcal{Y}(\mathcal{L}_i)$, traversing it (again using a random path) and testing the consecutive points $\mathbf{y}$ by computing the distance $d(\mathcal{D}(\mathbf{x}_i), \mathcal{D}(\mathbf{y}); \mathcal{L}_i)$. If the distance is lower than the predefined threshold t, then we set $Z(\mathbf{x}_i) = Z(\mathbf{y})$, otherwise a new point is tested. This step is repeated until a point with a sufficiently small distance is found or the maximal number of points (fN_{TI}) have been visited (N_{TI} stands for the number of nodes in the TI). In the latter case, the $Z(\mathbf{y})$ value corresponding to $\mathbf{y}$ yielding the smallest distance is used.

The distance threshold parameter t allows stopping immediately the search if a suitable candidate has been found. However, it also introduces a slight inconsistency, as it is not guaranteed that such a point exists. Therefore, a second condition is applied in the DS algorithm:

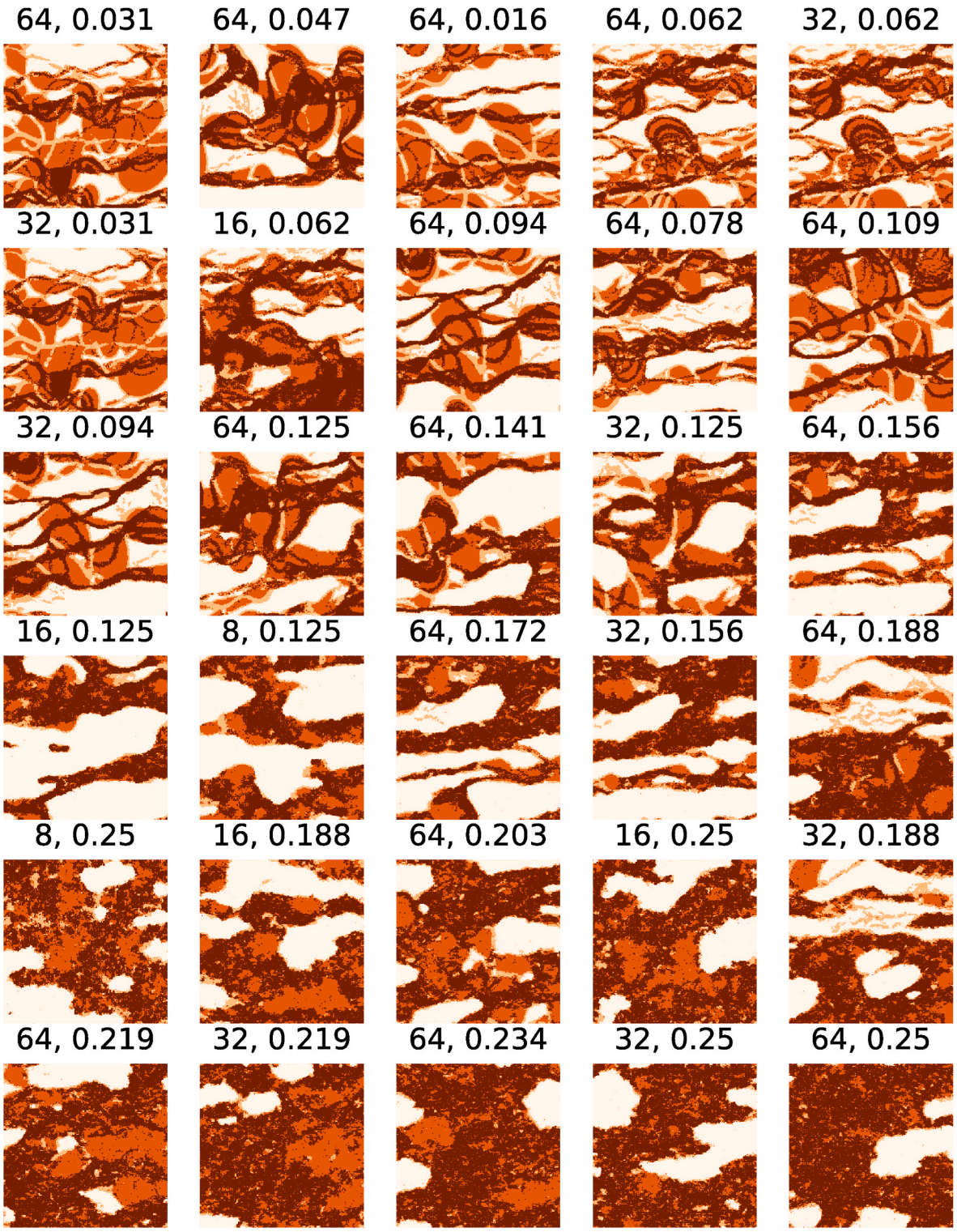


Fig. 2. Example DS realizations ordered by descending score (increasing error). Figure titles show n and t. For the color scale refer to Fig. 1.

when the maximal number of nodes have been scanned and no candidate satisfying the threshold has been found, the algorithm selects the best candidate found so far.

The proposed DSBC parametrization simplifies the workflow: the parameter t is set to 0. Only two parameters are kept: n and f. The parameter n has the same role in DS and DSBC. But the maximal scan fraction f becomes more important. When scanning the TI for a pattern, the algorithm always scans a fixed portion f and accepts the pattern with the smallest distance among all visited pixels (unless a perfect match is found — then it can be immediately accepted for saving computation time). The scan fraction f takes over a part of the role of t in the standard application of DS. When f is decreased, the scan is stopped earlier, and often an approximate pattern is accepted. When f is increased, a larger portion of TI is scanned in search of a better match, and then it is likely that a nearly perfect match is accepted. Therefore, increasing f with the DSBC approach corresponds to decreasing t in DS, but f has the

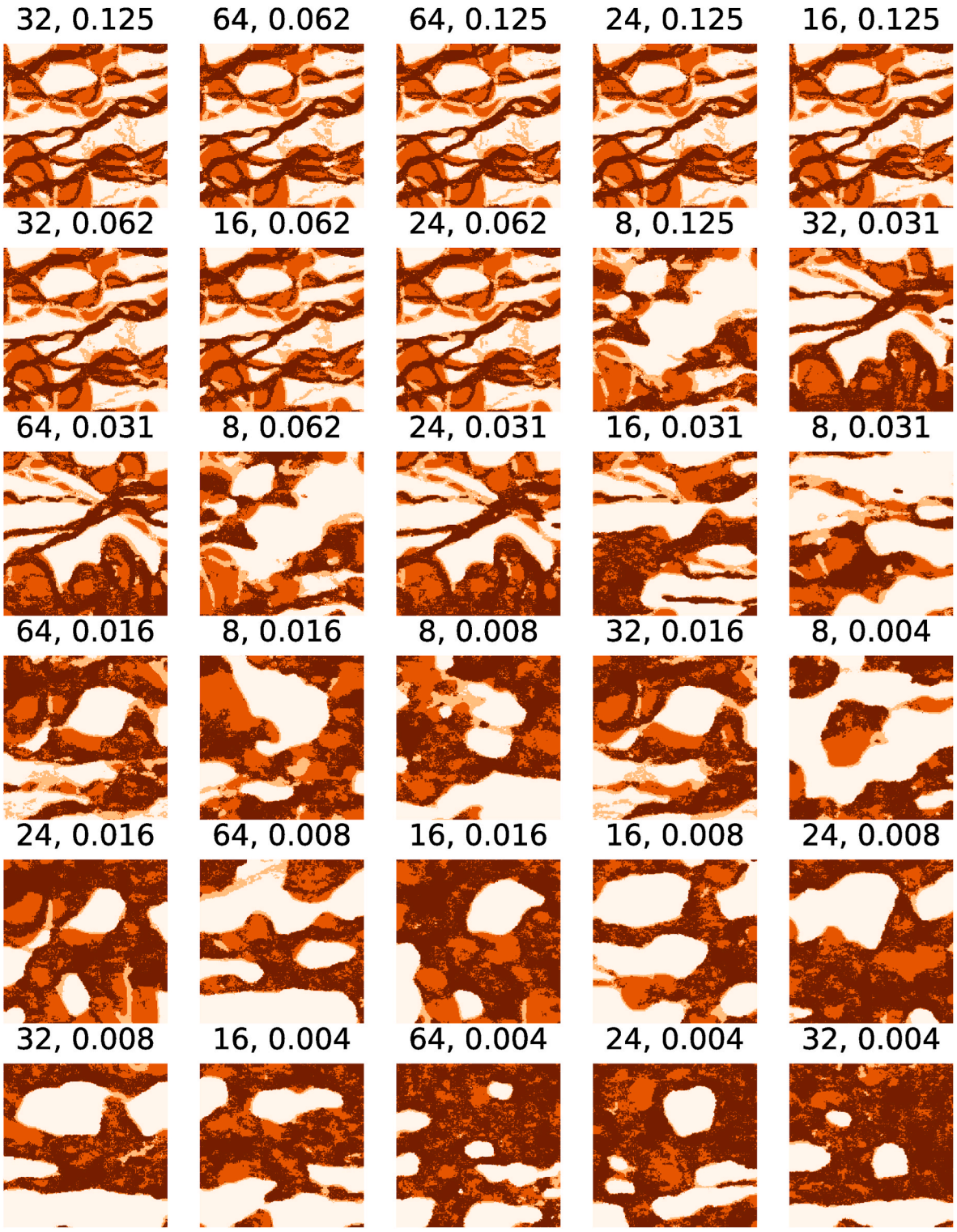


Fig. 3. Example DSBC realizations ordered by descending score (increasing error). Figure titles show n and f. For the color scale refer to Fig. 1.

advantage of producing a more intuitive response of the algorithm than t. For example, when choosing t in DS, one should note that in case of categorical simulations and standard distance function (6), distance takes discrete values and changing t might not change the algorithm behavior. In such a case, setting t to whatever value below $1/n$ would be equivalent to setting it to 0.

3. Quality metrics

To compare the quality of simulations obtained with different parameter sets, we used the following indicators.

The connectivity function $\tau_s(\mathbf{h})$ for category s is defined as the probability that two points of the same category s, away from each other by distance $\mathbf{h}$ are connected (in the sense of belonging to the same connected component):

Table 3
5 best parameter sets according to the total error for DS and DSBC.

	n	t	f	time (s)	ϵ_c	ϵ_v	ϵ_c	ϵ_v	$\epsilon_c + \epsilon_v$
DS	64	1/32	0.25	40	0.032	0.012	0.186	0.094	0.279
DS	64	3/64	0.25	37	0.033	0.012	0.194	0.092	0.285
DS	64	1/64	0.25	43	0.036	0.010	0.211	0.077	0.289
DS	64	1/16	0.25	33	0.043	0.009	0.252	0.069	0.321
DS	32	1/16	0.25	17	0.043	0.009	0.251	0.072	0.323
DSBC	32	0	1/8	26	0.034	0.019	0.200	0.146	0.347
DSBC	64	0	1/16	30	0.041	0.020	0.241	0.156	0.397
DSBC	64	0	1/8	44	0.047	0.018	0.271	0.139	0.410
DSBC	24	0	1/8	23	0.046	0.019	0.266	0.153	0.419
DSBC	16	0	1/8	18	0.044	0.021	0.257	0.166	0.423

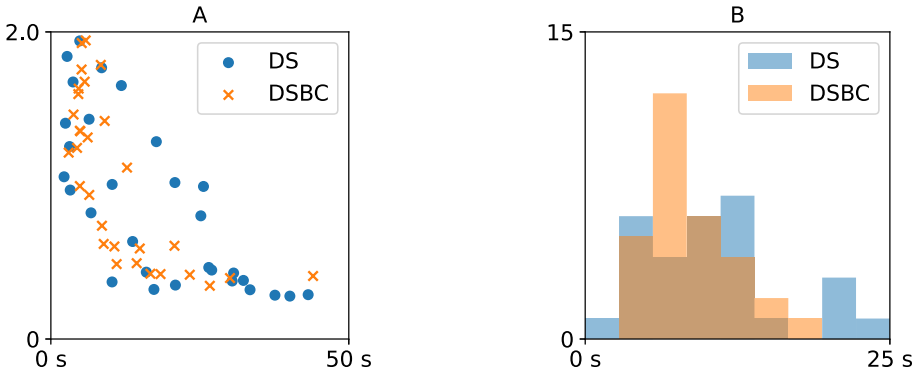


Fig. 4. Simulation time of a single realization versus the total error ϵ_t (A). Histogram of *time* $\times$ ϵ_t (B).

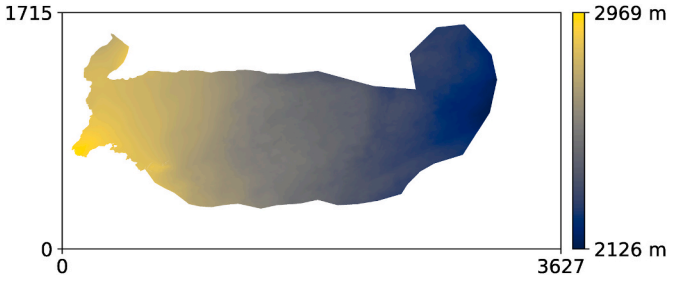


Fig. 5. Continuous topography training image (Neven et al., 2021). The dimensions are indicated in pixels, and elevation in m.

Table 4
DS and DSBC parameters for Case 2. The Cartesian product of the two lists for each algorithm gives all the parameter sets.

(A) DS parameters		
n	t	f
8, 16, 32, 64	0.001, 0.002, 0.005, 0.01	0.005
(B) DSBC parameters		
n	f	
8, 16, 32, 64	0.001, 0.0005, 0.0002, 0.0001	

$$\tau_s(\mathbf{h}) = \text{Prob}\{\mathbf{x} \Longleftrightarrow \mathbf{x}+\mathbf{h} : \text{category}(\mathbf{x}) = s,\ \text{category}(\mathbf{x}+\mathbf{h}) = s\} \tag{8}$$

where CATEGORY is a function returning category (facies) of the argument. The indicator variogram is a variogram of the indicator variable and is computed for each category s. It represents half the probability that for two points separated by $\mathbf{h}$, one belongs to the category s and the other does not:

$$\gamma_s(\mathbf{h}) = \frac{1}{2}\mathbb{E}\left[(\mathbb{1}_s(\mathbf{x}+\mathbf{h}) - \mathbb{1}_s(\mathbf{x}))^2\right] = \frac{1}{2}\text{Prob}\{\mathbb{1}_s(\mathbf{x}+\mathbf{h}) \neq \mathbb{1}_s(\mathbf{x})\}. \tag{9}$$

For the sake of simplicity, in our setting where channels go from left to right, we consider only the x direction (*i.e.* $\mathbf{h}$ parallel to x axis) for the connectivity functions and indicator variograms.

The statistics of the ensemble of simulations are usually compared with the TI as the reference, for example Meerschman et al. (2013) used connectivity functions and indicator variograms to compare how DS performed with different parameter sets in unconditional simulation case. However, due to grid size effects, the statistics of the TI differ from the statistics derived from sub-images of different sizes extracted from the TI. Therefore, we propose to compare two ensembles: a set of sub-images of the TI and a set of simulations. The comparison is achieved using the Wasserstein distance (or earth mover distance). The first Wasserstein distance between two probability distributions u and v can be computed as:

$$l_1(u,v) = \int_{-\infty}^{+\infty} |U(x) - V(x)|dx, \tag{10}$$

where U and V are the cumulative distribution functions (CDF) of u and v respectively.

If the true category values are known, it is possible to use scoring rules to assess the probabilistic forecasts (Gneiting et al., 2007). For example, when conditioning data are available, a cross-validation approach can be used (Juda et al., 2020) and quadratic score applied:

$$S(\mathbf{p}, i) = -\sum_{j=1}^{M} (\delta_{ij} - p_j)^2, \tag{11}$$

where $\mathbf{p}$ is the predictive probability vector, i is the true category and δ_{ij} is the Kronecker symbol. The cross-validation score is a mean score over multiple cross-validation runs.

If true values of a continuous variable are known, continuous ranked probability score (CRPS) can be used (Gneiting et al., 2007; Gneiting and

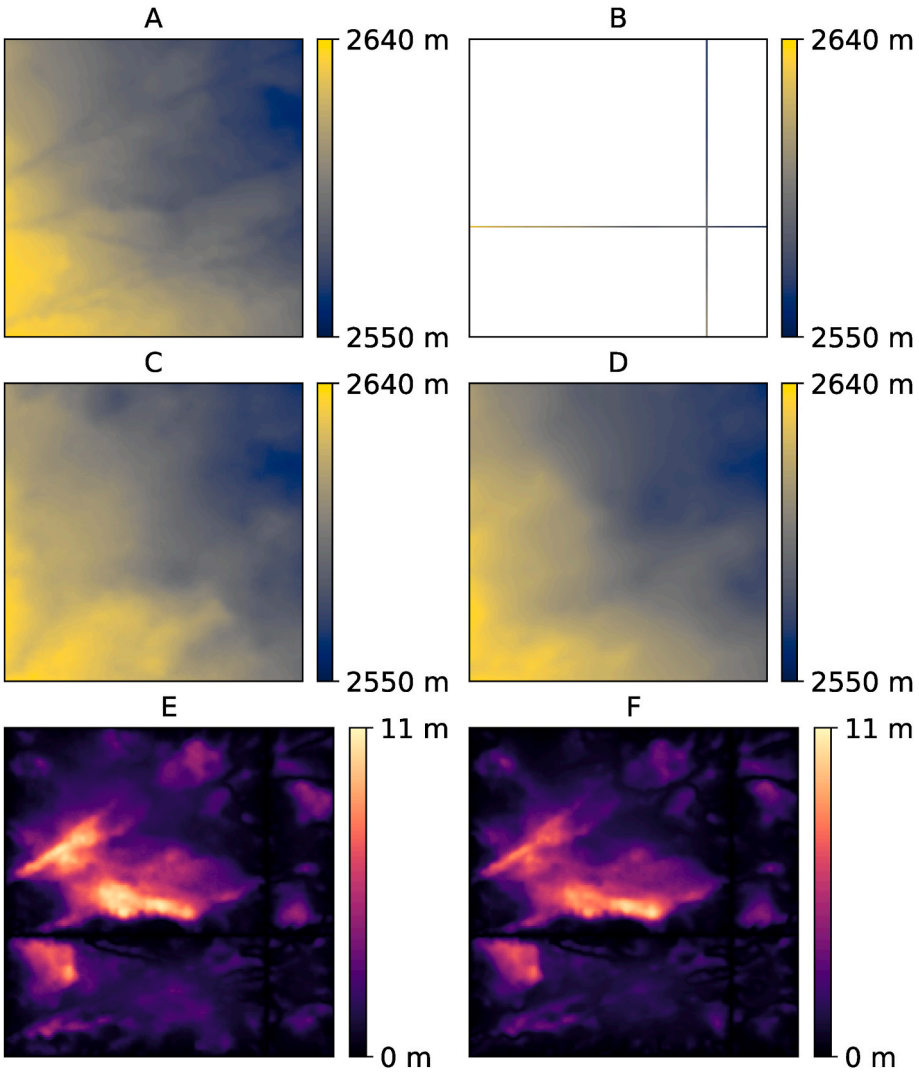


Fig. 6. An example area to be cut out (A), corresponding conditioning data (B), example simulation by DS (C), example simulation by DSBC (D), pixelwise CRPS score map for DS ensemble (E), pixelwise CRPS score for DSBC ensemble (F).

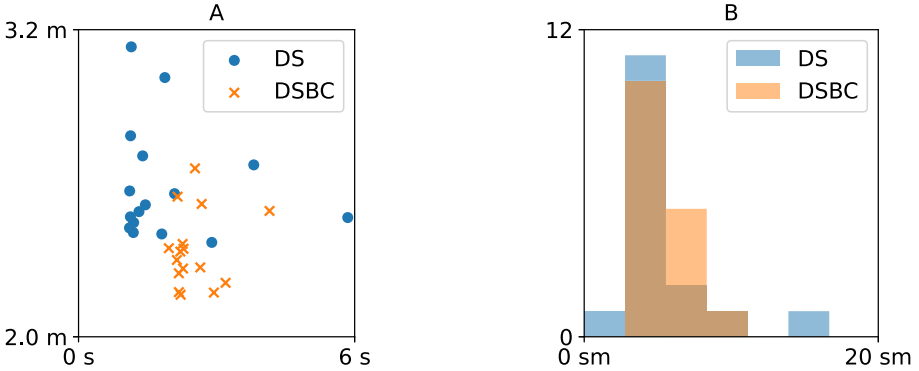


Fig. 7. Simulation time for Case 2 of a single realization versus the total error $MVE + MPE$ (A). Histogram of $time \times (MVE + MPE)$ (B).

Raftery, 2007):

$$CRPS(F,x) = \int_{-\infty}^{+\infty} (F(y) - \mathbb{1}(y > x))^2 dy, \tag{12}$$

where $F(y)$ is the CDF of predictive distribution and x the true value.

4. Numerical experiments and results

Three test cases are presented. They are designed to test the DSBC parametrization in three situations frequently encountered when using DS. The first test case is a categorical unconditional simulation study of a fluvial system. The second test case is a conditional continuous simulation of a topography with non-stationarities. And finally, the third case

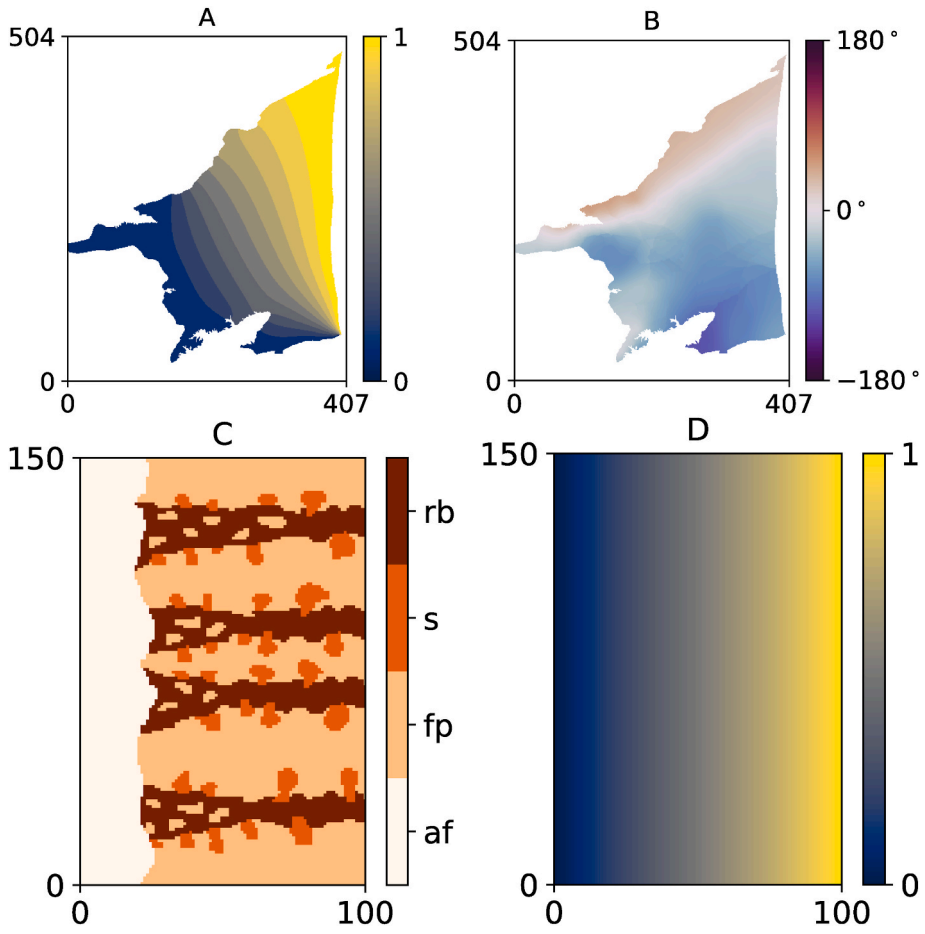


Fig. 8. Simulation area with defined trend (A), orientation (B), and training image (C) with its trend (D).

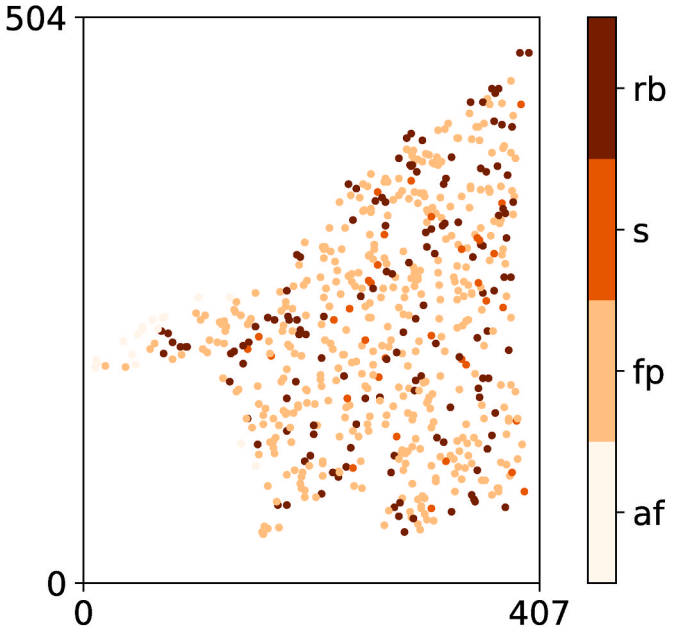


Fig. 9. Conditioning data for the Roussillon case.

Table 5
DS parameters for Case 3. The union of the Cartesian products of the two lists in each row gives all the parameter sets.

n	*t*	*f*
8	1/8, 2/8	0.1, 0.2, 0.4, 0.8
16	2/16, 3/16, 4/16	.1, 0.2, 0.4, 0.8
32	1/16, 2/16, 3/16, 4/16	.1, 0.2, 0.4, 0.8
64	1/32, 1/16, 2/16, 3/16, 4/16	.1, 0.2, 0.4, 0.8

Table 6
DSBC parameters for Case 3. The Cartesian product of the two lists gives all the parameter sets.

n	*f*
8, 16, 32, 64	.001, .002, .004, .006, .008, .01, .02, .04, .06, .08, .1, .2, .4, .6

Table 7
5 best DS parameter sets according to *CV* score.

n	*t*	*f*	time (s)	*CV*-score	*CV*-σ
32	0.06251	0.1	4.6	−0.31	0.05
32	0.06251	0.2	4.5	−0.32	0.03
32	0.06251	0.8	7.4	−0.32	0.03
64	0.03126	0.1	4.8	−0.33	0.03
32	0.06251	0.4	4.3	−0.33	0.03

Table 8
5 best DSBC parameter sets according to *CV* score.

n	*f*	time (s)	*CV*-score	*CV*-σ
16	0.02	4.3	−0.29	0.04
16	0.06	4.3	−0.29	0.03
16	0.04	4.2	−0.29	0.03
16	0.20	4.1	−0.29	0.03
16	0.08	4.2	−0.29	0.03

is a categorical conditional simulation of an alluvial plain with trends and rotations.

4.1. Test case 1. Categorical unconditional with pyramids

In this test case, the TI represents a fluvial system with four categories, its size is 800 × 1000 pixels (Fig. 1). It was first presented by Jäggli et al. (2018), who used it in an inversion set-up. The aim is to test how the standard DS and DSBC parametrizations influence the quality of the simulations. For each simulation method, 30 parameter configurations are considered. The DSBC parameter sets were obtained by applying the Cartesian product of the list of *n* and *f* values (Table 1). The DS parameter sets were obtained as union of the Cartesian products corresponding to the different rows of Table 2. The maximal scan fraction *f* equals to 0.25 for all simulations. The reason for setting individual ranges for different *n* values is that below a certain threshold value, diminishing further this parameter has no effect, as it corresponds to setting threshold to 0, and thus running DS in DSBC mode.

For each parameter set, an ensemble of 40 realizations is generated. The simulation grid has a size of 200 by 200 pixels. The realizations are unconditional. The multi-resolution mode of DS (Straubhaar et al., 2020) is applied with 2 levels and a reduction factor of 2 in both directions in each level. The connectivity functions and the indicator variograms are computed only in *x* directions. Let *h* be the lag distance, and *M* the total number of categories. The connectivity error is given by:

$$\varepsilon_c = \frac{1}{M}\frac{1}{n_x}\sum_{s=1}^{M}\sum_{h=1}^{n_x} l_1(u_s^\tau(x;h), v_s^\tau(x;h)), \tag{13}$$

with $u_s^\tau(x;h)$ is the distribution of values $\tau_s(h)$ deduced from the simulated ensemble, and $v_s^\tau(x;h)$ the distribution deduced from the ensemble of images of the same size $n_x \times n_y$ but extracted from the TI. The indicator variogram error is computed in the same manner, but the indicator variograms γ_s are considered in place of τ_s:

$$\varepsilon_v = \frac{1}{M}\frac{1}{n_x}\sum_{s=1}^{M}\sum_{h=1}^{n_x} l_1(u_s^\gamma(x;h), v_s^\gamma(x;h)). \tag{14}$$

These errors are then normalized: $\epsilon_c = \varepsilon_c/\max\varepsilon_c$, $\epsilon_v = \varepsilon_v/\max\varepsilon_v$, and the total error is computed: $\epsilon_t = \epsilon_c + \epsilon_v$. The maxima are computed over all parameter sets and over both simulation methods.

For each parameter set, an example realization is shown in Fig. 2 for DS and in Fig. 3 for DSBC. They are ordered by increasing total error. Table 3 shows the five best parameter sets in terms of the total error for DS and DSBC. In general, DS has achieved lower errors, a slightly better performance than DSBC. The time per simulation versus the total error are shown in Fig. 4A and the histogram of *time* × ϵ_t in Fig. 4B. The DS results are also more diverse, they include some excellent simulations (with a good time-quality trade off) but also some mediocre ones. The ideal simulations lie in the lower left corner of Fig. 4A and the poor ones in the upper right corner. The DSBC results are generally more concentrated and generally good. This less dispersed results of DSBC are confirmed by the histogram of *time* × ϵ_t values (Fig. 4B) that shows that the distribution of DS results is wider. All these results suggest that choosing a good parameter set is easier with the DSBC parametrization,

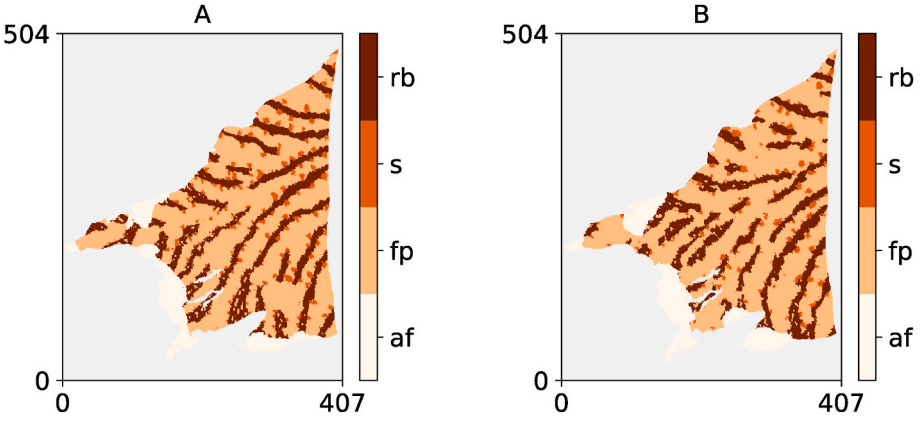


Fig. 10. Example of DS (A) and DSBC (B) simulations.

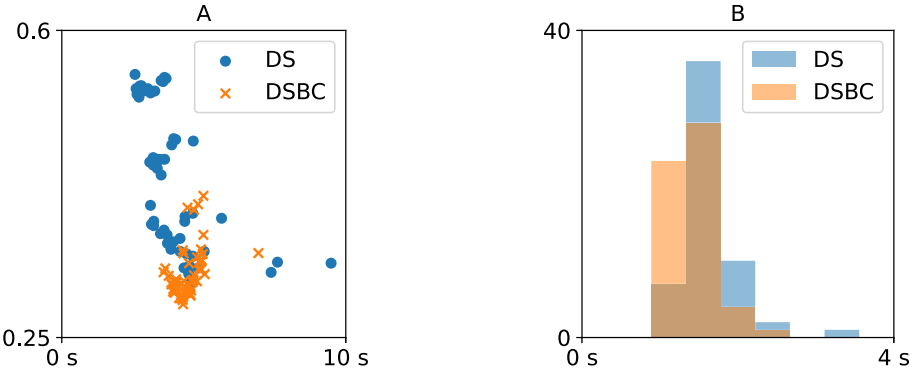


Fig. 11. Simulation time for Case 3 of a single realization versus the *CV* loss (A). Histogram of *time* × *CV* loss (B).

especially when only a few sets of parameters can be tested.

4.2. Test case 2. Continuous conditional simulations

This test case uses the methodology presented by Neven et al. (2021), which was employed to tune the DS parameters for simulating the Tsanfleuron glacier bedrock geometry in the Swiss Alps. The training image is the topography of Tsanfleuron glacier and the exposed bedrock after glacier retreat (Fig. 5) taken from Neven et al. (2021).

An area of 200 × 200 pixels is cut out from the TI, except for two perpendicular crossing lines which constitute the conditioning data (as an analog for ground penetrating radar (GPR) data). DS and the DSBC approach are used to reconstruct the missing area (by generating an ensemble of 40 realizations), different parameter sets are used (Table 4A for DS and Table 4B for DSBC). This procedure is repeated for 10 locations of bottom left corner (Table A.9).

An example area from the TI is depicted in Fig. 6A and the conditioning data in Fig. 6B. One DS and DSBC realizations out of 40 are shown in Fig. 6C and D, respectively. The pointwise CRPS errors are shown as a map Fig. 6E and F for DS and DSBC respectively.

The pointwise CRPS describes for each point in the simulation domain the CRPS score between the true elevation and the ensemble of simulated elevations for this point. When these scores are averaged over all locations, the mean pointwise error (MPE) is formed. The mean volume error (MVE) describes the error of the estimation of the glacier volume. When the bedrock elevations are first averaged over all locations for single realizations, and this distribution is compared to averaged true elevation, and then CRPS score computed; the MVE score is formed. More details about these scores were described in Neven et al. (2021).

Fig. 7A shows that DSBC achieved lower total error (MVE + MPE) values and therefore performed better in terms of simulation quality. As for the previous example, the scatter-plot and Fig. 7B show that the DSBC results are less dispersed than the DS results which show in particular more variability in terms of computing time. The detailed results are given in Tables A.10 and A.11 in the appendix.

4.3. Test case 3. Categorical multivariate simulation with cross-validation

The last test case is a synthetic, simplified, but realistic case, inspired from the model of the Roussillon aquifer in the South of France (Dal-l'Alba et al., 2020; Juda et al., 2020). In this example, conditioning data are available and can be used to tune the parameters of the geostatistical method. As in a real situation, we consider that the reality is unknown, and the only information are the conditioning data. In such a scenario, a K-fold cross-validation can be used to identify the best parameters (Juda et al., 2020).

The TI (Fig. 8C) is multivariate and has a trend (Fig. 8D) as a secondary variable. The simulation grid has also a trend attached to it (Fig. 8A) and the orientation of the patterns is guided by a rotation map (Fig. 8B). The TI has four categories: river bed (rb), crevasse splay (s), flood plain (fp) and alluvial fan (af).

We consider that 600 observation points are available (corresponding to borehole locations, Fig. 9). The 5-fold cross-validation approach with quadratic score (Juda et al., 2020) is used to compute the CV scores and their standard deviations for different parameters of DS and DSBC. The quadratic score lies between −2 and 0, with 0 corresponding to the ideal forecast. The quadratic (aka Brier) loss is the negative score, hence the lower, the better. The DS parameter sets (Table 5) and the DSBC parameter sets (Table 6) are very numerous, therefore only the 5 best results according to CV-score are reported for each method (Tables 7 and 8). Example simulations are shown in Fig. 10A and B for DS and DSBC respectively. The simulation plot versus CV loss (the lower, the better) and histogram of *time* × *CV* loss are shown in Fig. 11.

For this test case, DSBC yielded slightly higher cross-validation scores (Tables 7 and 8). Fig. 11A shows that the DSBC parametrization allows obtaining results that are in general better in terms of loss. Moreover, the distribution of the performance indicator is clearly better for DSBC with good performances being more common than for DS (Fig. 11B).

5. Summary and conclusion

This paper introduces a novel type of parametrization for the Direct Sampling algorithm and shows that this approach can help identify efficiently good parameter sets. We call this approach DSBC for Direct Sampling Best Candidate. DSBC can be considered as a simplified version or parametrization of the DS algorithm. Indeed, DSBC has only two major algorithmic parameters: the number of neighbors and the scan fraction, as compared to three for the DS method. This strategy facilitates parameter tuning, while retaining all the advantages and flexibility of the DS algorithm. As one could expect, since DSBC is a special case of DS, the three test cases that we studied confirmed that the quality and performance of DSBC is similar to DS. But what is important is that we show that DSBC is more likely to produce good simulations when the number of parameter sets that can be tested is limited by computational constraints. In two of the test cases, we could obtain better quality results with DSBC, and in one example the quality was similar between DS and DSBC. We consider that this is possible because when one has a limited computing resource available to test different parameter sets (as we did in the three examples), it is easier and faster to search in a two-dimensional space than in a three-dimensional space. DSBC can therefore be implemented as a simple and efficient default parameter tuning strategy for the existing DS implementations. Following these results, our recommendation is to start by applying the DSBC strategy and if the results are not satisfactory, then one can adjust more parameters using the complete DS algorithm as it has more tuning capabilities.

Finally, a possible direction for further research could be to use the DSBC idea to optimize the implementation of the algorithm using GPUs. The fact that the loop corresponding to the scan in the TI is simplified might lead to an optimal implementation tailored specifically for DSBC. A similar idea of performing full TI scan (removing distance threshold parameter and fraction scan, but introducing k parameter for "k-sampling") and specializing its implementation using FFT was proposed by Gravey and Mariethoz (2020). Their QuickSampling algorithm is statistically similar to DSBC (with $f = 1/k$) and computationally efficient, but it is not clear how it could include advanced DS features. Therefore, combining the flexibility of DSBC and computational advantages of QuickSampling could be very interesting.

Credit author statement

Przemysław Juda: Conceptualization, Methodology, Software, Investigation, Visualization, Writing – Original Draft. Philippe Renard: Conceptualization, Methodology, Writing – Review & Editing, Supervision. Julien Straubhaar: Conceptualization, Methodology, Software.

Code and data availability

The code and data documenting test cases are available in the following repository: https://github.com/randlab/simplified-ds. For the purpose of this study, DSBC algorithm was not reimplemented, instead a state-of-the-art Direct Sampling implementation (DeeSse) was used for both DS and DSBC simulations. The DeeSse implementation and python package geone can be consulted in the repository: https://github.com/randlab/geone.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence

the work reported in this paper.

Acknowledgements

The authors would like to thank Alexis Neven and Valentin Dall'Alba-Arnau for providing the data of the Tsanfleuron test case. Valentin Dall'Alba-Arnau also shared the Roussillon test case data.

Appendix A. Additional information and results

In this appendix, we provide some additional information and results related to the tests of the DSBC method.

Table A.9
Bottom left positions extracted images.

x	y
1200	500
800	500
1700	600
2600	800
1900	900
2200	700
1000	900
500	800
2700	1200
1500	1000

Table A.10
All the tested DS parameter sets in Case 2 with corresponding scores sorted according to MVE score.

n	f	t	MPE	MVE	time (s)
8	0.005	0.002	1.76	0.64	1.2
8	0.005	0.010	1.77	0.65	1.1
8	0.005	0.005	1.80	0.67	1.1
32	0.005	0.001	1.70	0.67	2.9
16	0.005	0.005	1.77	0.68	1.2
16	0.005	0.001	1.72	0.68	1.8
8	0.005	0.001	1.79	0.69	1.3
64	0.005	0.001	1.75	0.72	5.8
16	0.005	0.010	1.84	0.73	1.1
16	0.005	0.002	1.78	0.74	1.5
32	0.005	0.002	1.79	0.77	2.1
32	0.005	0.005	1.90	0.81	1.4
64	0.005	0.002	1.85	0.82	3.8
32	0.005	0.010	1.96	0.83	1.1
64	0.005	0.005	2.04	0.97	1.9
64	0.005	0.010	2.12	1.01	1.1

Table A.11
All the tested DSBC parameter sets in Case 2 with corresponding scores sorted according to MVE score.

n	f	MPE	MVE	time (s)
16	0.0005	1.58	0.58	2.2
16	0.0002	1.59	0.58	2.2
16	0.0010	1.58	0.59	2.9
32	0.0010	1.61	0.61	3.2
8	0.0010	1.66	0.61	2.6
32	0.0005	1.62	0.63	2.2
8	0.0002	1.68	0.65	2.2
32	0.0002	1.65	0.65	2.1
8	0.0001	1.71	0.66	2.3
8	0.0005	1.69	0.66	2.0
16	0.0001	1.61	0.66	2.3
32	0.0001	1.68	0.67	2.3
64	0.0002	1.82	0.73	2.2
64	0.0010	1.76	0.73	4.1
64	0.0005	1.78	0.73	2.7
64	0.0001	1.88	0.78	2.5

References

Dagasan, Y., Renard, P., Straubhaar, J., Erten, O., Topal, E., 2018. Automatic parameter tuning of multiple-point statistical simulations for lateritic bauxite deposits. Minerals 8, 220.

Dall'Alba, V., Renard, P., Straubhaar, J., Issautier, B., Duvail, C., Caballero, Y., 2020. 3D multiple-point statistics simulations of the Roussillon Continental Pliocene aquifer using DeeSse. Hydrol. Earth Syst. Sci. 24, 4997–5013. https://doi.org/10.5194/hess-24-4997-2020.

Gneiting, T., Balabdaoui, F., Raftery, A.E., 2007. Probabilistic forecasts, calibration and sharpness. J. Roy. Stat. Soc. B (Stat. Methodaol.) 69, 243–268.

Gneiting, T., Raftery, A.E., 2007. Strictly proper scoring rules, prediction, and estimation. J. Am. Stat. Assoc. 102, 359–378.

Gravey, M., Mariethoz, G., 2020. QuickSampling v1.0: a robust and simplified pixel-based multiple-point simulation approach. Geosci. Model Dev. (GMD) 13, 2611–2630. https://doi.org/10.5194/gmd-13-2611-2020.

Jäggli, C., Straubhaar, J., Renard, P., 2018. Parallelized adaptive importance sampling for solving inverse problems. Front. Earth Sci. 6, 203. https://doi.org/10.3389/feart.2018.00203.

Juda, P., Renard, P., Straubhaar, J., 2020. A framework for the cross-validation of categorical geostatistical simulations. Earth Space Sci. 7, e2020EA001152 https://doi.org/10.1029/2020EA001152.

Lam, D.T., Renard, P., Straubhaar, J., Kerrou, J., 2020. Multiresolution approach to condition categorical multiple-point realizations to dynamic data with iterative ensemble smoothing. Water Resour. Res. 56, e2019WR025875 https://doi.org/10.1029/2019WR025875.

Mariethoz, G., 2010. A general parallelization strategy for random path based geostatistical simulation methods. Comput. Geosci. 36, 953–958. https://doi.org/10.1016/j.cageo.2009.11.001.

Mariethoz, G., Caers, J., 2015. Multiple-Point Geostatistics: Stochastic Modeling with Training Images. John Wiley & Sons.

Mariethoz, G., Kelly, B.F., 2011. Modeling complex geological structures with elementary training images and transform-invariant distances. Water Resour. Res. 47.

Mariethoz, G., Renard, P., Straubhaar, J., 2010. The Direct Sampling method to perform multiple-point geostatistical simulations. Water Resour. Res. 46.

Meerschman, E., Pirot, G., Mariethoz, G., Straubhaar, J., Meirvenne, M.V., Renard, P., 2013. A practical guide to performing multiple-point statistical simulations with the Direct Sampling algorithm. Comput. Geosci. 52, 307–324.

Neven, A., Dall'Alba, V., Juda, P., Straubhaar, J., Renard, P., 2021. Ice volume and basal topography estimation using geostatistical methods and ground-penetrating radar measurements: application to the Tsanfleuron and Scex Rouge glaciers, Swiss Alps. Cryosphere 15, 5169–5186. https://doi.org/10.5194/tc-15-5169-2021.

Straubhaar, J., Renard, P., 2021. Conditioning multiple-point statistics simulation to inequality data. Earth Space Sci. 8 https://doi.org/10.1029/2020EA001515.

Straubhaar, J., Renard, P., Chugunova, T., 2020. Multiple-point statistics using multi-resolution images. Stoch. Environ. Res. Risk Assess. 34, 251–273.