Algorithm-driven and Representation-driven Random Function: A New Formalism for Applied Geostatistics

Alexandre Boucher
Dept of Geological and Environmental Sciences
Stanford University

Abstract

This paper makes the case for a new look at random function, one that is consistent with current and successful geostatistics practice. Two classes of random functions (RF) are presented defined either by an algorithm, termed algorithm-driven RF or explicitly through a set of (data-conditioned) representations, termed representation-driven RF. These new types of RF are alternatives to the classical analytical and stick closer to their actual utilization for modeling and imaging the spatial distribution of a natural phenomenon. The proposed RFs accepts that algorithm implementation is critical and that geostatistical simulations do not aim at visualization but at providing input to further processing by a transfer function such as a flow simulator.

1 Introduction

Stochastic methods are a proven and efficient tool to model our knowledge and simultaneously our uncertainty about a spatially distributed phenomena (Journel and Huijbregts, 1978; Chilès and Delfiner, 1999; Olea, 1999; Deutsch, 2002; Wackernagel, 1995; Goovaerts, 1997; Lantuejoul, 2002; Caers, 2005). For instance, from the information provided by a few wells with a sampling volume far less than 1 % of the total reservoir volume, the permeability distribution
in a petroleum reservoir is highly uncertain, in particular the spatial locations of critical high and low values are poorly known. The same remark applies to mining and environmental applications. Therefore, any numerical representation of that phenomenon must account for uncertainty. In geostatistics, such uncertainty is modeled with a random function (RF), and is quantified through a set of equiprobable realizations of that RF which take the form of alternative numerical models or maps. The random function provides a “structural model” for the spatial relations linking data and unsampled values, while the data allows to anchor the realizations at sample locations.

That both data and structural model contribute to the results of an estimation or simulation process is well understood. What is often forgotten is the impact of the algorithm used to implement the structural model and the conditioning to data. A mathematical model or RF, i.e. one fully defined by a set of equations, does not necessarily lend itself into an efficient algorithm. Often iterations or implicit approximations are necessary, such as taking the first terms of a series or limiting the extent of the type of data utilized.

Implementation of any model calls for an algorithm which carries necessarily specific approximations resulting in non predictable deviations from the model. Also, there may be alternative implementation algorithms each with its own set of approximations, none perfect. When actual data are involved as is the case for all practical situations, the impact of the algorithm is even more critical. The final data-conditioned results, whether kriging map or simulated realizations, depart from the original model because,

1. the data never fit perfectly the model,
2. the algorithm is already imperfect in reproducing the model prior to data and,
3. most critically, that algorithm imposes additional approximations when conditioning the model to the data.

As a consequence, in actual applications, the final results (those actually used) are never exactly as expected from the prior structural model, whether RF or else. However, one can still analyze these final results and reject them if deemed inappropriate (Tarantola 2006). The problem is that in many instances the impact of the algorithm overwhelms that of the prior model yet the modeler believes he still deals with the original analytical model with all its
clean and convenient properties. The honest solution consists of incorporating the algorithm within the model: the model is what the algorithm delivers! This calls for accepting that no result can be fully predicted or analyzed before-the-fact, that is before the prior model is implemented by the specific algorithm retained.

Instead of striving to produce an algorithm that fit best an ideal but limited-in scope RF model, we suggest to consider the algorithm as part of an implicit RF model then work on the algorithm to improve the model. Equations can be used to build a model but the algorithm implementation remains an integral part of that model and this should be formally recognized.

1.1 The Gaussian experience

Consider the simplest of all random function models, the standard multivariate normal model, hereafter called Gaussian model, fully characterized by a single covariance matrix. Even if that covariance matrix is inferred from the consistent normal score transformed data, the same used for estimation and simulation, it is highly unlikely that this Gaussian model is consistent with the data statistics of higher order. Sequential Gaussian simulation (sgsim) (Journel and Gómez-Hernández 1993) gives an excellent example of a pristine analytical RF, fully defined through clean and concise equations (Anderson 2003), but whose simulated realizations particularly after conditioning to real (not synthetic) data may display significant departures from the expected properties of the prior Gaussian RF.

Hence, this algorithm is “imperfect” due to unavoidable inconsistency between data and prior model. But much more troublesome is the fact that even in the absence of data (the case of non conditional simulation), that algorithm is already imperfect. Consider the non conditional simulation of a vector of \( N \) correlated Gaussian values, with \( N \) large, say \( N = 10^6 \), still a small size for a 3D application. The best algorithm for implementing that Gaussian model and generating such simulation would call for the LU decomposition of the \( N \) by \( N \) covariance matrix (Davis 1987), a task that would ask for approximations with not fully predictable consequences on the actual statistics of the simulated realizations. In short, the results one gets is not that specified by the model but that specified by the complex combination \{ prior model + somewhat inconsistent data + an implementation algorithm with its specific approximations \}. We will
define in the next section such a combination as an algorithm-driven random function (ADRF). The deviations from the prior model brought by the data and the implementation algorithm, although unpredictable, can be observed on the results, after-the-fact. For example, one can draw many realizations from that algorithm-driven random function and observe their statistics of any order [Deutsch 1994]. In many cases, the deviations from the prior model although substantial would be deemed beneficial because,

1. those deviations go towards aspects of the conditioning data that were not or could not be accounted for by the prior model

2. the algorithm was purposely designed to deviate from unwelcome side effects of the model.

Even if modern computers could handle with great accuracy the LU decomposition of the previous very large covariance matrix [Vargas-Guzman and Dimitrakopoulos 2002] [Dimitrakopoulos and Luo 2004], one may still prefer using a sequential Gaussian algorithm with limited data neighborhoods and ordinary kriging replacing simple kriging. Indeed the latter algorithm shows greater robustness with regard to local departure of the data from the global stationarity required to infer the large \((N \times N)\) covariance matrix. But that sequential Gaussian simulation algorithm brings in its own set of approximations and departure from the clean, albeit restrictive, properties of the Gaussian RF model. In other words, one may prefer the algorithm-driven results of sequential Gaussian simulation to the results of the model-consistent LU algorithm. Some may argue that the numerous approximations of the sequential Gaussian algorithm prevent from predicting the distribution of the simulated results; in particular that distribution would not be anymore Gaussian. The answers to such criticism are multiple:

1. Since the conditioning data most likely do not fit a multivariate Gaussian RF, the results of the more rigorous LU decomposition algorithm are equally non Gaussian and as unpredictable before actually running the algorithm and checking the stats of the results.

2. Why should one be dogmatic about the results being Gaussian given that the prior Gaussian model is but only a model, not unique and known to be inconsistent with the data?

3. Last and not least for applications, sequential simulation over a large field
is order of magnitude faster than the LU decomposition approach.

Regarding point (2) above, it serves to recall that a random function and its spatial law (multivariate distribution) is only a model, not some physical reality that one should strive to identify [Dubrule, 1994]. A model is, by definition, not unique; its goodness can be judged only by its ability to provide the results expected. For the same data set, different models could be considered for different goals. If a model is to be judged by its results, it is perfectly acceptable to design the implementation algorithm to modify the starting model towards it yielding better results. Last, since there is no practice of a model without an implementation algorithm, and any practical implementation implies some departure from that model, why not accept that a model can only be defined fully through its implementation algorithm, which is exactly the concept of an algorithm-driven random function?

1.2 The need for a new formalism

Should an algorithm be ignored just because the underlying RF cannot receive a concise analytical formulation? The simple fact that an algorithm produces realizations (and these realizations are reproducible) indicates that there is a RF; that RF is simply not analytically known.

This is not a new concept; the modeling of phenomena with high complexity such as climate can only be done with computer-based algorithms. Even the best such model can be seen as a patchwork of assumptions, calling for approximations and educated guesses. The complexity is such that no analytical formulation can represent or capture the non-linear intrication of the different components and parameters of the problem. To force an analytical solution would lead to such over simplifications that the resulting model would no longer be in agreement with the observations not to mention the physics of the phenomenon itself. Algorithms specifically designed for a particular application provide a custom-made, fine-tuned, framework to integrate more information about the specific process under study.

In most applications, the critical variable is the output decision variable, i.e. the final variable that triggers the set of decisions. That decision variable is a function of the spatially distributed variables plus many other engineering parameters pooled together through a typically non-analytical transfer function,
an example of which is a flow simulator. The compound effect of all approximations and assumptions involved and the non linearity of the transfer function cannot be summarized from a set of equations.

Figure 1: Overview of the sequence of choosing the simulation algorithm, generating a number of realizations and their processing by a transfer function into a decision variable.

The geostatistical solution consists of generating alternative numerical models of the spatially distributed variables, then process each such realization through the transfer function, see Figure 1. These numerical models are treated as equiprobable realizations of the unknown spatially distributed reality. Geostatistical modeling aims at imaging the spatial process towards a specific goal, that of helping into a specific decision making process. The permeability field in reservoir modeling is not simulated for visualization, but to be processed through a flow simulator. The key is that these input realizations must be equiprobable to ensure an appropriate transfer of the spatial uncertainty depicted by these alternative realizations. If the model is analytical, the implementation must be such that the realizations generated constitute an unbiased sampling of the RF, not an easy requirement to check.

In summary, two practical observations drive the need to define and develop new classes of random functions:

**Observation 1** In most applications, implementation modifies in an unpredictable way the properties of the prior model.

**Observation 2** The uncertainty of interest is not that related to the spatially distributed attributes (be it gold grades or permeability values). It is the uncertainty related to the decision variable resulting from application of
transfer function. That transfer function being complex there is no ana-
lytical linkage between the input spatial variable and the output decision
variable.

2 Definitions

We propose two types of non analytical RF. The first is the algorithm-driven
RF (ADRF), it is implicitly defined by an algorithm and its parameters. The
second is the representation-driven RF (RDRF) which is explicitely defined by
a finite number of numerical representations. The ADRF is build in response
to Observation 1; because the results actually used depend so much on the
implementation algorithm, that algorithm is made part of the model. The
RDRF responds to Observation 2; given that in practice only a limited number
of realizations can be processed through the transfer function and the final
results depend on these specific realizations, the RF model should make explicit
these realizations.

2.1 Algorithm-driven RF (ADRF)

An ADRF type, denoted $Z(u)$, is defined by an algorithm $F(\eta; \theta)$ parametrized
by a vector of parameters $\theta$ and a seed number $\eta$. That seed seed number
is subsequently used to initiate the pseudo-random number generator. The
ADRF contains all the $L'$ realizations $z(l)(u), l = 1, \ldots, L'$ that can be generated
with its algorithm $F(\eta; \theta)$. Note that $L'$ can be infinity. Selecting $L$ seed
values $\eta_l$, $l = 1, \ldots, L$ allows sampling that ADRF by generating $L$ realizations
$z(l)(u), l = 1, \ldots, L$. We will use the symbolic notation:

$$F(\eta_l; \theta) \mapsto z(l)$$

Note that for a given set $\theta$ of parameters values, the probability distribution
of the ADRF is that of the seed number $\eta$. If that seed number $\eta$ is uni-
formly distributed then the corresponding realizations $z(l)(u)$ of the ADRF are
equiprobable. We are assuming that the algorithm underlying the ADRF is
such that any single realization $z(l)(u)$ is completely identified by $\eta_l$, that is it
can be completely and exactly retrieved through its seed number.

Two realizations are said to be different if their indices are different $l \neq l'$. 

$$F(\eta_l; \theta) \mapsto z(l)$$

Note that for a given set $\theta$ of parameters values, the probability distribution
of the ADRF is that of the seed number $\eta$. If that seed number $\eta$ is uni-
formly distributed then the corresponding realizations $z(l)(u)$ of the ADRF are
equiprobable. We are assuming that the algorithm underlying the ADRF is
such that any single realization $z(l)(u)$ is completely identified by $\eta_l$, that is it
can be completely and exactly retrieved through its seed number.

Two realizations are said to be different if their indices are different $l \neq l'$. 

$$F(\eta_l; \theta) \mapsto z(l)$$

Note that for a given set $\theta$ of parameters values, the probability distribution
of the ADRF is that of the seed number $\eta$. If that seed number $\eta$ is uni-
formly distributed then the corresponding realizations $z(l)(u)$ of the ADRF are
equiprobable. We are assuming that the algorithm underlying the ADRF is
such that any single realization $z(l)(u)$ is completely identified by $\eta_l$, that is it
can be completely and exactly retrieved through its seed number.
even if the values of these two realizations are all equal:

\[ \{z(l)(u), u \in S\} \neq \{z(l')(u), u \in S\} \quad \text{iff} \quad l \neq l' \]  \hspace{1cm} (2)

Then, by definition every seed number \( \eta \) defines a realization \( z(l)(u) \) of that ARDF.

From an ADRF perspective, an algorithm is not developed to ensure a clean implementation of some prior analytical RF, the ADRF replaces or voids the need for such an analytical formulation. Properties of the ADRF can be evaluated a posteriori by generating any number \( L \) (possibly very large) of realizations. Whereas the implementation of an analytical RF is validated by comparing its realizations properties to the properties expected from the model, the properties of an ADRF are those of its realizations.

Note that in practice the advantage of having access to a possibly infinite pool of realizations \( (L \to \infty) \) is moot. As stated above in Observation 2, it is the uncertainty of the decision variable which is relevant. With limited time and computing power only a finite number of specific realizations \( z(l)(u), l = 1, \ldots, L \) can be processed to infer the distribution of the decision variable(s). In the end, what matters are the \( L \) realizations drawn and actually processed through the transfer function.

### 2.1.1 Conditioning realizations of an ADRF

Conceptually, conditioning an ADRF is done by ensuring that its defining algorithm is able to honor all the data provided. The notation \( [1] \) is then extended to:

\[ Z(u|\Omega) = F(\eta; \theta|\Omega) \mapsto \{z(l)(u|\Omega), l = 1, 2, \ldots, L'\} \]  \hspace{1cm} (3)

The observation data \( \Omega \) can be “hard data”, e.g. direct measurements of the attributes \( Z \) defined over any support volume, or include indirect data related to non-linear functions of multiple \( Z \)-values such as a well test or known production data. Any algorithm that can reproduce to an acceptable level the data retained for conditioning would define its own ADRF. Note that the ADRF formalism requires that \( F(\eta; \theta) = F(\eta; \theta|\Omega = \text{null}) \) to include non-conditional simulations.

The main purpose of conditioning is to anchor a structural model to known observations. The spatial patterns of the structural model can be visualized by generating a set of unconditional simulations. A good algorithm will minimize
the distortion of these spatial patterns when morphing them to match data. Both conditional and non-conditional simulations should share the same structural patterns. This requires to identify which patterns are of importance for the final result (the decision variable), then define an ADRF that can generate these patterns correctly anchored to data.

Whenever the shape and distribution of patterns vary significantly between unconditional and conditional simulations, then

- either the structural model is deemed inconsistent with the data, then that structural model should be changed which amounts to change the ADRF.
- or the implementation algorithm for data conditioning is not good enough. Hence it should be changed which amounts to change the ADRF.
- or the data are considered only soft and their reproduction can be lax and the original algorithm can be kept which amounts not to change the ADRF.

Note that any algorithm that cannot perform data conditioning is of no use in practice and is only of academic interest.

2.2 Representation-driven RF (RDRF)

Instead of considering all the possible realization \( \{ z^{(l)}(u), l = 1, ..., L' \} \) of an ADRF with \( L' \) possibly infinite, we proposed to consider the more practical situation of a RF represented by a finite, although possibly very large, number \( L \) of numerical models of the natural phenomenon. The representation-driven RF (RDRF) is explicitly defined by this finite set of \( L \) representations:

\[ Z(u) \mapsto \{ z^{(l)}(u), l = 1, ..., L \} \tag{4} \]

As opposed to an ADRF defined by an algorithm \( F(\eta; \theta) \), the RDRF makes no mention of the generating algorithm algorithm but only of its result, a set of \( L \) explicit numerical representations \( \{ z^{(l)}(u), l = 1, ..., L \} \). A realization is then one representation drawn among the finite number \( L \) of representations constituting that RDRF. This type of RF is the only one that is fully explicit in the sense that any expert can fully visualize it and decide if it is suitable for the application at hand.
In all generality, the \( L \) representations of an RDRF need not be generated by the same algorithm, some can be computer-generated, others can be hand drawn. Again, the algorithm(s) used to generate the representations are irrelevant; an RDRF should be judged solely from its \( L \) constitutive representations irrespective of how they were generated.

Furthermore, it is possible to define a priori the probability of occurrence of each of the \( L \) realizations. In most applications, however, equal probability will be considered with each realization having probability \( 1/L \). Availability of additional information or expert opinion could be used to give higher or lower probability of occurrence to some representations.

The main advantage of the RDRF is that it is unequivocally defined by a finite number \( L \) of visually explicit realizations; there is no “what” or “if other realizations” were to be drawn. It also sticks closer to actual practice whereby the input spatial uncertainty is frozen to be that provided by the \( L \) realizations retained for processing by the transfer function. The uncertainty is given by the \( L \) representations; adding one additional representation amounts to generating another RDRF.

A conditional RDRF is such that all its \( L \) constitutive representations honor all data to an acceptable level of exactitude:

\[
Z(u|\Omega) = \{z^{(l)}(u|\Omega), l = 1, 2, ..., L\}
\]  

(5)

3 Validating an ADRF or RDRF

The choice of any random function is necessarily a subjective decision since natural phenomena are not created by any RF nor even by a computer-based algorithm but from past physical processes. To use a probabilistic formulation for a natural phenomena at any scale larger than the atom is to acknowledge our inability to model its underlying physics. The random function model only aims at a direct after-the-fact representation of the (spatial) patterns which are the consequences of the physical phenomenon. That random function model rarely aims at re-creating step-by-step the physical processes.

Whichever model is chosen, analytical RF, ADRF or RDRF, that choice is necessarily subjective. For instance, there is no a priori objective reason to choose a Gaussian RF to model permeability or to retain a training image-based
algorithm, such as snesim [Strebelle 2002], to model a clastic reservoir. The choice of a particular algorithm or set of $L$ representations is based on many subjective criteria, some being dependent on the particularly transfer function used to process these realizations.

Deutsch (1994) proposes three criteria to select an “algorithmically-defined” RF; a good algorithm must 1) be executed in a reasonable amount of time 2) integrate the maximum of prior information and 3) maximize the entropy of the distribution of the decision variable. These guidelines remains appropriate for the extended ADRF formalism proposed in this paper. However, in any practical application it would be difficult if not impossible to select the algorithm that ensures maximization of the entropy of the distribution of the decision variable, because of the complexity of the non linear transfer function. Instead between two ADRFs or two RDRFs of equal performance as to the two first criteria, it suffices to retain the one that maximizes the entropy of the distribution of decision variable.

From the point of view of structural patterns reproduction, choosing and fine-tuning an algorithm $F(\eta; \theta)$ and inferring its parameters $\theta$ can be done by performing non-conditional realizations. The structural model implicit to that generation algorithm can then be visualized and validated in regards to the prior structural expectation and its relevance to the transfer function retained.

The selection of an ADRF and the inference of its parameters are open questions and questions that are case-dependent. A good ADRF should be such that the algorithm parameters can be easily inferred/optimized. This is no different from selecting an analytical RF, with some being easier to infer than others. The useability of an ADRF depends on that ease of inference of its critical parameters.

The building of a RDRF is done by first selecting one or several algorithm(s) to generate the $L$ numerical representations. Each of these representations must honor all data and be a valid image representative of the natural phenomenon. Most of the time, a RDRF is build from a set of $L$ realizations from an ADRF.

The spatial properties of an ADRF or a RDRF can only be characterized a posteriori. The spatial statistics of an ADRF is calculated from any number of realizations, while those of an RDRF are computed from the particular set of $L$ representations that defines it.
3.1 ADRF-RDRF and nonlinear transfer function

The usefulness of random functions lies in their ability to model uncertainty through their simulated realizations. However, as mentioned before, the goal is not in the realizations themselves but in their processing through a transfer function. This processing is usually expensive, e.g. through a flow simulator or a mine planning software, and can only performed on a few realizations.

Consider the decision random variable $T$, obtained by processing the spatial RF $Z(u)$ by the transfer function $Ψ(·)$.

$$T = Ψ(Z(u))$$  \hspace{1cm} (6)

In case of a reasonably complex and non linear transfer function $Ψ(·)$ the complete distribution of $T$ calls for processing all the $L$ realizations of $Z(u)$, such that

$$\{t(1), t(2), ..., t(L)\} = \{Ψ(z(1)(u)), Ψ(z(2)(u)), ..., Ψ(z(L)(u))\}$$

Hence the sampling of $T$ is directly related to that of $Z(u)$. Irrespective of other realizations that could have been generated with an ADRF or an analytical RF, the distribution of $T$ is unequivocally identified by the previous $L$ realizations $t(l), l = 1, ..., L$.

To further understand the differences between ADRF and RDRF consider the following hypothetical, yet common, scenario. One wants to estimate the time that a contaminant takes to travel between two wells using a flow simulator: $T = Ψ(Z(u))$, where $Z(u)$ is the permeability field. With a specific algorithm $F(η|Ω; θ)$ $L$ representations are generated $\{z(1)(u), ..., z(L)(u)\}$ and processed by the flow simulator into $L$ travel times $\{t(1), ..., t(L)\}$. To check the robustness of this latter distribution, $L$ more simulations $\{z(L+1)(u), ..., z(2L)(u)\}$ are generated and processed yielding a second distribution $\{t(L+1), ..., t(2L)\}$.

With an ADRF model, one can evaluate the robustness of the first distribution $\{t(1), ..., t(L)\}$. If the second distribution $\{t(L+1), ..., t(2L)\}$ is deemed significantly different from that first one, then one can conclude that the first $L$ realizations of $Z(u)$ were not sufficient to characterize the consequent distribution of the contaminant travel time $T$. Note that both set of realizations are coming from the same ADRF.

The interpretation is different when using the RDRF concept: one would be comparing the outputs of two different RDRFs, the first one with $L$ realiza-
tions \{z^{(1)}(u), ..., z^{(L)}(u)\}, and a second one with the \(L\) different realizations \{z^{(L+1)}(u), ..., z^{(2L)}(u)\}. One would like these two RDRFs to yield similar distributions for the travel time \(T\). By merging the two sets of realizations into one of size \(2L\), one define yet a third RDRF. Everytime a new representation \(z^{(l)}(u)\) is added to an existing set of representations, a new RDRF is being considered. Note also by adding \(L'\) representations to the original \(L\) realizations the equal likelihood of each realization decreases from \(1/L\) to \(1/(L+L')\).

4 Discussion

This paper has proposed a formalism for creating new random functions that are more consistent with the current and foreseeable practice in geostatistics. Instead of developing an algorithm that tries to mimic the properties of an analytical probabilistic model, the algorithm-driven RF (ADRF) includes in its definition the implementation algorithm while the representation-driven RF (RDRF) is characterized by a finite set of explicit numerical representations. Both ADRF and RDRF provide equiprobable realizations by definition, this is not an assumption but a property of the new random functions.

We stress that these random functions are neither a replacement for the more traditional analytical RF, nor an excuse for producing adhoc algorithms without a solid understanding of theory. They are valuables alternatives to the RF models. Instead of judging an algorithm in terms of it approximating an analytical formulation, an algorithm should be judged based on its output which are the realizations generated. An algorithm that fails to reproduce critical features is a bad one no matter the pedigree of the underlying model.

Of the two models proposed, ADRF and RDRF, the former is closest to the classical concept of a random function but with algorithms and computer codes replacing the equations. It also suffers a drawback shared by analytical RFs, that is providing a potentially large pool of possible realizations which can never be fully explored.

The RDRF model is more atuned to utilization through a transfer function, as no claim is made about the existence of a possibly infinite numbers of alternative realizations.
4.1 Open source software and ADRF

Since an ADRF includes in its definition the implementation algorithm, that algorithm including its computer code should be supplied. The inspection and testing of the code should be done carefully and is as necessary as the understanding of the theory underlying a traditional RF.

We strongly believe that open source softwares are necessary with these type of RF. Not disclosing the code is analogous of not writing down the equations of an analytical RF. It is only by having access to the code that one can fully appreciate the assumptions and simplifications that were implemented.

5 Conclusion

The formalism of algorithm-driven and representation-drive RF is in line with the current practice and research in geostatistics. With the focus of these RF weighted toward the end-results, that is the simulated realizations, these two RF classes provides more latitude in developing algorithms. It also has the potential to lessen the grip of Gaussian-related RF in the modeling of natural phenomenon. Above all, it recognizes that the only component of a RF that is actually used is the realizations that have been actually drawn.

References


