Computers & Geosciences 80 (2015) 49–61

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

A conflict-free, path-level parallelization approach for sequential simulation algorithms

Luiz Gustavo Rasera *, Péricles Lopes Machado, João Felipe C.L. Costa

Department of Mining Engineering, Federal University of Rio Grande do Sul, Av. Bento Gonçalves 9500, Porto Alegre-RS, 91501-970, Brazil

ARTICLE INFO

Article history:
Received 11 November 2014
Received in revised form
31 March 2015
Accepted 31 March 2015
Available online 1 April 2015

Keywords:
Geostatistics
Sequential simulation
Parallelization
Random path
Conflict-free

ABSTRACT

Pixel-based simulation algorithms are the most widely used geostatistical technique for characterizing the spatial distribution of natural resources. However, sequential simulation does not scale well for stochastic simulation on very large grids, which are now commonly found in many petroleum, mining, and environmental studies. With the availability of multiple-processor computers, there is an opportunity to develop parallelization schemes for these algorithms to increase their performance and efficiency. Here we present a conflict-free, path-level parallelization strategy for sequential simulation. The method consists of partitioning the simulation grid into a set of groups of nodes and delegating all available processors for simulation of multiple groups of nodes concurrently. An automated classification procedure determines which groups are simulated in parallel according to their spatial arrangement in the simulation grid. The major advantage of this approach is that it does not require conflict resolution operations, and thus allows exact reproduction of results. Besides offering a large performance gain when compared to the traditional serial implementation, the method provides efficient use of computational resources and is generic enough to be adapted to several sequential algorithms.

1. Introduction

Numerical modeling of natural phenomena is commonly performed with geostatistical techniques owing to their conditioning capabilities and relatively fast processing time when compared to process-based and surface-based methods. These techniques rely on spatial correlation of available data for statistical inference of parameters required to derive the uncertainty distribution at uninformed grid nodes.

In the last few years, the size of the simulation grids used to represent geological models in petroleum, mining, and environmental applications has increased by orders of magnitude. Although efficient pixel-based geostatistical simulation algorithms have been developed, the computational costs remain high for very large grids. With the availability of multiple-processor computers and multicore central processing units (CPUs), as well as graphics processing units (GPUs) for general purpose computing, development of parallel schemes for traditional sequential simulation algorithms is essential for full utilization of these resources, which can reduce the execution time and optimize memory consumption. Moreover, such a strategy can contribute to further increases in the resolution and overall size of these simulation grids.

Parallel computing is a common strategy for reducing processing time and increasing the resolution capacity (scalability) of algorithms. Despite their benefit in terms of performance, some of the parallel versions of sequential algorithms require conflict management systems. This is especially true when one is considering parallelization of a simulation at the path level. Conflicts occur when nodes being simulated in parallel have overlapping neighborhoods. In order to preserve the statistical quality of the results, a system must manage these conflicts through resimulation, waiting, or postponing operations. The biggest issue related to this approach is that it does not allow the exact reproduction of results, making it unsuitable to several applications.

Here we present a general framework consisting of a straightforward conflict-free parallelization method that can be applied to any sequential simulation algorithm using shared or distributed memory architectures. The proposed strategy is a path-level parallelization approach mainly inspired by previous work by Vargas et al. (2007). Our method avoids the requirement for a conflict resolution system, allowing the reproduction of results, but also minimizing the amount of communication between processors and waiting operations.

The methodology is illustrated by practical implementation of a parallel version of the sequential Gaussian simulation (SGS) algorithm as a plugin in the SGeMS software (Remy et al., 2009). The results are compared with classical serial SGS implementation.

* Corresponding author.
E-mail address: lgrasera@gmail.com (L.G. Rasera).

http://dx.doi.org/10.1016/j.cageo.2015.03.016
0098-3004/

2. Parallelization strategies for sequential simulation

Sequential simulation comprises a broad class of simulation methods widely used in geostatistics to generate realizations of random fields (Goovaerts, 1997; Deutsch and Journel, 1998; Chiles and Delfiner, 1999; Caers, 2005; Remy et al., 2009). This large group of methods includes traditional techniques based on two-point statistics, such as SGS (Isaaks, 1990; Journel, 1994), sequential indicator simulation (SIS) (Isaaks, 1984; Journel and Alabert, 1989), direct sequential simulation (DSS) (Soares, 2001), and their co-simulation variants (Goovaerts, 1997; Chiles and Delfiner, 1999). The majority of these techniques use a variogram as the main tool for characterizing spatial continuity. Multiple-point statistics (MPS) algorithms such as SNESIM (Strebelle, 2002), Direct Sampling (Mariethoz et al., 2010), and IMPALA (Straubhaar et al., 2011) rely on training images for statistical inference of spatial patterns.

Despite their specific features, all sequential simulation methods follow the same formalism to generate a set of realizations. These techniques are based on decomposition of the multivariate probability density function (pdf) of a stationary and ergodic random function to the product of univariate posterior distribution functions (Scheuer and Stoller, 1962; Rubinstein, 1981; Johnson, 1987; Ripley, 1987). The decomposition process allows modeling and sampling of a one-point conditional cumulative distribution function (ccdf) at each node of the simulation grid. To ensure that the spatial structure of the phenomena is reproduced, each local ccdf is made conditional not only on the available hard data but also on previously simulated nodes. These conditional distributions are determined by a spatial model that describes the spatial patterns of the random field. The spatial model can be a single variogram or a set of variograms in the case of SGS and SIS, respectively, or a training image and its corresponding search tree in MPS methods. In general, grid nodes are visited along a random path, but alternative simulation sequences (e.g., raster paths) can also be adopted.

All the aforementioned simulation methods share a common peculiarity: they are all based on serial simulation of a single node at a time. This strategy offers excellent conditioning capabilities, but it is also extremely inefficient for simulation on large grids. Exceptions to this node-by-node framework are the FILTERSIM (Zhang et al., 2006), SIMPAT (Arpat and Caers, 2007), DISPAT (Honarkhah and Caers, 2010), CCSIM/MS-CCSIM (Tahmasebi et al., 2012, 2014a) and CIQ (Mahmud et al., 2014) algorithms. These algorithms are patch-based simulation methods. Instead of simulating a single node at a time, these techniques simulate groups of nodes depicting entire geometric patterns extracted from a training image. These methods provide a significant improvement in performance, but they suffer from conditioning problems and may generate models with reduced spatial uncertainty (i.e., less variability among realizations).

Mariethoz (2010) discussed different parallelization approaches for sequential simulation in detail. The author classified parallelization strategies at three distinct levels: realization, path, and node levels.

Parallelization at realization level involves computation of each realization by a different processor. The major advantage in parallelizing at this level is that no communication between processors is required. However, this is an inappropriate solution when performing sensitivity analyses of results for specific parameters in cases with large simulation grids.

The second approach consists of parallelizing at the path level. In this case, the simulation grid can be subdivided into zones composed of groups of nodes, and different processors are assigned to compute the calculations for each zone (Vargas et al., 2007). A limiting factor is that this strategy is only practical in situations in which the size of the simulation grid is significantly greater than the dimensions of the search neighborhood. Fortunately, this is usually the case for numerical models of petroleum reservoirs and most metal deposits. Mariethoz (2010) proposed a master–slave parallelization architecture that distributes the grid nodes among several processors. A master processor is responsible for managing the path, search neighborhoods, and conflicts, while the slave processors carry out the simulation itself. More recently, Tahmasebi et al. (2012) presented a path-level parallelization methodology also based on a master–slave architecture, but using a processor as the master and several processors in GPUs as slaves. Furthermore, specific GPU-based parallel computing schemes for SNESIM and Direct Sampling algorithms were presented by Huang et al. (2013b) and Huang et al. (2013a), respectively.

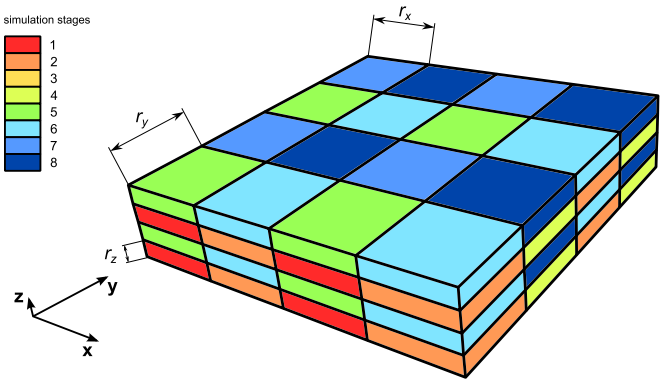

Fig. 1. PGPs simulation stages.

The main drawback of the master–slave architecture is that it generates conflicts when a node to be simulated has other nodes within its neighborhood being simulated by other processors. Remediation of conflicts commonly requires resimulation procedures, alterations in the simulation path, waiting operations, or postponing strategies (Mariethoz, 2010; Tahmasebi et al., 2012). In distributed memory machines, these actions increase the number of messages between processors, which consequently slows down the simulation process. Moreover, a major disadvantage of conflict resolution procedures is that they do not allow reproducible results, even in systems with exactly the same hardware configurations. The third option consists of parallelizing at the node level. In this approach, computation of each grid node is parallelized (Nunes and Almeida, 2010; Peredo and Ortiz, 2011; Straubhaar et al., 2011; Huang et al., 2013b).

In the following section, the proposed method is presented. The methodology consists of a path-level parallelization scheme for sequential simulation that avoids the use of conflict resolution procedures. The algorithm is an evolution of the methodology presented by Vargas et al. (2007), but instead of adapting a single random path for simulation of each subzone of the grid, the algorithm generates multiple paths for each zone from a single global seed. In addition, the concept of parallel sections introduced by Vargas et al. (2007) is generalized and replaced by defining parallel grid partitions (PGPs) and simulation stages to manage parallel simulations of the grid partitions.

3. Proposed method

Consider a grid G_N composed of N nodes discretizing a stationary random field $Z(\mathbf{u}_i)$, $\mathbf{u}_i \in \mathbf{R}^n$, $i = 1, \ldots, N$, also denoted as Z_i, and a set of conditioning data $\mathbf{d}_n = \{z(\mathbf{u}_\alpha), \alpha = 1, \ldots, n\}$. In addition, consider the data set Λ_i, which can contain both hard data