Original paper

A parallel computing strategy for Monte Carlo simulation using groundwater models

Esther Leyva-Suárez, Graciela S. Herrera* and Luis M. de la Cruz

Received: August 15, 2013; accepted: January 01, 2014; published on line: June 30, 2015

DOI: 10.1016/j.gi.2015.04.020

Resumen

En este artículo se presentan los resultados de una estrategia de paralelización para reducir el tiempo de ejecución al aplicar la simulación Monte Carlo con un gran número de realizaciones obtenidas utilizando un modelo de flujo y transporte de agua subterránea. Desarrollamos un script en Python usando mpi4py, a fin de ejecutar GWMC y programas relacionados en paralelo aplicando la biblioteca MPI. Nuestro enfoque consiste en calcular las entradas iniciales para cada realización y correr grupos de estas realizaciones en procesadores separados y después calcular el vector medio y la matriz de covarianza de las mismas. Esta estrategia se aplicó al estudio de un acuífero simplificado en un dominio rectangular de una sola capa. Presentamos los resultados de aceleración y eficiencia para 1000, 2000 y 4000 realizaciones para diferente número de procesadores. Eficiencias de 0,70, 0,76 y 0,75 se obtuvieron para 64, 64 y 96 procesadores, respectivamente. Observamos una mejora ligera del rendimiento a medida que aumenta el número de realizaciones.

Palabras clave: Agua subterránea, flujo y transporte, simulación Monte Carlo, cómputo paralelo distribuido, Python.

Abstract

In this paper we present the results of a parallelization strategy to reduce the execution time for applying Monte Carlo simulation with a large number of realizations obtained using a groundwater flow and transport model. We develop a script in Python using mpi4py, in order to execute GWMC and related programs in parallel, applying the MPI library. Our approach is to calculate the initial inputs for each realization, and run groups of these realizations in separate processors and afterwards to calculate the mean vector and the covariance matrix of them. This strategy was applied to the study of a simplified aquifer in a rectangular domain of a single layer. We report the results of speedup and efficiency for 1000, 2000 and 4000 realizations for different number of processors. Efficiencies of 0.70, 0.76 and 0.75 were obtained for 64, 64 and 96 processors, respectively. We observe a slightly improvement of the performance as the number of realizations is increased.

Key words: groundwater, flow and transport, Monte Carlo simulation, distributed parallel computing, Python.

E. Leyva-Suárez
Posgrado en Ciencias de la Tierra
Universidad Nacional Autónoma de México
Ciudad Universitaria
Delegación Coyoacán, 04510
México D.F., México

G. S. Herrera*
L. M. de la Cruz
Instituto de Geofísica
Universidad Nacional Autónoma de México
Ciudad Universitaria
Delegación Coyoacán, 04510
México D.F., México
*Corresponding author: ghz@geofisica.unam.mx

Introduction

Stochastic hydrogeology is a field that deals with stochastic methods to describe and analyze groundwater processes (Renard, 2007). An important part of it consists of solving stochastic models (stochastic partial differential equations) describing those processes in order to estimate the joint probability density function of the parameters (e.g., transmissivity, storativity) and/or state variables (e.g., groundwater levels, concentrations) of those equations or more commonly some of their moments. Monte Carlo simulation (MCS) is an alternative for solving these stochastic models, it is based on the idea of approximating the solution of stochastic processes using a large number of equally likely realizations. For example, the pioneering work on stochastic hydrogeology by Freeze (1975) applies this method.

The large number of realizations required by MCS can be very demanding in computing resources and the computational time can be excessive. Nowadays there exist many parallel computing platforms that can be used to alleviate this problem. Some previous works have focused in this direction, for example Dong *et al*. (2012) describe a parallelization strategy for stochastic modeling of groundwater systems using the Java Parallel Processing Framework (JPPF). This tool is very powerful and can be used as a GRID middle-ware (Foster *et al*., 2001) to distribute tasks across several computing systems. Dong *et al*. (2012) take advantage of this tool to avoid any modification of MODFLOW and related programs. However, when the JPPF is used in a cluster alone, a simply master-worker parallel model is obtained. They also report that the combination of two levels of parallelism, using a parallel solver to reduce the execution time by an order of two. However, this technique pays off only for very large grids, over 10^6 points. In our case, we do not require such massive grids.

In this paper, we propose a distributed parallel computing method for stochastic modeling with the software *Groundwater Monte Carlo* (GWMC), a component of the Groundwater Quality Monitoring (GWQMonitor) package (Herrera, 1998). GWMC is used together with an assimilation method called Ensemble Smoother of Herrera (ESH) in order to estimate groundwater contaminant concentration assimilating concentration data. The best known version of Ensemble Smoother was developed by van Leeuwen and Evensen (1996). Herrera de Olivares developed a version of the assimilation method independently, and originally she called it static Kalman filter (Herrera, 1998).

Ground Water Monte Carlo (GWMC)

GWMC is a program written in FORTRAN by Herrera (1998), and subsequently modified by Olivares-Vázquez (2002).

It implements Monte Carlo simulation using a flow and transport simulator in which hydraulic conductivity is a random field and the contaminant concentration at the contaminant source is a time series at each node. Therefore, multiple realizations of those two parameters are obtained and for each realization the flow and transport equations are solved by the Princeton Transport Code (PTC), a finite element simulator (Babu *et al*., 1993). Finally, different averages of the concentration solutions are calculated to obtain their space-time mean vector and covariance matrix. In this paper, GWMC is parallelized in order to reduce the program execution time.

Methodology

The methodology implemented in this work is as follows:

Step 1. The input files for PTC are generated.

Step 2. The input files for GWMC are generated.

Step 3. A number of realizations of the natural logarithm of the hydraulic conductivity field are generated using the sequential Gaussian simulation (SGSIM) program (Deutsch and Journel 1997). This program generates standard normal simulated values with a given correlation spatial structure on a rectangular mesh.

Step 4. A transformation to get the hydraulic conductivity field (a lognormal field) is calculated for each realization using the Nrm2log program. This program transforms the standard normal values to a normal variable with a given mean and variance and then applies the exponential function to get the lognormal field.

Step 5. For each node at the source of contaminant concentration, the same number of realizations of time series is generated using the RandTS2 program.

Step 6. If the PTC finite element mesh is not equal to the SGSIM rectangular mesh, the SGSIM mesh is mapped into the nearest node of the PTC mesh.