Environmental Modelling & Software 61 (2014) 39–58

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

A parallel computational framework to solve flow and transport in integrated surface–subsurface hydrologic systems

H.-T. Hwang [a, c, *], Y.-J. Park [a, c], E.A. Sudicky [a], P.A. Forsyth [b]

[a] Department of Earth and Environmental Sciences, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1
[b] David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1
[c] Aquanty, Inc., 564 Weber Street North, Unit 12, Waterloo, Ontario, Canada N2L 5C6

ARTICLE INFO

Article history:
Received 29 August 2013
Received in revised form
28 June 2014
Accepted 30 June 2014
Available online

Keywords:
High performance computing
Parallel algorithms
OpenMp
Hydrologic simulation
Jacobian matrix
HydroGeoSphere
BiCGSTAB
Linear iterative solver
Newton method
Integrated surface–subsurface modeling

ABSTRACT

Hydrologic modeling requires the handling of a wide range of highly nonlinear processes from the scale of a hill slope to the continental scale, and thus the computational efficiency of the model becomes a critical issue for water resource management. This work is aimed at implementing and evaluating a flexible parallel computing framework for hydrologic simulations by applying OpenMP in the HydroGeoSphere (HGS) model. HGS is a 3D control-volume finite element model that solves the nonlinear coupled equations describing surface–subsurface water flow, solute migration and energy transport. The computing efficiency of HGS is improved by three parallel computing schemes: 1) parallelization of Jacobian matrix assembly, 2) multi-block node reordering for performing LU solve efficiently, and 3) parameter privatization for reducing memory access latency. Regarding to the accuracy and consistency of the simulation solutions obtained with parallel computing, differences in the solutions are entirely due to use of a finite linear solver iteration tolerance, which produces slightly different solutions which satisfy the convergence tolerance. The maximum difference in the head solution between the serial and parallel simulations is less than 10^{-3} m, using typical convergence tolerances. Using the parallel schemes developed in this work, three key achievements can be summarized: (1) parallelization of a physically-based hydrologic simulator can be performed in a manner that allows the same code to be executed on various shared memory platforms with minimal maintenance; (2) a general, flexible and robust parallel iterative sparse-matrix solver can be implemented in a wide range of numerical models employing either structured or unstructured mesh; and (3) the methodology is flexible, especially for the efficient construction of the coefficient and Jacobian matrices, compared to other parallelized hydrologic models which use parallel library packages.

1. Introduction

With population growth, land-use change and urbanization, pressure on the capacity of water resources to meet societal needs is expected to increase due to rising demands from agricultural, municipal, industrial and environmental uses (Gleick, 1993; WHO, 2000; Gilbert, 2012; UNESCO, 2012; Wu and Tan, 2012). Prediction of the scarcity and need for water resources can be made more reliable by analyzing the distribution and movement of water as well as its quality in the atmospheric, the surface and the subsurface components comprising the hydrologic cycle (Maxwell and Kastenberg, 1999; Kollet and Maxwell, 2006, 2008; Kundzewicz et al., 2007; Yang and Zehnder, 2007; Gruber et al., 2009). As an example of river management, Gilfedder et al. (2012) and Rassam et al. (2013) identified the need to consider groundwater–surface water interactions to analyze the vulnerability of river flow more accurately during a drought period. In order to address such complicated water resource problems, numerical modeling is essential.

Among the numerical simulators developed for water resource analyses, HydroGeoSphere (HGS) is one of several physically-based, fully-integrated surface–subsurface water flow and solute and thermal energy transport models that have emerged in recent years. As Bennett et al. (2013) and Jakeman et al. (2006) indicated, the evaluation and quantification of model performance is a crucial portion during a model development. The HGS model has been verified and validated for its various hydrologic components and has been successfully applied to clarify the roles of various physical

* Corresponding author. Aquanty, Inc., 564 Weber Street North, Unit 12, Waterloo, Ontario, Canada N2L 5C6. Tel.: +1 519 279 1080x123; fax: +1 519 279 1081.
E-mail addresses: hthwang@uwaterloo.ca, naplshwang@gmail.com (H.-T. Hwang).

http://dx.doi.org/10.1016/j.envsoft.2014.06.024
1364-8152/

and chemical processes associated with overland flow and variably-saturated subsurface flow at hill slope, watershed, and even continental scales (Lemieux et al., 2008; Sudicky et al., 2008; Brookfield et al., 2009; Brunner et al., 2009; Goderniaux et al., 2009; Partington et al., 2011; Pérez et al., 2011).

In the subsurface, HGS solves Richards' equation which can be viewed as being derived from a two phase air–water system using the passive air phase approximation. The original system is a mixed parabolic–hyperbolic system and the passive air phase approximation reduces this system to a single highly nonlinear degenerate parabolic equation. This nonlinear parabolic equation results in solutions which can have "shock-like" properties, since this is an approximation to the original parabolic–hyperbolic system. For simulation of surface water flow, the HGS model solves the diffusion-wave approximation of the shallow water equations. Note that this equation is an approximation to the purely hyperbolic shallow water equations and this PDE has been termed "doubly nonlinear" in Santillana and Dawson (2010). This paper also points out that this PDE has properties ranging from Richards' equation to the p-Laplacian. Only recently have methods been developed for the p-Laplacian, based on viscosity solution theory (Oberman, 2013). To summarize, for both subsurface and surface water flows, the PDEs are extremely nonlinear, and exhibit shock-like solutions and these PDEs cannot be considered as behaving like standard linear parabolic equations. If there is one lesson we have learned from feedback from HGS users over many years, it is that robustness and reliability are of paramount importance.

Because HGS requires the handling of a wide range of highly nonlinear processes, computational efficiency has become one of the most critical issues for their application over large spatial and temporal scales (Esposito et al., 2001; Saad, 2003; Park et al., 2008, 2009; Tiedeman and Woodward, 2011; Wang et al., 2011). As a result, there has been a significant amount of research aimed toward making efficient use of high performance computing (HPC) platforms and improving the numerical efficiency of the models. One of the more straightforward ways to adapt HPC for hydrological applications is to utilize pre-developed HPC packages or libraries such as Aztec (Lastovetsky, 2002), DBuilder (Saad and Zhang, 2001), hypre (Falgout and Yang, 2002), KINSOL (Collier et al., 2009), PETSc (Balay et al., 2002). The parallel solver packages implemented in hydrologic simulators are largely based on MPI. For example, parallel hydrologic simulators such as PARSWMS (Hardelauf et al., 2007), ParFlow (Ashby and Falgout, 1996; Jones and Woodward, 2001; Maxwell et al., 2010), PFLOTRAN (Mills et al., 2007), TOUGH2-MP (Zhang et al., 2008), and WASH123D (Cheng et al., 2005) apply parallel preconditioner and solver packages such as PETSc, Azetc, hypre, and DBuilder. Similarly, OpenGeoSys applies a domain decomposition approach, which uses MPI, but not a HPC package, with the Schur-complement-method (Wang et al., 2009). According to Michalakes and Vachharajani (2008), the massive parallelism is highly dependent on the CPU clock speed and communication speeds rather than parallel algorithms. They suggested a fine-grained parallel method for GPUs and demonstrated that the GPU algorithm improved parallel efficiencies for scientific climate modeling. Using MPI, Vivoni et al. (2011) developed a domain partitioning scheme that divides a simulation domain into sub-basins in the order from upstream to downstream. It is noted that MPI-based massively parallel computing is essential for those problems which cannot be solved in a small shared system and the relatively intense effort required for implementation and optimization can be recovered. OpenMP-based parallel computing, on the other hand, can also be valuable for many hydrologic applications because it is relatively easy to implement while reasonable performance improvement can be achieved with more accessible, relatively inexpensive shared memory systems.

According to Amdhal's law (Amdahl, 1967), the potential speedup of an algorithm or a simulator on a parallel computing platform (S_p) is highly dependent on the portion of the code that cannot be parallelized (α) such that $S_p = 1/[(1-\alpha)/p + \alpha]$ and $\lim_{p\to\infty} S_p = 1/\alpha$ where p represents the number of processors. For example, if 95% of the code execution is performed in parallel, the achievable speedup (with zero parallel overhead) is bounded by 20. Although excellent scalability has been demonstrated with massive parallel processing using 100s and 1000s of computing units (Hardelauf et al., 2007; Kollet et al., 2010), note that this is possible only when most of operations (99.9–99.99%) can be performed in parallel. It is understood that excellent scalability in massively parallel computing helps to solve extremely challenging problems with billions of degrees of freedom where the parallel floating point operations become dominant over the remaining sequential portion of the code. However, for the majority of practical applications, only millions of degrees of freedom are required when employing a carefully designed unstructured mesh.

This research is aimed at implementing and evaluating a flexible parallel computational framework for integrated water resources quantity/quality simulation by using OpenMP, which can be utilized on most personal multi-core computers. Users typically want to explore various engineering solutions and perform their analyses with a range of hydraulic and hydrogeological parameters. A numerical method which performs well for a restricted range of parameters, but fails to converge for other input data is problematic in practice. This has led us to the following choice of numerical methods.

- We use unstructured grid, fully implicit, control volume finite element methods. This discretization method has excellent properties (e.g., mass conservation, monotonicity). Unstructured grids are required for modeling complex topography and geology. Fully implicit methods are necessary to avoid exceedingly small time steps.
- The nonlinear discretized equations are solved at each time step using full Newton iteration. Since the construction of the Jacobian is a highly parallel operation, and inexpensive RAM is readily available, we do not believe that it is necessary to use matrix-free methods.
- We use a variant of an incomplete LU (ILU) preconditioner coupled with BiCGSTAB. We have found this method to be very reliable. We will use a multi-block ordering method to facilitate use of multicore architectures, but we attempt to perturb the original ILU as little as possible in order to avoid a large increase in the number of iterations (Magolu monga Made and van der Vorst, 2002). We freely admit that this approach is limited to a relatively small (i.e., 8–64) cores; however, it needs to be noted that developing robust preconditioners which can take advantage of a higher degree of parallelism, is not a solved problem. It is interesting to notice that a combined use of an ILU preconditioner and BiCGSTAB accelerator is a popular choice for the preconditioned Krylov subspace iterative matrix solver in serial hydrologic simulators but the choices for many parallel simulators are different. If an algebraic multigrid preconditioner is used (e.g., Maxwell et al., 2010), the coefficient matrix is required to be a strict M-matrix. Since the coefficient matrices built for hydrologic simulations are not necessarily always M-matrices, some of the nonlinear terms need to be dropped in the matrix construction. This approximation can be clearly disadvantageous in terms of the convergence of the Newton iteration. It cannot be better than the case without requiring that approximation. Therefore, a choice of the pre-conditioner