PARSWMS: A Parallelized Model for Simulating Three-Dimensional Water Flow and Solute Transport in Variably Saturated Soils

H. Hardelauf, M. Javaux, M. Herbst, S. Gottschalk, R. Kasteel, J. Vanderborght,* and H. Vereecken

Three-dimensional vadose zone models are used more and more for solving hydrological problems on a broad range of scales with large amount of nodes. Currently, the problems we can solve in reasonable computational time may have up to $5 \times 10^6$ nodes. However, distributed models may need up to $10^{10}$ nodes to properly predict flow and transport at the watershed scale. The speed and efficiency of current flow and transport models therefore need to be improved. The parallelization of the code is one possible way to decrease the computational time by distributing a complex large geometry problem over multiple processors working in parallel. This is the solution we implemented by developing PARSWMS, a parallelized version of SWMS_3D (Simunek et al., 1995). The objective of this technical note is to describe the PARSWMS model, test its reliability, and show its performance and efficiency compared with single processor runs.

Agrosphere, ICG-IV, Forschungszentrum Jülich GmbH, D-52425 Jülich, Germany. Received 24 Oct. 2006. *Corresponding author (j.vanderborght@fz-juelich.de).

Vadose Zone J. 6:255–259

doi:10.2136/vzj2006.0156

© Soil Science Society of America
677 S. Segoe Rd. Madison, WI 53711 USA.
All rights reserved. No part of this periodical may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher.
reliability, and show its performance and efficiency as compared to single processor runs.

**Model Description**

**SWMS_3D Main Features**

SWMS_3D is a code for simulating water flow and solute transport in three-dimensional, variably saturated media (Simunek et al., 1995). This model was chosen for its flexibility; its performances are ensured by its broad use through the vadose zone hydrology community (e.g., Herbst and Diekkruger, 2002, 2003; Lewandowska et al., 2004; Javaux and VanclOoster, 2006; Javaux et al., 2006). Water flow is solved using Richards’ equation, while transport is described with the CDE. Galerkin-type linear finite element schemes are used for the discretization of the flow and transport equations. The resulting equations are solved in an iterative fashion, by linearization and subsequent Gaussian elimination for banded matrices, a conjugate gradient method for symmetric matrices, or the ORTHOMIN method for asymmetric matrices (Mendoza et al., 1991). Additional measures are taken to improve solution efficiency in transient problems, including automatic time step adjustment and checking if the Courant and Peclet numbers do not exceed preset levels. The water content term is evaluated using the mass-conservative method proposed by Celia et al. (1990). For any additional information about the code, refer to Simunek et al. (1995).

**Overview of the Parallel Implementation**

The PARSWMS code is based on the SWMS_3D code. Most of the subroutines, functions, and variable names have been kept the same. Apart from the parallelization itself, the main change concerns the programming language, which was switched from FORTRAN 77 to C++ to take advantage of the dynamic allocation of all variables. The output file format was switched from FORTRAN 77 to C++ to take advantage of the demand for single processor runs.

The new code is based on Message-Passing Interface (MPI), a library specification for message passing between different processors. Message-Passing Interface provides source-code portability and allows efficient implementation across a range of computer architectures. It is free software for LINUX or UNIX operating systems, which needs to be installed on the working machine.

**Grid Distribution**

The distribution of the flow domain or partitioning of the problem between processors is automatically performed using algorithms of the ParMETIS library. With the aid of this library, even the partitioning process itself is parallelized. ParMETIS is a MPI-based parallel library that implements a variety of algorithms for partitioning unstructured graphs, meshes, and for computing fill-reducing orderings of sparse matrices (http://glaros.dtc.umn.edu/gkhome/views/metis/parmetis/). This library allows dynamic partitioning and an adaptive or irregular grid, and it can be run on heterogeneous clusters. Currently, the partitioning is static, in the sense that it is done once at the beginning of the run. The algorithm optimizes the partitioning between processors based on weights given by the user to each node or element by minimizing the surface of the subvolumes on each processor as well as the connection numbers between processors.

**Matrix Solving**

The solution of the system of linear equations is achieved for the nodes of the sub-volume allocated on each processor. The PETSc (Portable, Extensive Toolkit for Scientific Computation) library (http://www-unix.mcs.anl.gov/petsc/petsc-as/) was used because it allows the user to choose between a large range of solvers and preconditioners that can handle linear and nonlinear problems in parallel mode. Therefore, the solver implemented in PARSWMS is different than SWMS_3D and could easily be further optimized. Herbst et al. (unpublished data) investigated this problem by comparing the efficiency of several preconditioning approaches used in PARSWMS. For water-flow simulations, we used a Jacobi preconditioning method for a parallel Conjugate Gradient solver.

**Methodology**

To check the accuracy and stability of the new code, we first compared PARSWMS with SWMS_3D simulations for a few benchmark scenarios developed for testing one-dimensional flow and transport numerical codes (Vanderborght et al., 2005). Then, we evaluated the performance of the PARSWMS code in terms of computational time by running two computationally demanding problems for different numbers of processors on a massive parallel system.

**Benchmarking Scenarios**

The benchmarking scenarios (S1) considered simple one-dimensional flow processes for simple boundary conditions applied to uniform (loam) or two-layered soils (loam over sand). The problem was discretized into 100 elements of 1-cm thickness each along the z axis. We found that the SWMS_3D code compared favorably with the analytical solutions (results not shown) so that we focused on the comparison between SWMS_3D and PARSWMS. A summary of the scenarios we used is given in Table 1. More detailed information about these scenarios is given by Vanderborght et al. (2005).

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1.1</td>
<td>Infiltration in a uniform soil profile (constant head) for loamy soil</td>
</tr>
<tr>
<td>S1.2</td>
<td>Infiltration in a uniform soil profile (constant flux) for loamy soil</td>
</tr>
<tr>
<td>S1.3</td>
<td>Infiltration in layered soil profiles: loam over sand/sand over loam</td>
</tr>
<tr>
<td>S1.4</td>
<td>Steady-state evaporation from a uniform loamy soil with a groundwater table</td>
</tr>
<tr>
<td>S1.5</td>
<td>Transient evaporation from a uniform loamy soil with mixed surface boundary and coarse (c) or fine (f) soil grid</td>
</tr>
<tr>
<td>S2.1</td>
<td>Solute transport in macroscopically and microscopically heterogeneous monolith</td>
</tr>
<tr>
<td>S3.1</td>
<td>Flow for climatic boundary conditions in micro-heterogeneous soil</td>
</tr>
</tbody>
</table>

Table 1. Benchmark scenario.
Three-Dimensional Flow and Transport Scenarios

Two other scenarios were tested to compare the performances of SWMS_3D and PARSWMS models for more complex three-dimensional cases. The first (S2) concerned solute transport and was derived from a study of Javaux et al. (2006). It consists of a steady-state short pulse release of an inert tracer on the surface of a cylindrical heterogeneous soil monolith. The column geometry is described by a mesh of 492 264 nodes. The heterogeneity of the subsoil core was characterized at two spatial levels. The macrostructure was described by the delineation of three texturally contrasted bodies (sand, clay, and concretions). Additionally, a microvariability was implemented in the sand body with isotropic scaling factor distributions for the van Genuchten–Mualem parameters (Vogel et al., 1991). Table 2 shows the three averaged parameter sets. The scaling factor auto- and cross-variograms were parameterized using a nested model with a short range (range = 0.25 m) exponential behavior and a larger scale (range = 0.9 m) Gaussian model. More details may be found in Javaux et al. (2006).

A second complex scenario (S3) consisted of bare soil submitted to a time series of variable climatic conditions. The description of the upper boundary condition is shown in Fig. 1. The problem geometry consisted in a cube of 2.5-m side length, discretized in 275 706 nodes in total. The nodal distance was 5 cm in the horizontal direction compared with 2.5 cm in the vertical direction, with a spatial refinement in the upper 2.5 cm. This soil cube had a heterogeneous parameter distribution represented by independent random fields of scaling factors for the water retention curve and the hydraulic conductivity function. Averaged soil characteristic curves were modeled using the modified van Genuchten–Mualem model with parameters $\theta_r = 0.078$, $\theta_s = 0.430$, $\theta_m = 0.431$, $\alpha = 0.036 \text{ cm}^{-1}$, $n = 1.56$, $K_s = 10.4 \text{ cm h}^{-1}$, $K_m = 1.04$, and $l = 0.5$, and linear scaling factors were used to account for the spatial variability of the soil hydraulic properties (Vogel et al., 1991).

Time Scaling

We investigated the behavior of PARSWMS in terms of computational time reduction when the number of processors is increased. A massive computer system was used to run the scenarios S2 and S3 on a range between 1 and 256 processors. Details about the technical features of the supercomputer are given in the Appendix.

Results and Discussion

One-Dimensional Benchmarking Scenarios

First results confirm that SWMS_3D and PARSWMS give convergent simulations for all cases, under several benchmark-

![Graph A](image1.png)

![Graph B](image2.png)

![Graph C](image3.png)

**Fig. 1.** Cumulative outflow (upper subplot) and average surface pressure head (middle subplot) resulting from the atmospheric boundary conditions (bottom subplot).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sandy matrix</th>
<th>Clay layer</th>
<th>Stony layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_r$</td>
<td>0.088</td>
<td>0.415</td>
<td>0.015</td>
</tr>
<tr>
<td>$\theta_s$</td>
<td>0.328</td>
<td>0.0091</td>
<td>0.430</td>
</tr>
<tr>
<td>$\theta_m$</td>
<td>0.0194</td>
<td>1.057</td>
<td>8 × 10^{-5}</td>
</tr>
<tr>
<td>$K_s$</td>
<td>3.625</td>
<td>1.65</td>
<td>1.5</td>
</tr>
<tr>
<td>$K_m$</td>
<td>103.7</td>
<td>-3.67</td>
<td>8.64 × 10^{-4}</td>
</tr>
<tr>
<td>$l$</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Table 2. Parameters of van Genuchten–Mualem soil hydraulic functions for the complex solute transport scenario (S2).**
ing scenarios for water. Visual inspection of the data revealed identical results. The computed coefficient of determination $R^2$ between SWMS_3D and the parallelized version are 0.9999468, 0.9999994, 0.9999982, 0.9999913, and 0.9997436 for scenarios S1.1 to S1.5, respectively (Table 1). $R^2$ is always larger than 0.999, which means that the new encoded PARSWMS routines and the new parallelized solver do not generate discrepancies between simulations.

In addition, the PARSWMS simulations for scenarios S2 and S3 (Table 1) compared well with the SWMS_3D results. Figure 2 shows a cross-section of the cylindrical lysimeter concentration distribution during the solute breakthrough, and again, both simulations yield almost identical results.

These tests indicate a successful parallelization and reliability of the new implementation on modeling simple and complex flow and transport cases.

**CPU Time Scaling Behavior**

The speedup of using parallel processors on a problem compared with using only one serial processor is governed by Amdahl’s law (Amdahl, 1967). This law states that if $F$ is the fraction of a calculation that is sequential, and $(1 - F)$ is the fraction that is parallelized, the maximum speedup that can be achieved on $n_p$ processors is

$$\text{speedup} = \frac{1}{F + (1 - F)/n_p}$$

In our case, since roughly 100% of the code is parallelized, speedup versus $n_p$ should optimally be a 1:1 line (or –1 slope line when time gain is plotted in function of $n_p$). Figure 3 shows the time scaling for the scenarios S2 and S3 compared with Amdahl’s law. We plotted the relative time ($t_{n_p}/t_{n_p=1}$) as a function of the number of processors $n_p$ used for the run. The time scaling is almost linear and remarkably close to the optimum defined by Amdahl’s law. There is a short deviation from the linear trend for large numbers of processors, which is likely due to the architecture of the supercomputer, which is composed of a series of nodes having 32 interconnected processors. The internode being lower than the intranode connection speed, doubling the number of processors beyond $2^n$, apparently decreases the code performance.

Fitted linear equations give a slope of –0.93 and –0.7, respectively, for scenarios S2 and S3. Thus, doubling the number of processors results in a computational time decrease of 47.5 and 39%, respectively, while Amdahl’s law predicts an optimum of 50%. The difference in performance between scenarios depends on the boundary conditions and the type of geometry, both of which impact the quantity of information that has to be exchanged between processors. In the S3 case, the complex atmospheric boundary conditions caused a frequent switch between head- or flux-type boundary conditions, which needed more numerical time steps to converge with lower mass balance errors on PARSWMS compared with SWMS_3D.

**Fig. 2.** Comparison of the SWMS_3D (upper row) and PARSWMS (bottom) simulations of solute transport in a heterogeneous monolith (vertical cross-section): scenario S2.
We successfully developed a parallel code based on SWMS_3D for solving Richards’ equation and CDE under variably saturated conditions. The PARSWMS code was developed so that it can be compiled and run on any network (or even a uniprocessor) of LINUX or UNIX workstations or on specialized parallel hardware as long as MPI, PETSc, and PARMETIS free ware is installed. A comprehensive benchmarking was performed to validate the new code. The performances were investigated on a massive parallel computer. The two case studies indicate that doubling the number of processors may lead to a decrease of the computational time up to 48%. However, this figure may be affected by the type of boundary conditions and the geometry of the problem. Thanks to this parallelized algorithm and with better accessibility to cluster calculation and supercomputers, we expect that problems with larger amount of nodes may be tackled in the near future.

Next steps include a coupling with a particle tracking code and with advanced root soil modeling approaches.

Appendix

The scale evolution of computational time performance was performed on the jump supercomputer at the Forschungszentrum Juelich IBM Regatta p690+. Main characteristic features of this machine include the following:

- 41 SMP nodes with 32 processors each (total 1312)
- Processor type: Power4+ 1.7 GHz
- Overall peak performance: 8.9 teraflops
- Main memory: 41 × 128 Gbytes (aggregate 5.2 terabytes), 567 MHz
- Operating system: AIX 5.2
- Operating mode: interactive and batch
- Disk capacity for user data: 8 × 7 × 14 × 72 GB = 56 terabytes
- Disk capacity for system data: 23 terabytes

Acknowledgments

We thank the John von Neumann, Institute for Computing (NIC), Juelich, for providing the opportunity to use their parallel computing resources.

References


