A Description of Improvement in Running Time Strategies for a 3-D Finite Element Seepage/ Groundwater Model using High Performance Computing (HPC) Systems

Fred T. Tracy
US Army Engineer Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180
USA
tracyf@ex1.wes.army.mil.usa

Alex Carrillo
US Army Engineer Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180
USA
carrillo@bugs.wes.army.mil.usa

The running time of a 3-D finite element (FE) seepage/groundwater model written/modified by the authors which approximates unsaturated flow by assuming unconfined flow with a phreatic surface was significantly improved. This paper summarizes the improvements and gives the resulting running times for the following 3-D problems: (1) regional groundwater flow in the vicinity of a contaminant plume, (2) seepage near Cerrillos Dam in Puerto Rico, and (3) a generic aquifer problem containing a river through the middle of the flow domain, three wells, and a cut-off wall. The regional groundwater flow problem is poorly conditioned because of the heterogeneous nature of the medium, the quality of the grid (extensively modified by hand), and the type of finite elements used; whereas the other two problems are well-conditioned. The grid for the Cerrillos Dam problem was generated by a small FORTRAN program, and the grid for the generic aquifer problem was generated using the joining of 16 structured subregions. These 3-D problems form a good variety of problem types. The categories for improvement described are (1) a more efficient bandwidth minimizer for the direct solvers, (2) an improved direct solver, (3) a more efficient use of factorization, (4) the addition of a conjugate gradient method, and (5) improvements to input/output (I/O). The results will be presented for various HPC systems.

1 Introduction

There is significant research regarding the efficiency of different computational techniques used in groundwater modeling and simulation. These results, of course, vary with the number of phases, components, and processes being modeled; the complexity of the site characterization; and the sophistication of the hardware and software used. This paper adds additional data to the discussion by providing the authors' experience in improving the running time of a finite element (FE) code when used to model three real-world problems using various high performance computing (HPC) systems.

2 Model used

A 3-D FE seepage/groundwater model written by Tracy [1] which approximates unsaturated flow by assuming unconfined flow with a phreatic surface was used in this study. As the problems considered have unconfined flow, an iterative solution is required to achieve a steady-state solution.

3 Description of problems

Three problems were used to test the different improvements to the FE program, and a brief description of each follows.

3.1 Regional groundwater flow near a plume

The problem consists of partially confined and partially unconfined groundwater flow in a region where pumping is being done to stop the progress of a contaminant plume. The 2-D triangular mesh used as a starting point to generate the 3-D grid is shown in Figure 1. The nodes with triangles have heads specified as a boundary condition, and the nodes identified with circles are wells where differing amounts of water are being extracted. The immediate purpose of the computation is a calibration where the unknown hydraulic conductivities are adjusted. A slurry trench has been installed in the flow region to modify flow, and a zoom of the grid for this region is shown in Figure 2. The slurry trench is identified by all the nodes along line segment AB. An impervious wall such as a slurry trench is modeled by having different nodes on one side of the wall as compared to the other side.

There are two basic layers which are alluvium and bedrock. The hydraulic conductivities, however, vary greatly within these two broad categories of material. The 3-D grid was generated from the 2-D data and is shown in Figure 3. The 3-D FE program was modified to have a layer specified for each element and hydraulic conductivity specified at each node for that layer. 23,513 nodes and 38,795 elements were used in this problem.

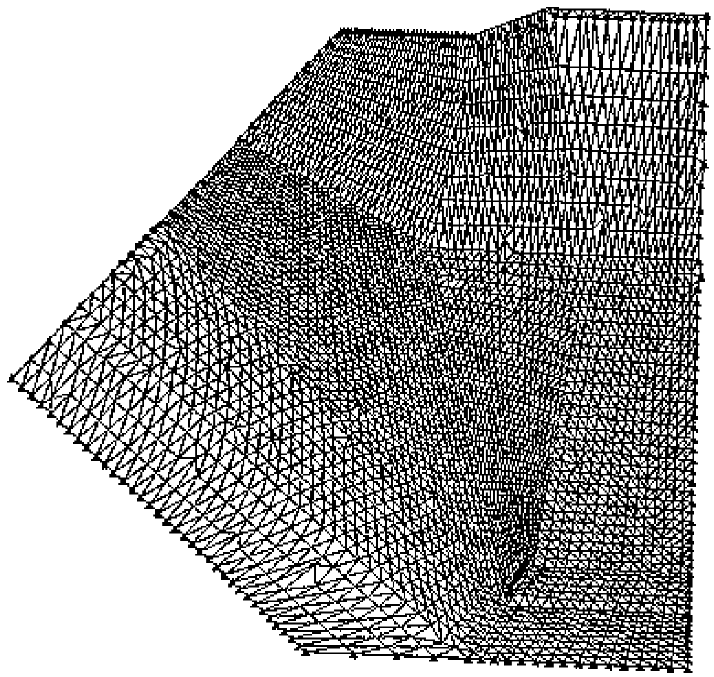

Figure 1. 2-D FE grid

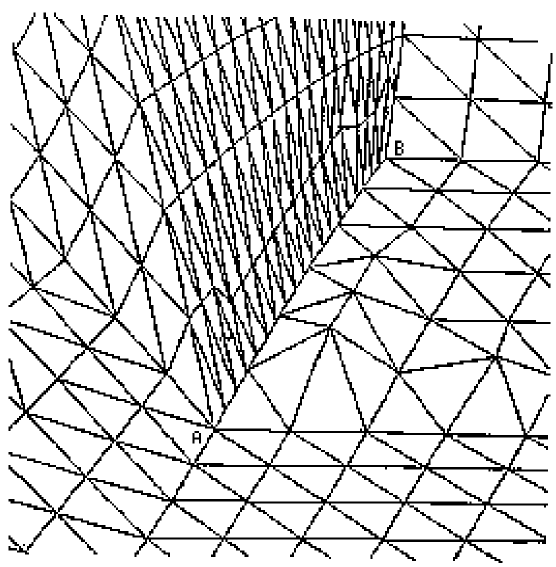

Figure 2. Zoomed area showing slurry trench

3.2 Cerrillos Dam

This is a traditional 3-D study of seepage near Cerrillos Dam in Puerto Rico, and an isometric view of the grid is shown in Figure 4. The flow is unconfined, and both a small grid of 10,915 nodes and 3,469 elements and a large grid of 87,572 nodes and 77,124 elements are used in the study.

3.3 Generic aquifer problem

The problem consists of a part of an aquifer containing a river crossing through the region with three partially penetrating wells and a cut-off wall. The region with the two wells is highly anisotropic, and the region under the river is a rather impervious clay. The region with the cut-off wall has soil properties of a pervious sand. The FE grid as shown in Figure 5 was generated using EAGLE written by Thompson [2] and contains 16 structured subregions. 11,578 nodes and 9,855 elements were used in this study.

4 Computational improvements

The regional groundwater flow problem is poorly conditioned because of the heterogeneous nature of the medium, the quality of the grid (extensively modified by hand), and the type of finite elements used; whereas the other two problems are well-conditioned. The grid for the Cerrillos Dam problem was generated by a small FORTRAN program, and the grid for the generic aquifer problem was generated using a sophisticated grid generation program. These 3-D problems form a good variety of problem types (see Table 1 for more detailed information). A driving factor in the development of the FE model was portability. An out-of-core capability was also highly desirable due to unpredictable memory sizes. Both factors limited the use of vendor-specific math software libraries. A description of the improvements made is now given.

4.1 Nonlinear iteration

The i'th nonlinear iteration to steady-state can be summarized as formulating and solving the following set of Newton iteration equations:

$$\begin{aligned} [K]^i \{\Delta\phi\}^i &= -\{Q\}^i \\ \{\phi\}^{i+1} &= \{\phi\}^i + \{\Delta\phi\}^1 \end{aligned} \qquad (1)$$

where [K] is the stiffness matrix, $\{\Delta\phi\}$ is the change in total head or potential vector, and {Q} is the residual flow vector. Because the solution is steady-state, a significant savings can be achieved by only updating $[K]^i$ with respect to changing boundary conditions, so this is what is done. $\{Q\}^i$ must, however,

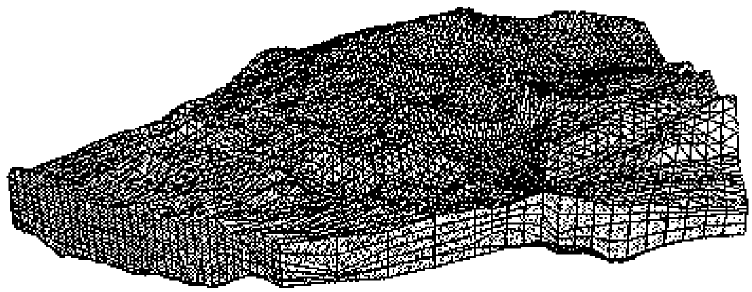

Figure 3. 3-D grid

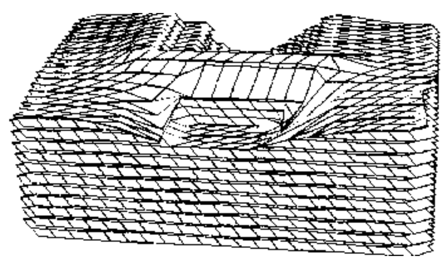

Figure 4. Plan view of aquifer

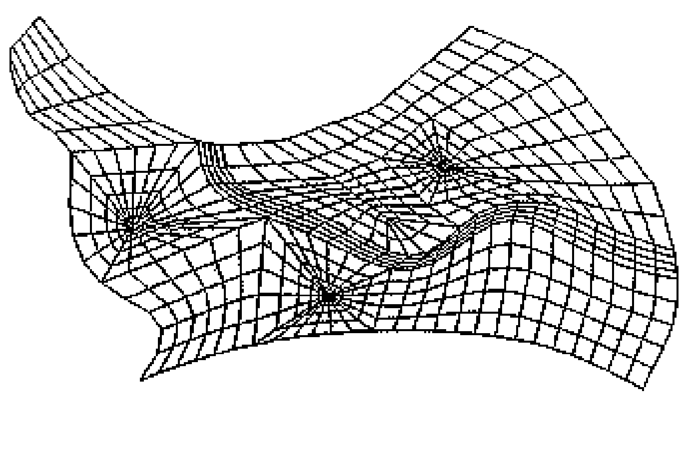

Figure 5. Plan view of grid

Table 1. Model information

	Plume	Aquifer	S Cerrillos	L Cerrillos
Nodes	23,513	11,578	10,915	87,572
Elements	38,796	9,855	8,469	77,124
Iterations	48	5	8	9
Original global BW	883	1,594	502	1,974
Original aver. local BW	364	1,075	467	1,855
New global BW	482	675	383	1,481
New aver. local BW	356	436	283	1,092

be computed using the current values of the element stiffness matrices at each iteration.

4.2 Solution process

Improvements to the solution algorithm branched into two directions. First, a more efficient block Cholesky factorization solver was developed and implemented using a LAPACK [3] library subroutine as a template . Minor modifications were made to take advantage of the smaller local bandwidth of each block, and an out-of-core capability was built around the routine. For an unchanging stiffness matrix between nonlinear iterations, re-use of the factorization from the previous iteration greatly reduced the computations required for some problems. Coupled with a more sophisticated bandwidth minimization routine from Gibbs [4], significant time reductions were obtained for the direct solution process. Table 1 gives bandwidth (BW) data for the three problems.

Secondly, a conjugate gradient solver developed by Kincaid [5] was added, which further reduced memory requirements and eliminated the need for an out-of-core solver. A conjugate gradient routine from the Cray Research, Inc. Scientific Software Library (Scilib) was also tested.

4.3 Input/output (I/O)

I/O was closely evaluated since it could easily account for a substantial portion of the run time. First, the stiffness matrix was switched from a banded storage format to a sparse matrix format, which allowed assembly and boundary condition modifications to be completed in-core and eliminated many of the inefficiencies associated with the out-of-core solver. The reduced bandwidths also decreased the I/O requirements for the out-of-core solver, making the use of faster disks more feasible. Finally, the iterative solver eliminated the need for out-of-core I/O.

4.4 General improvements

Enhanced vectorization and other improvements in coding were also accomplished. Through effective use of arrays, redundant work was eliminated, and several functions were transformed into level 3 BLAS routines [6].

4.5 Results

Table 2 shows the CPU times in seconds for the CRAY C90, CRAY YMP, and SGI Power Challenge Array (PCA) in single processor mode.

4.6 Multi-tasking

A fine-grained parallel version of the FE model has been created. Since the restructured code relies heavily on vendor-tuned and multi-tasked BLAS routines, by adding a few parallel directives, good parallel performance was achieved for the large Cerrillos Dam problem on the CRAY architectures. Using 16 dedicated processors on the CRAY C90, 8.83 Gflops were sustained, reducing overall run time from 1380 sec to 132 sec. The other two problems were too small to fully exploit the multi-tasking capabilities of the CRAY C90. Further work in developing a more coarse-grained parallel model using domain decomposition and some of the more recently developed parallel iterative methods could further improve the performance on both shared memory and.

Table 2. Single processor CPU times (sec)

	Plume	Aquifer	S Cerrillos	L Cerrillos
Original code - YMP	12,548	3,747	517	33,148
New direct solution				
CRAY C90	74	12	18	1,380
CRAY YMP	158	29	43	3,890
Silicon Graphics PCA	372	54	78	7,366
Public domain iterative solver				
CRAY C90	Non conv.	85	60	961
CRAY YMP	Non conv.	130	93	1,497
Silicon Graphics PCA	8,226	79	56	1,553

Cray Research, Inc. Scilib iterative solver				
CRAY C90 CRAY YMP	571 879	18 27	11 17	141 217

distributed memory architectures

Acknowledgment

This work was supported in part by a grant of HPC time from the DoD HPC Major Shared Resource Center at the US Army Engineer Waterways Experiment Station (WES). Other funding was received from the WES In-house Laboratory Independent Research (ILIR) and Groundwater Modeling programs. The use of specific vendor computer hardware and software in this paper does not constitute an endorsement. The Chief of Engineers has approved the publication of this paper.

Key words

Groundwater modeling, finite element method, high performance computing

References

1. Tracy, Fred T. (1991). Application of Finite Element, Grid Generation, and Scientific Visualization Techniques to 2-D and 3-D Seepage and Groundwater Modeling. Technical Report ITL-91-3. US Army Engineer Waterways Experiment Station, Vicksburg, MS.

2. Thompson, J. F. (1987). A Composite Grid Generation Code for General 3-D Region. American Institute of Aeronautics and Astronautics 25th Aerospace Science Meeting. Reno, NV.

3. Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1990). LAPACK: A Portable Linear Algebra Library for High-Performance Computers. *Supercomputing '90*, pp. 2-11. Los Alamitos, CA.

4. Gibbs, N. E., Poole, W. G., Jr., and Stockmeyer, P. K. (1976). An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix. *SIAM Journal of Numerical Analysis*, 13 (2), pp. 235-251.

5. Kincaid, David R., Respess, John R., Young, David M., and Grimes, Roger G. (1996). ITPACK 2C: A FORTRAN Package for Solving Large Sparse Linear Systems by Adaptive Accelerated Iterative Methods. Center for

Numerical Analysis, University of Texas, Austin, TX.

6. Dongarra, J., Du Croz, J., Hammarling, S., and Duff, I. (1990). A Set of Level 3 Basic Linear Algebra Subprograms. *ACM Transactions of Mathematical Software*, 16(1), pp. 1-17.