LBL-35518
UC-400

T2CG1, A Package of Preconditioned Conjugate Gradient Solvers for TOUGH2

G. Moridis, K. Pruess, and E. Antúnez

Earth Sciences Division
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

March 1994

This work was supported, in part, by the Director, Office of Civilian Radioactive Waste Management, Yucca Mountain Site Characterization Project Office, Regulatory and Site Evaluation Division, by the WIPP Project, Sandia National Laboratories, under Document Number 129847, and by the Assistant Secretary for Energy Efficiency and Renewable Energy, Geothermal Division, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

T2CG1, A PACKAGE OF PRECONDITIONED CONJUGATE GRADIENT SOLVERS FOR TOUGH2

G. Moridis, K. Pruess, and E. Antúnez

Earth Sciences Division, Lawrence Berkeley Laboratory
University of California, Berkeley, California 94720

ABSTRACT

T2CG1, a package of preconditioned conjugate gradient solvers, has been added to TOUGH2 to complement its direct solver and significantly increase the size of problems tractable on PCs. T2CG1 includes three different solvers: a Bi-Conjugate Gradient (BCG) solver, a Bi-Conjugate Gradient Squared (BCGS) solver, and a Generalized Minimum Residual (GMRES) solver. Results from six test problems with up to 30,000 equations show that T2CG1 (1) is significantly (and invariably) faster and requires far less memory than the MA28 direct solver, (2) it makes possible the solution of very large three-dimensional problems on PCs, and (3) that the BCGS solver is the fastest of the three in the tested problems.

INTRODUCTION

Most of the computational work in the numerical simulation of fluid and heat flows in permeable media arises in the solution of large systems of linear equations. The simplest technique for solving such equations is by direct methods. However, because of large storage requirements and accumulation of roundoff errors, the application of direct solution techniques is limited, depending on matrix bandwidth, to systems of a few hundred to at most a few thousand simultaneous equations.

The matrices arising in fluid and heat flow simulations are generally sparse, i.e., typically only a few percent of matrix elements are non-zero. Matrix bandwidth and "infill" during direct solution increase dramatically with the dimensionality of the flow problem.

This limits direct matrix methods to problems with a few hundred grid blocks in 3-D, while one-dimensional calculations are feasible up to a few thousand grid blocks.

An attractive alternative for large linear systems is provided by iterative matrix methods [*Varga*, 1962]. Conventional iterative methods converge only for diagonally dominant matrices, but conjugate gradient methods have no such limitation [*Hestenes and Stiefel*, 1952]. We have added a package of preconditioned conjugate gradient solvers to TOUGH2 [*Pruess*, 1991] to complement its direct solver and significantly increase the size of problems tractable on personal computers. The conjugate gradient solvers decrease execution time and memory requirements substantially, and make possible the simulation of three-dimensional flow problems with of the order of 10,000 grid blocks on workstations and PCs.

This report briefly summarizes the selective adaptation of an off-the-shelf conjugate gradient package to TOUGH2, and presents applications to a variety of fluid and heat flow problems. A more complete documentation of the conjugate gradient package is in preparation, which will include a full suite of sample problems, and a diskette with TOUGH2 input files and code enhancements.

CONJUGATE GRADIENT PACKAGE

T2CG1 was derived from SLAP Version 2.0 [*Seager*, 1988], a package developed for the solution of large sparse linear $N \times N$ systems

$$\mathbf{A} \bullet \mathbf{x} = \mathbf{b} \qquad (1)$$

where N is the order of the $\mathbf{A}$ matrix. SLAP is a collection of various conjugate gradient solvers, which come with two possible matrix preconditioning options: diagonal scaling (DS) and modified incomplete LU factorization (ILU).

In TOUGH2 the matrix $\mathbf{A}$ is a Jacobian with certain consistent characteristics. In systems with regular geometry, $\mathbf{A}$ has a known block structure with well defined sparsity

patterns. In general, A matrices arising in TOUGH2 simulations are non-symmetric M-matrices with typically no diagonal dominance. Although A can be positive definite in regular systems with homogeneous property distributions, it usually is not, and ill-conditioning is expected in realistic heterogeneous large-scale simulations. Due to the fact that A is a Jacobian, the elements of A in a single row often vary by several orders of magnitude. In TOUGH2 simulations of flow and transport through fractured media, the implementation of the "multiple interacting continua" concept results in a large number of zeroes on the main diagonal of A, making pivoting impossible and resulting in very ill-conditioned matrices. It is evident that TOUGH2 simulations create matrices which are among the most challenging, with all the features that cause most iterative techniques to fail. In addition, the general-purpose nature of TOUGH2 means that different matrix characteristics may arise for different types of problems. This explains the past heavy reliance of TOUGH2 on the direct solver Ma28 [*Duff*, 1977].

Extensive testing of the SLAP package in a variety of flow and transport problems identified the most promising conjugate gradient methods. The properties of the A matrix essentially precluded the use of DS preconditioning, a fact which was confirmed in the process of testing SLAP. Without exception, ILU preconditioning was far more effective and often the only possible option. Of the 15 methods available in SLAP, three were identified as the ones with the most potential. In order of increasing robustness, these were the (1) *Bi-Conjugate Gradient* (BCG) method, (2) the Lanczos-type *Bi-Conjugate Gradient Squared* (BCGS) method, and (3) the *Generalized Minimum Residual* (GMRES) method. In terms of the SLAP terminology, these methods corresponded to the subroutines DSLUBC, DSLUCS, and DSLUGM, respectively.

Fletcher [1976] proposed BCG for the solution of linear, but not necessarily positive definite or symmetric systems. Theoretical analysis of the properties of BCG indicates that as long as the recurrences in the method do not break down, it must terminate in $m < N$

iterations. Although there is no guarantee of reduction of the quadratic functionals (i.e. that the recurrences will not break down or become unstable), in practice this is rare [*Press et al.*, 1986]. If a good preconditioner is used, BCG is an effective method [*Seager*, 1988].

The BCGS [*Sonneveld*, 1989] method is related to the BCG, but it does not involve adjoint matrix-vector multiplications, requires half the computational work, and the expected convergence rate is about twice that of BCG. For a $N \times N$ problem, BCGS was theoretically shown to terminate in at most N steps. *Seager* [1988] reports that when BCG diverges, BCGS diverges twice as fast, and when BCG stagnates, BCGS is more likely to diverge. He also suggested using BCGS after first successfully applying BCG. However, in most TOUGH2 applications, we did not observe similar behavior. We observed a non-monotonic reduction in the error of BCGS, with many (and sometimes significant) local peaks in the convergence performance. These local peaks are also observed in BCG, but they are usually smaller in magnitude.

The GMRES method of *Saad and Schultz* [1986] is a Lanczos-type extension of conjugate gradients for general non-symmetric systems which is expected to converge in $m < N$ steps for any non-singular matrix if truncation errors are not considered. It generates an orthonormal basis from the Krylov subspace

$$\mathrm{K}(m) = span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \mathbf{A}^3\mathbf{r}_0, \ldots, \mathbf{A}^{m-1}\mathbf{r}_0\} \qquad (2)$$

where $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{r}_0$ is the initial residual. Since storage requirements increase with m and the number of multiplications with m^2, m has to be much smaller than N. If the convergence criterion is not met within m iterations, the iteration can be restarted using as a starting value of x the one obtained at the m-th iteration of the previous cycle. The GMRES we used employs this approach. We found that a m=20 to 30 is needed in most TOUGH2 simulations. For m<15 we generally obtain unsatisfactory performance, and it is usually pointless to use m>35 (since this probably indicates that GMRES may not be a

good method in that particular problem). A unique feature of GMRES is that the residual norm is diminished at every iteration, i.e. the error decreases monotonically.

In the T2CG1 package we maintained the nomenclature of SLAP, but substantially modified the structure and content of the subroutines. We eliminated most subroutines used in the SLAP structure and reprogrammed large segments of the code to take advantage of the well-defined sparsity pattern of matrix A. This resulted in a compact code optimized for TOUGH2, which is substantially faster, but which lacks the modular structure of SLAP. The native TOUGH2 mode uses a matrix storage scheme which is identical to the SLAP Triad Matrix Storage Format, which was maintained unaltered in T2CG1. The ILU preconditioner was maintained for use in simulations with irregular geometry. However, for simulations with regular geometry we made use of the known structure of the A matrix (determined by the integrated finite difference formulation of TOUGH2) to develop an optimized *Incomplete Block LU* factorization (IBLU) preconditioner [*Sonneveld,* 1989]. The IBLU preconditioner was based on an approach proposed by *Meyerink* [1983], and significantly sped up the convergence rate of the three methods compared to the ILU. Moreover, we confirmed Sonneveld's observation that the IBLU factorization has the additional advantage of being less sensitive to special directions in the problem (e.g. the advection direction in the advection-diffusion equation, layering, etc.).

Storage requirements in T2CG1 remained the same as in SLAP and are described in detail in *Seager* [1988]. DSLUBC and DSLUCS have the same storage requirements, while DSLUGM needs several times more memory. In terms of speed, our experience in a large number of TOUGH2 simulations indicated that DSLUCS is the fastest by a substantial margin, followed by BCG. DSLUGM was the slowest, but also the most robust, and managed to solve efficiently some of the most demanding problems. Contrary to Seager's observations, DSLUCS was the second most robust. Although one or two methods in the T2CG1 package occasionally failed to converge successfully, we have not encountered a case where all three methods failed.

SAMPLE PROBLEMS

REPOSITORY PERFORMANCE ASSESSMENT AT YUCCA MOUNTAIN

This is a two-dimensional radially symmetric model that represents, in a schematic way, alternating layers of fractured-porous (welded) and porous (non-welded) tuffs (see Fig. 1). The flow domain extends from the land surface to the water table and has a radius of 5000 m. It consists of 630 grid blocks with 1209 connections between them. With the EOS4 fluid property module a total of 1890 simultaneous equations have to be solved. The repository is modeled as a circular disk of 1500 m radius. Heterogeneity is moderately strong, with permeability contrast between different layers of up to 10^4. The system is initialized with gravity-capillary equilibrium at zero net infiltration, and the response to repository heating is simulated. Special features include effective continuum treatment for the fractured units, and strong vapor pressure lowering effects from formation drying. Full problem specifications and discussion of simulated system behavior are given in *Pruess and Tsang* [1993] and *Tsang and Pruess* [1990].

The problem was run on an IBM RS/6000 workstation, using 64-bit arithmetic. Execution time for 10 time steps, corresponding to a simulated time of 6.48 days, was 693.5 seconds for the MA28 direct solver [*Duff*, 1977]. The same simulation required 456.5 seconds with the DSLUCS iterative linear equation solution. For the specified convergence criterion of 10^{-6} only 3-5 iterations were needed for each equation solution.

CHANNELIZED WATER FLOW AT YUCCA MOUNTAIN

Among the concerns being addressed in site suitability studies for Yucca Mountain is the possibility of rapid channelized water flow along fast paths. Such fast paths may arise from the heterogeneity within individual fractures as well as fracture networks. Effects such as capillary imbibition into the rock matrix and vaporization from radioactive decay heat would tend to diminish channelized water flow in fractures. We have set up several models to examine the conditions under which liquid water flow may serve as a pathway

for contaminants. The model discussed here is a three-dimensional X-Y-Z model with 6 x 8 x 21 = 1008 grid blocks and 2682 connections between them. It represents a one-fourth symmetry element of the area of a single waste package in an idealized vertical emplacement configuration as shown in Fig. 2. The gridding in the vertical (Z) direction is identical to the previous R-Z model. The first Y-gridding has a width of 1 mm and represents a fracture with a high permeability of 9×10^{-12} m^2. The issue to be addressed by the model is whether vapor generated near the waste packages can be discharged into fractures and then condense at some distance from the waste packages in a sufficiently focused manner to cause rapid and persistent downflow of water past the repository horizon. With the Topopah Spring matrix rock assigned a permeability of 1.9×10^{-18} m^2, heterogeneity is rather strong with a maximum permeability contrast of approximately 5×10^6.

Using the EOS4 fluid property module, 3x1008 = 3024 simultaneous equations are to be solved. The simulations were again carried out in 64-bit arithmetic on an IBM RS/6000 workstation. With the DSLUCS iterative solver, a run to 10 years of simulated time required 36 time steps and 4603.9 CPU-seconds. The linear equation solution with a specified convergence tolerance of 10^{-6} required an average of 40 iterations. MA28 failed for this problem. After 2 hours of CPU time it had not yet completed a single linear equation solution whereupon the run was terminated. The failure of MA28 occurred in spite of the fact that very large memory allocations were made for the problem-size dependent arrays, which would in fact have been sufficient for 10,000 grid blocks and 24,000 connections with the iterative solvers.

ENVIRONMENTAL REMEDIATION

This is a modified version of a problem developed by *S. Webb* [private communication, 1993] to study the TEVES (Thermal Enhanced Vapor Extraction System) process being designed and built at Sandia National Laboratories. In this process the ground is electrically heated, and borehole(s) at the center of the heated zone are maintained

at a vacuum to draw air and vaporized contaminants into the borehole and to a subsequent treatment facility. The ground above the heated zone and beyond is insulated to minimize heat loss to the environment. A vapor barrier is also used over a larger area to provide a more complete air sweep of the contaminated soil. The simulated domain consists of 1300 elements in a three-dimensional grid. The problem is to be run with the EOS3 fluid property module for water, air, and heat, resulting in a system of 3,900 simultaneous equations. Repeated attempts to solve this problem using the MA28 direct solver failed because of insufficient memory. We allowed a maximum of 10 time steps, and tested two of the three conjugate gradient methods, DSLUCS and DSLUGM.

Both methods performed remarkably well. On a Macintosh Quadra 800 microcomputer, DSLUCS required 658.12 seconds (109.08 for input and 549.03 for calculations) when the convergence criterion was set to 10^{-10}, and needed a minimum of 7 and a maximum of 23 iterations for each matrix solution. This compares very favorably with the total number of equations $N = 3900$, which is equal to the maximum number of iterations for convergence.

For a convergence criterion of 10^{-8}, DSLUCS required 619.57 seconds (108.64 for input and 510.94 for calculations) and needed a minimum of 6 and a maximum of 17 iterations for each matrix solution. DSLUGM also performed very well, but was slower than DSLUCS. It required 703.69 seconds (108.17 for input, 595.52 for calculations) to reach the closure criterion of 10^{-10}, and reached convergence after a minimum of 12 and a maximum of 32 iterations for each matrix solution. It must be noted that the calculation times in this and all subsequent examples include a significant amount of overhead (e.g. the time needed for the initial set-up, to write voluminous results in the output files, etc.). Therefore, an even better performance is expected in longer runs.

VERTICAL SECTION OF WIPP

The WIPP (Waste Isolation Pilot Plant) is a planned repository for defense nuclear wastes in a bedded salt formation near Carlsbad, New Mexico. The present simulation problem as designed by *S. Webb* [private communication, 1993] includes a preliminary model of the repository and the surrounding detailed stratigraphy with explicit representation of the various layers of pure halite, argillaceous halite, polyhalitic halite, and anhydrite. The purpose of the model is to evaluate effects of gas generation and two-phase flow on repository performance within a complex stratigraphy, and to compare with other models that use a simplified representation of the stratigraphy.

The simulated domain consists of 1200 elements in a two-dimensional vertical section grid. This is an isothermal two-phase flow problem to be run with EOS3 (water, air), and results in a system of 2,400 simultaneous equations. Permeabilities in the problem were stratified, generally very low, with extremely high permeability contrast in the vicinity of the more permeable repository. This type of problem is among the most challenging for iterative solvers because elements of the Jacobian matrix along the same row may differ by many orders of magnitude. We tested the direct solver MA28 and two of the three conjugate gradient methods, DSLUCS and DSLUGM, in 10 time step runs on a Macintosh Quadra 800 microcomputer.

MA28 was able to solve the problem and required 531.47 seconds (448.87 for calculations, 82.60 for input). Both conjugate gradient methods significantly outperformed the direct solver. DSLUCS concluded the 10-time step simulation in 355.42 seconds (272.78 for calculations, 82.63 for input), and reached the specified convergence criterion of 10^{-10} in a minimum of 7 and a maximum of 11 iterations for each matrix solution. When the convergence criterion was increased to 10^{-8}, 346.62 seconds were needed (263.53 for calculations, 83.08 for input) and the number of iterations per matrix solution varied between 6 and 10. The performance of DSLUGM for a closure criterion of 10^{-8} was very

similar: 354.38 seconds were required (272.25 for calculations, 82.13 for input), and 7 to 12 iterations were needed per matrix solution. Comparing calculation times only (and disregarding the fact that these include the substantial times spent on overhead), the conjugate gradient solvers are about 1.7 times faster than the direct solver. It is expected that the relative speed will increase in longer runs.

THREE-DIMENSIONAL GEOTHERMAL RESERVOIR MODEL

Five three-dimensional simulation models with different discretization were constructed [*Antúnez et al.*, 1994]. The different discretizations ranged between 500 and 10,000 elements and resulted in (a) 1,000 to 20,000 equations in single-phase systems and (b) 1,500 to 30,000 equations in two-phase systems. The simulation models have an areal extent of 5 x 4 km (20 km^2) and a thickness of 1000 m, divided in ten layers of 100 m each (Fig. 3). The same discretization of the vertical reservoir dimension was maintained in all cases, but finer discretizations were used in the X and Y directions. All grids have a well producing at a constant rate of 30 kg/s in the sixth layer, an injection well operating at a rate of 30 kg/s in the third layer, and a 30 MW heat source at the bottom layer (layer 10). The wells are located at the node of the element closer to the points (500, 500, 550) for the producer and (4500, 3500, 250) for the injector. The heat source is distributed among the required elements to cover an area of $4x10^5$ m^2 (1000 m in x and 400 m in y) at the center of bottom layer (Fig. 3). All of the models were used to perform simulations for single-phase and two-phase conditions.

For the single-phase cases the initial conditions are 40 MPa and 280°C in all blocks; for the two-phase cases, 10 MPa and S_g=0.20 in all blocks. No-flow boundaries to mass and heat are employed. Relative permeabilities correspond to Corey's curves with residual saturations of liquid and steam equal to 0.3 and 0.05, respectively. Capillary pressures are neglected.

CERRO PRIETO GEOTHERMAL FIELD

The Cerro Prieto geothermal field developed by the Comisión Federal de Electricidad (CFE), is located approximately 35 km south of Mexicali, Baja California, México. Since the beginning of the exploitation of Cerro Prieto in 1973, one of the most important operational problems that CFE has had to face was the handling of the waste brine [*Hiriart and Gutiérrez Puente*, 1992]. Up to date most of the brine is sent to evaporation ponds that presently cover an area of 18.6 km^2, Figure 4. An infiltration area west of the ponds is used during the winter, when the evaporation rate is lower.

Recently (1992-93), CFE started a series of cold brine (approximately at 20°C) injection tests, using brine from the evaporation ponds. The objective of these tests was to monitor the reservoir's response to the injection and to test the injectivity of different areas of CP1 in the western part of the field. Under the DOE/CFE cooperative agreement on geothermal energy, a numerical model for CP1 was developed, using data provided by CFE. The computational grid, covering an area of 89 km^2, was defined based on the geological model of the field and the location and completion of the production and injection wells (Fig. 4).

In the vertical direction the model extends from the surface to 5,000 m depth, and is divided into six layers. All the layers have the same discretization and have 235 grid elements (Fig. 4), except layer five that has 47 additional blocks in the NE simulating the volume of the CP2, CP3 and CP4 areas. The numerical model has a total of 1,411 elements (resulting in 4233 simultaneous equations) and was developed as a single porosity model [*Antúnez and Lippmann*, 1993]. Finer discretizations in the vertical direction resulted in discretizations of 5,644 to 8,466 elements and 16,932 to 25,398 equations. The model was calibrated with production and piezometric data, and was used to test several injection strategies.

For the Cerro Prieto model, the timing of the Newtonian iterations was conducted using the following scenario: Inject 3,500 t/h of 20°C brine evenly distributed between injection wells M-48, 101, 104, E-6, O-473 and M-6. Production wells will continue producing at a rate equal to that measured at the end of 1991 (for that year, the average field production was 5,459 t/h of steam and 6,394 t/h of separated brine). Injection well locations are shown in Figure 4.

Table 1 presents a summary of the results of testing the different solvers [*Antunez et al.*, 1994]. Case 1 and 2 correspond to the Cartesian models for single and two-phase conditions. Cases 3 and 4 are for the two-phase conditions using an irregular grid with single and double porosity. The reported total number of iterations are the sum of: a) the Newtonian iterations (external iterations); b) the repeated external iterations due to convergence failure (after 9 Newtonian iterations without reaching convergence, the incremental time used in the current time step is divided by five and the iteration procedure for that time step is repeated); and c) the convergence iterations (iterations that do not need to call the solver since convergence has been attained) one per prescribed time step. The average timing per Newtonian iteration only includes the completed Newtonian iterations; convergence iterations are not considered in this column. The CPU time corresponds to execution time for all iterations Newtonian and non-Newtonian, plus the time to write the output files.

Time comparisons for the different cases indicates that conjugate gradients outperformed the direct solver in all the cases where MA28 could be applied. The BCGS solver was consistently the fastest method, followed by the BCG solver. The GMRES method does not seem to be a good choice for the geothermal systems tested, as it was often slower than the direct solver MA28. However, due to significantly lower memory requirements, the GMRES method could be used for the solution of large three-dimensional problems intractable with MA28. In terms of CPU time, BCGS was between 1.5 to 3.2 faster than MA28 in the smaller three-dimensional cartesian grid problems (up to

2,000 elements and 4,000 equations). Due to very large memory requirements, MA28 could not be used in larger problems, in which BCGS was the only method used.

However, it is important to emphasize that iterative methods are problem-specific. A solver that performs well in a given problem is not guaranteed to work with all problems. The GMRES solver was found to be the slowest of the three tested conjugate gradient solvers but in previous testing of some highly heterogeneous fluid and heat flow problems it was the only one that could converge. Preliminary testing of the CG solvers in a specific problem is strongly recommended in order to select the best for the task.

CONCLUSIONS

A suite of preconditioned conjugate gradient solvers has been implemented in TOUGH2, considerably enhancing the size of tractable problems. On PCs, microcomputers and workstations two and three-dimensional flow problems can be run with as many as 10,000 or grid blocks or more. This compares with problem size limits of a few thousand grid blocks (for 2-D) when using the MA28 direct solver , and a few hundred grid blocks for 3-D problems. Memory requirements and execution times of the conjugate gradient routines are modest, increasing only approximately linearly with problem size.

ACKNOWLEDGEMENT.

Thanks are due to Steve Webb of Sandia National Laboratories, Albuquerque, NM, for making available to us his TOUGH2 input files for several multiphase flow problems, and for helpful discussions. This work was supported, in part, by the Director, Office of Civilian Radioactive Waste Management, Yucca Mountain Site Characterization Project Office, Regulatory and Site Evaluation Division, by the WIPP Project, Sandia National Laboratories, under Document Number 129847, and by the Assistant Secretary for Energy Efficiency and Renewable Energy, Geothermal Division, of the U.S. Department of

Energy under Contract No. DE-AC03-76SF00098. Drs. J. Wang and C. Oldenburg are thanked for their helpful review comments.

REFERENCES

Antúnez, E., and Lippmann, M. (1993) Numerical Study of the Effects of Brine Injection on the CP1 Production Area of Cerro Prieto. Earth Sciences Division Annual Report 1992, Lawrence Berkeley Laboratory report LBID-33000, pp. 97-99.

Antúnez, A., G. Moridis and K. Pruess. Large-Scale Geothermal Reservoir Simulation on PCs (1994), Lawrence Berkeley Laboratory Report LBL-35192, presented at 19th Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, CA, January 1994.

Duff, I. S. (1977) MA28 - A set of Fortran Subroutines for Sparse Unsymmetric Linear Equations, AERE Harwell Report R 8730.

Fletcher, R. (1976) Conjugate gradient methods for indefinite systems. Numerical Analysis, Lecture Notes in Mathematics 506, Springer-Verlag, New York. pp. 73-89.

Hestenes, M.R. and E. Stiefel (1952). Methods of Conjugate Gradients for Solving Linear Systems, *J. Res. Nat. Bur. Standards*, Vol. 49, No. 6, pp. 409 - 436.

Hiriart, G.L. and Gutiérrez, H.P. (1992) An Update of Cerro Prieto Geothermal Field, Twenty Years of Commercial Power. Geothermal Resources Council Bull. Sept/Oct, pp. 289-294.

Meyerink, J.A. (1983) Iterative methods for the solution of linear equations based on incomplete block factorization of the matrix. Paper SPE 12262, Proc. 7th SPE Symposium on Reservoir Simulation, San Francisco, CA, November 15-18, pp. 297-304.

Press, W.K., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling (1986) *Numerical Recipes, The Art of Scientific Computing (FORTRAN Version)*, Cambridge University Press, Cambridge, MA.

Pruess, K. (1991) *TOUGH2* - A General-Purpose Numerical Simulator for Multiphase Fluid and Heat Flow, Lawrence Berkeley Laboratory report LBL-29400.

Pruess, K. and Y. Tsang. Modeling of Strongly Heat-Driven Flow Processes at a Potential High-Level Nuclear Waste Repository at Yucca Mountain, Nevada, Lawrence Berkeley Laboratory Report LBL-33597, presented at International High Level Radioactive Waste Management Conference, Las Vegas, NV , April 26-30, 1993.

Saad, Y. and Schultz, M.H. (1986) GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., Vol. 7, No. 3, pp. 856-869.

Seager, M.K. (1988) A SLAP for the Masses. Paper presented at "Methods & Algorithms for PDE's on Advanced Processors", Austin, TX, Oct 17-18, 1988, Lawrence Livermore National Laboratory report UCRL-100195.

Sonneveld, P. (1989) CGS, A fast Lanczos-type solver for nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., Vol. 10, No. 1, pp. 36-52.

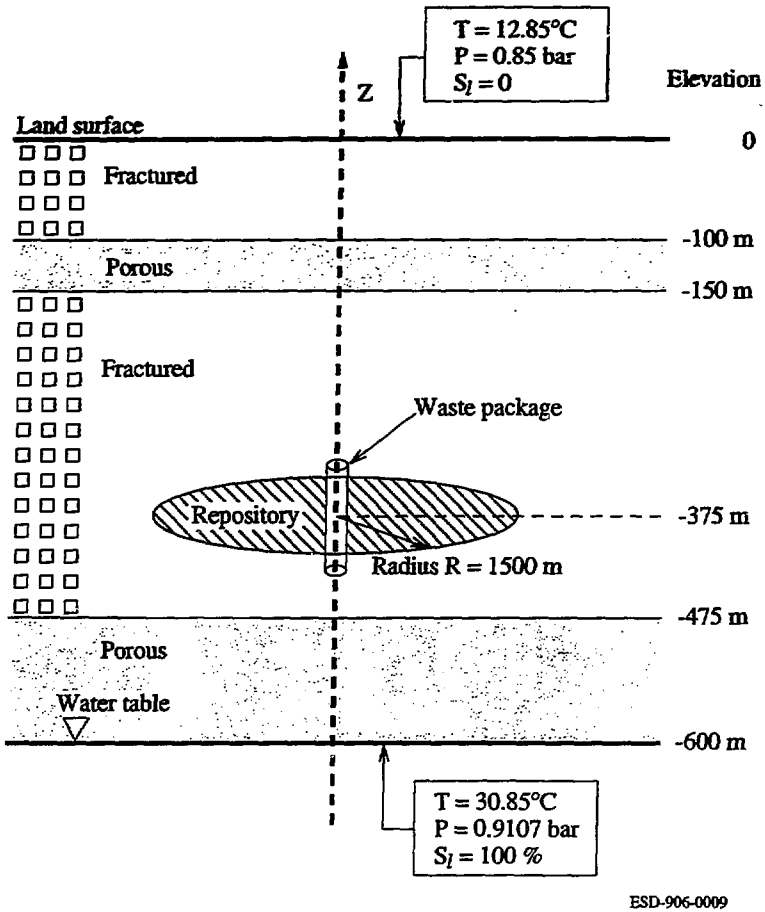

Fig. 1: Two-dimensional radially symmetric (R-Z) model of a high-level nuclear waste repository at Yucca Mountain.

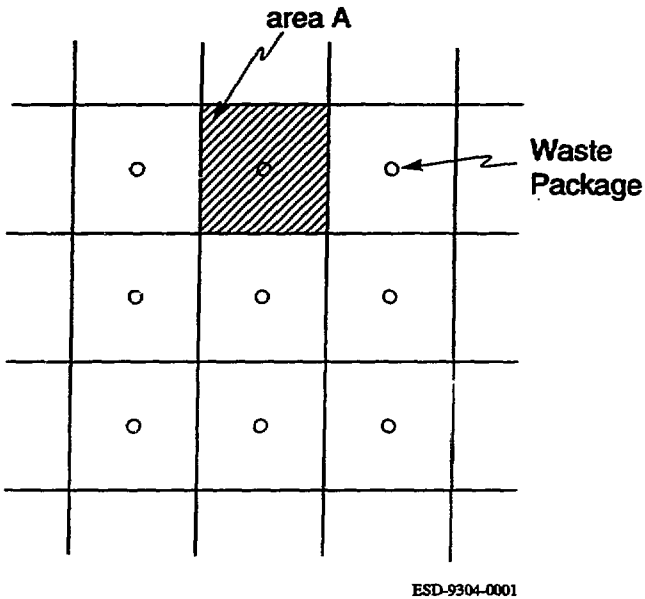

Fig. 2: Areal view of an idealized repository with vertical waste package emplacement. Each waste package occupies an area A.

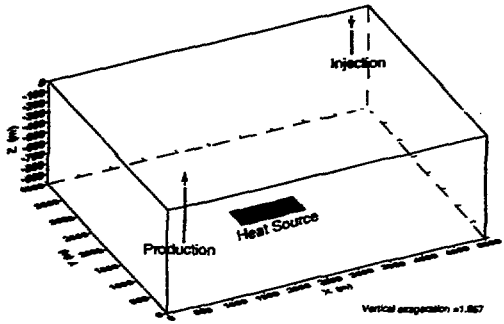

Fig. 3: Characteristics of the Cartesian grid models

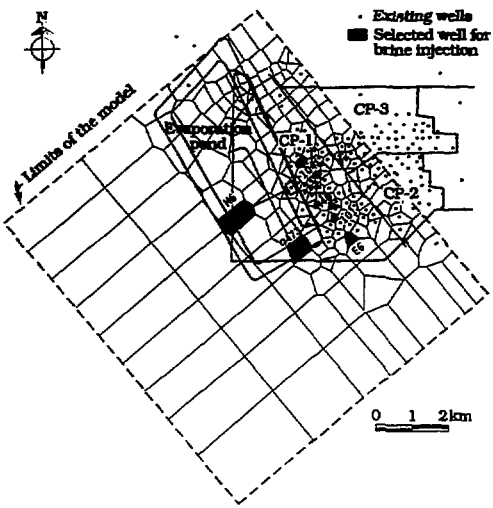

Fig. 4: Cerro Prieto model. Characteristics of the irregular computational grid.

Table 1: Timing of the test runs for TOUGH2/PC with the solvers package

Newtonian iteration tolerance = $1x10^{-5}$
Closure in CG solvers = $1x10^{-6}$

Case	Grid size	Solver	Number of iterations			Time (sec)					Observations
			Total[1]	Newtonian	Repeated due to convergence failure	per Newtonian iteration	Input	CPU	Total execution	Simulated	
1	500	MA28	96	71	0	10.84	4.07	791.31	795.38	7.1677E9	Standard vers.
		GMRES	108	82	1	10.35	3.63	870.73	874.26	4.5053E9	
		BCG	104	78	1	7.56	3.46	611.21	614.67	4.7101E9	
		BCGS	97	71	1	6.79	3.46	518.06	521.52	4.7101E9	
	1,000	MA28	96	71	0	77.57	9.12	5551.59	5560.71	19.660E9	Standard vers.
		GMRES	126	97	4	46.78	8.08	4593.14	4601.22	4.0957E9	
		BCG	103	75	3	27.45	7.74	2246.84	2254.58	6.9629E9	
		BCGS	98	70	3	20.60	7.97	1726.42	1734.39	4.3005E9	
	2,000	MA28	100	75	0	226.01	21.64	17041.46	17063.10	4.3005E9	Standard vers.
		GMRES	144	113	6	227.88	19.55	25878.42	25897.97	1.2285E9	
		BCG	112	82	5	127.92	19.06	11947.56	11966.62	1.3821E9	
		BCGS	99	70	4	77.68	18.95	6655.65	6674.60	1.9453E9	
	5,000	GMRES	101	68	8	1381.20	80.57	94437.19	94517.76	1.7250E8	
		BCG	106	75	6	471.38	79.20	48964.26	49043.46	5.5501E8	
		BCGS	102	74	3	244.97	80.91	31499.60	31580.51	7.1650E8	
	10,000	GMRES	57	39	3	4803.00	261.83	187876.22	188138.1	4.1110E7	15 time steps
		BCG	109	77	7	1287.75	268.80	163326.51	163595.3	2.1730E8	
		BCGS	97	64	8	472.05	269.85	106601.76	106871.6	3.3250E8	
2	500	BCGS	140	113	2	4.02	3.79	485.38	489.17	9.0900E7	
	1,000	"	122	97	0	10.00	7.85	1017.44	1025.29	9.0900E7	
	2,000	"	134	108	1	25.75	18.73	2910.39	2929.12	9.0900E7	
	5,000	"	132	107	0	69.95	82.77	7732.52	7815.29	3.8100E7	
	10,000	"	134	109	0	162.41	274.29	18203.14	18477.43	2.5300E7	
3	1,411	BCGS	145	117	3	21.20	33.51	2595.45	2628.96	4.8061E8	
4	5644	BCGS	194	154	15	55.56	114.85	10347.09	10461.94	4.1001E8	3 MINC shells
	8466	"	179	142	12	85.53	169.94	16808.09	16978.13	4.1003E8	5 MINC shells

[1] The total number of iterations includes one additional convergence iteration per prescribed time step, 25 in total. At each iteration convergence is checked and if convergence is satisfied a new time step is started.