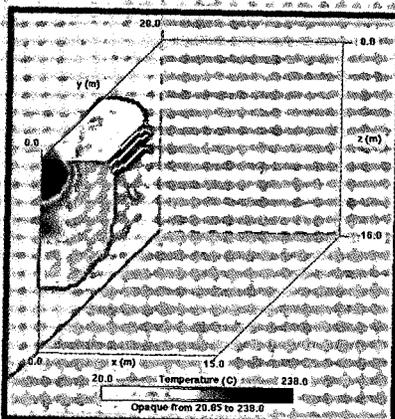
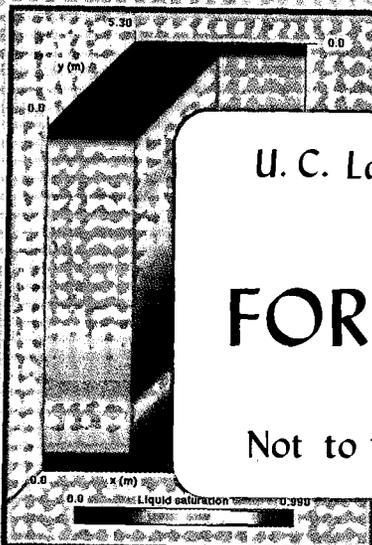
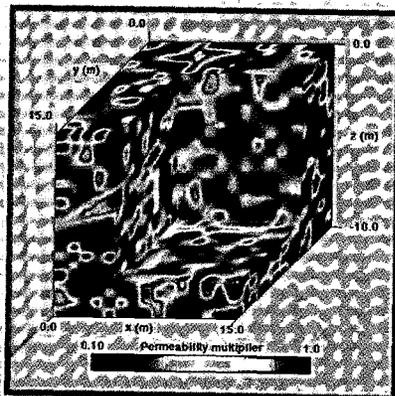


Flow and Transport Simulations Using T2CG1, A Package of Conjugate Gradient Solvers For the TOUGH2 Family of Codes



U. C. Lawrence Berkeley Laboratory
Library, Berkeley

FOR REFERENCE

Not to be taken from this room

G. Moridis and K. Pruess

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

April 1995

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.

REFERENCE COPY
Does Not Circulate
Bldg. 50 Library.
Copy 1

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, or The Regents of the University of California.

Available to DOE and DOE Contractors
from the Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, VA 22161

Lawrence Berkeley Laboratory is an equal
opportunity employer.

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-36235
UC-400

**FLOW AND TRANSPORT SIMULATIONS USING T2CG1,
A PACKAGE OF CONJUGATE GRADIENT SOLVERS
FOR THE TOUGH2 FAMILY OF CODES**

G. Moridis and K. Pruess

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

April 1995

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.

ABSTRACT

T2CG1, a package of preconditioned conjugate gradient solvers, has been added to the TOUGH2 family of codes to complement its direct solver (MA28) and significantly increase the size of tractable problems^(*). T2CG1 is *'plug-compatible'* with the 1991 Version 1.0 of TOUGH2, and retains the option of using the direct solver MA28. T2CG1 includes three new solvers: a Bi-Conjugate Gradient (BCG) solver, a Conjugate Gradient Squared (CGS) solver, and a Generalized Minimum Residual (GMRES) solver. T2CG1 is tested in the solution of 16 non-linear test problems ranging from one-dimensional single-phase isothermal flow to complex three-dimensional three-phase multicomponent non-isothermal flow and transport with phase changes. Results from a variety of computing platforms and from systems with up to 30,000 equations show that the CG routines in T2CG1 (1) demonstrate a problem-, compiler-, and machine-specific performance, as is the case for all iterative solvers, (2) are significantly (and invariably) faster than MA28, and (3) require far less memory than the MA28 direct solver, and thus (4) enable the solution of much larger problems (previously tractable on supercomputers only) on microcomputers. The DSLUCS solver has the best overall performance.

The present series of tests includes multiphase, multicomponent fluid and heat flow problems related to geothermal reservoirs, vadose zone hydrology, nuclear waste disposal, and environmental remediation. Although one or two solvers performed poorly in certain problems, each of the 16 test problems could be efficiently and accurately solved by at least one of the T2CG1 routines.

* The T2CG1 module is available from: Energy Science and Technology Software Center (ESTSC), P.O. Box 1020, Oak Ridge, TN 37831; telephone: (615) 576 2606; e-mail: estsc@adonis.osti.gov.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF FIGURES	vii
LIST OF TABLES	x
1. INTRODUCTION	1
1.1. Direct and Iterative Solvers	1
1.2. The TOUGH2 Family of Codes and its Solvers	2
1.3. Purpose	4
1.4. Organization and Approach	4
2. THE CONJUGATE GRADIENT METHODS IN T2CG1	6
2.1. Origin and Development	6
2.2. The Mathematical Concepts of Conjugate Gradients	7
2.3. The Bi-Conjugate Gradient (BCG) Method	8
2.4. The Bi-Conjugate Gradient Squared (BCGS) Method	9
2.5. The Generalized Minimum Residual (GMRES) Method	10
2.6. Preconditioning	11
3. DATA REQUIREMENTS AND USE OF T2CG1	13
3.1. Solver Selection	13
3.2. Maximum Number of Iterations	13
3.3. Convergence Criterion	14
3.4. Solver-Related Output	14
4. TEST PROBLEMS	16
4.1. Test Problem 1: The Theis Problem	16
4.1.1. Problem description	16
4.1.2. Solver performance	16
4.2. Test Problem 2: Radial Heat Transport	20
4.2.1. Problem description	20
4.2.2. Solver performance	20
4.3. Test Problem 3: Convection Cell Experiment	25
4.3.1. Problem description	25
4.3.2. Solver performance	27
4.4. Test Problem 4: Determination of Effective Permeability in a Homogeneous Medium with Embedded Impermeable Objects	35
4.4.1. Problem description	35
4.4.2. Solver performance	35

4.5.	Test Problem 5: Determination of Effective Permeability in a Medium with Strong Spatially Correlated Heterogeneity	41
4.5.1.	Problem description	41
4.5.2.	Solver performance.....	41
4.6.	Test Problem 6: 2-D Infiltration Experiment.....	46
4.6.1.	Problem description	46
4.6.2.	Solver performance.....	47
4.7.	Test Problem 7: Vertical Section of WIPP.....	52
4.7.1.	Problem description	52
4.7.2.	Solver performance.....	52
4.8.	Test Problem 8: Saturated-Unsaturated Flow in a 3-D Medium.....	57
4.8.1.	Problem description	57
4.8.2.	Solver performance.....	58
4.9.	Test Problem 9: The Sandia Thermal Enhanced Vapor Extraction System - TEVES.....	63
4.9.1.	Problem description	63
4.9.2.	Solver performance.....	63
4.10.	Test Problem 10: 3-D Geothermal Reservoir Model.....	68
4.10.1.	Problem description	68
4.10.2.	Solver performance.....	68
4.11.	Test Problem 11: Cerro Prieto Geothermal Field	77
4.11.1.	Problem description	77
4.11.2.	Solver performance.....	77
4.12.	Test Problem 12: Repository Performance Assessment at Yucca Mountain.....	80
4.12.1.	Problem description	80
4.12.2.	Solver performance.....	80
4.13.	Test Problem 13: Channelized Water Flow at Yucca Mountain	83
4.13.1.	Problem description	83
4.13.2.	Solver performance.....	83
4.14.	Test Problem 14: Solutal Convection in Porous Media	88
4.14.1.	Problem description	88
4.14.2.	Solver performance.....	88
4.15.	Test Problem 15: The Buckley-Leverett Problem.....	92
4.15.1.	Problem description	92
4.15.2.	Solver performance.....	92

4.16. Test Problem 16: Air Barrier System for Waste Containment in the Vadose Zone.....	95
4.16.1 Problem description.....	95
4.16.2 Solver performance.....	96
5. CONCLUSIONS AND DISCUSSION.....	99
5.1. Summary and Conclusions.....	99
5.2. Choice of Solver.....	100
5.3. Important Issues and Precautions.....	104
6. ACKNOWLEDGMENTS.....	106
7. REFERENCES.....	105
APPENDIX A - SOURCE CODE CHANGES	111
APPENDIX B - MAILING LIST FOR YUCCA MOUNTAIN SITE CHARACTERIZATION PROJECT	113

LIST OF FIGURES

Figure 1.	The Theis problem: radial flow to a single well.	18
Figure 2.	Comparison of the analytical and the TOUGH2 solutions to the Theis problem at $t = t_{max} = 10 \text{ days}$	19
Figure 3.	The radial heat transport system in Test Problem 2.....	22
Figure 4.	Comparison of the analytical and TOUGH2 solutions to the radial heat transport problem (Test Problem 2).....	23
Figure 5.	Analytical and TOUGH2 similarity solutions in Test Problem 2.....	24
Figure 6.	Schematic of the convection cell apparatus in Test Problem 3 (from <i>Reda</i> [1984]).	29
Figure 7.	The convection cell experiment in Test Problem 3.....	30
Figure 8.	Comparison of TOUGH2 predictions with experimental data at the top and bottom ($z/\Delta r = 3$ and 2 respectively) of the heating element in Test Problem 3. The bars attached to the TOUGH2 curves indicate the predicted temperature on the surface of the heater.	31
Figure 9.	Comparison of TOUGH2 predictions with experimental data at $z/\Delta r$ $= 3.5$ and $z/\Delta r = 3.75$	32
Figure 10.	CG solver performance in the first Newtonian iteration of the first timestep in Test Problem 3 when flow channeling is neglected.	33
Figure 11.	CG solver performance in the first Newtonian iteration of the first timestep in Test Problem 3 when permeability enhancement is considered.	34
Figure 12.	The computational grid in Test Problem 4.	38
Figure 13.	The distribution of embedded impermeable objects in Test Problem 4.....	39
Figure 14.	Performance of the CG routines in the first Newtonian iteration of the 10th timestep in Test Problem 4.	40
Figure 15.	Distribution of the permeability modifier in Test Problem 4.....	43
Figure 16.	Performance of the DSLUBC and DSLUGM routines in the first Newtonian iteration of the first timestep in Test Problem 5.....	44
Figure 17.	Divergence of the DSLUCS solution in the first Newtonian iteration of the first timestep in Test Problem 5.....	45
Figure 18.	The two-dimensional infiltration system of <i>Vauclin's</i> [1979] experiment in Test Problem 6.	48
Figure 19.	The TOUGH2 grid for the simulation of Test Problem 6.....	49

Figure 20.	TOUGH2 predictions and experimental measurements at $x = 0.19\ m$ in Test Problem 6.	50
Figure 21.	TOUGH2 predictions and experimental measurements at $x = 1.39\ m$ in Test Problem 6.	51
Figure 22.	Schematic of typical WIPP underground rooms showing stratigraphic positions in Test Problem 7.	55
Figure 23.	Gas saturation distribution in the subsurface in the vicinity of a WIPP underground room in Test Problem 7.	56
Figure 24.	Discretization in test Problems 8 and 8a.	59
Figure 25.	Water saturation distribution in the homogeneous three-dimensional domain of Test Problem 8a.	60
Figure 26.	Water saturation distribution in the highly heterogeneous three-dimensional domain of Test Problem 8a.	61
Figure 27.	Relative performance of the CG solvers in the first Newtonian iteration of the 15th timestep in Test Problem 8a.	62
Figure 28.	Schematic of the Thermal Enhanced Vapor Extraction System (TEVES) in Test Problem 9.	65
Figure 29.	Temperature distribution at $t = 1.897 \times 10^6\ sec$ in the TEVES system simulated in Test Problem 9.	66
Figure 30.	Relative performance of the CG solvers in the first Newtonian iteration of the 50th timestep in Test Problem 9.	67
Figure 31.	The simulation domain in Test Problem 10.	72
Figure 34.	Performance of the CG solvers in the first Newtonian iteration of the 11th timestep in Test Problem 10 (5000 gridblocks, $N = 10,000$ equations, single-phase conditions).	75
Figure 32.	Test Problem 10: temperature distribution at $t_{max} = 1.33 \times 10^9\ sec$ (single-phase geothermal reservoir, 5000 cells).	73
Figure 33.	Test Problem 10: liquid saturation distribution at $t_{max} = 6.57 \times 10^8\ sec$ (two-phase geothermal reservoir, 5000 cells).	74
Figure 34.	Performance of the CG solvers in the first Newtonian iteration of the 11th timestep in Test Problem 10 (5000 gridblocks, $N = 10,000$ equations, two-phase conditions).	75
Figure 35.	Performance of the CG solvers in the first Newtonian iteration of the 11th timestep in Test Problem 10 (5000 gridblocks, $N = 10,000$ equations, two-phase conditions).	76

Figure 36.	Characteristics of the irregular computational grid of the Cerro Prieto model in Test Problem 11.....	79
Figure 37.	Two-dimensional radially symmetric (r,z) model of a high-level nuclear waste repository at Yucca Mountain.....	81
Figure 38:	Liquid saturation distribution at $r = 0.1515 m$ and $t = 5.6 \times 10^5 sec$ in Test Problem 12.....	82
Figure 39.	Areal view of an idealized repository with vertical waste package emplacement. Each waste package occupies an area A	85
Figure 40.	Distribution of liquid saturation in the three-dimensional simulation domain of test Problem 13 at $t = 365.25$ years.....	86
Figure 41.	Performance of the CG solvers in the first Newtonian iteration of the 10th timestep in Test Problem 13 (1008 gridblocks, $N = 3,024$ equations, two-phase conditions).....	87
Figure 42.	Discretization (uniform $N_Y \times N_Z = 60 \times 32$ grid blocks) for the Elder problem.	90
Figure 43.	Results for the Elder free-convection problem at $t = 20$ years.....	91
Figure 44.	Analytical solution and T2VOC results for the Buckley-Leverett problem in Test Problem 15.....	94
Figure 45.	Evolution of the dry air barrier in Test Problem 16.....	97
Figure 46.	Performance of the CG solvers in the first Newtonian iteration of the 5th timestep in Test Problem 16 (1720 gridblocks, $N = 6,880$ equations).....	98
Figure 47.	Relationship between the CPU time per Newtonian iteration and the order of the matrix, N , in the DSLUBC solver ($C_{BC} = 2.09 \times 10^{-5}$).	101
Figure 48.	Relationship between the CPU time per Newtonian iteration and the order of the matrix, N , in the DSLUCS solver ($C_{CS} = 4.79 \times 10^{-5}$).....	102
Figure 49.	Relationship between the CPU time per Newtonian iteration and the order of the matrix, N , in the DSLUGM solver ($C_{GM} = 1.19 \times 10^{-5}$).	103

LIST OF TABLES

Table 1.	Solver Performance in Test Problem 1	17
Table 2.	Solver Performance in Test Problem 2	21
Table 3.	Solver Performance in Test Problem 3	28
Table 4.	Parameters for Test Problem 4	36
Table 5.	Solver Performance in Test Problem 4	37
Table 6.	Solver Performance in Test Problem 5	42
Table 7.	Solver Performance in Test Problem 6	47
Table 8.	Solver Performance in Test Problem 7	54
Table 9.	Solver Performance in Test Problem 8	58
Table 10.	Solver Performance in Test Problem 9	64
Table 11.	Domain Discretizations in Test Problem 10.....	70
Table 12.	Solver Performance in Test Problem 10 (5000 cells).....	70
Table 13.	Solver Performance on a PC in Test Problem 10	71
Table 14.	Solver Performance on a PC in Test Problem 11	78
Table 15.	Solver Performance in Test Problem 12	80
Table 16.	Solver Performance in Test Problem 13	84
Table 17.	Parameters for the Elder [1967] Problem (Test Problem 14).....	89
Table 18.	Solver Performance in Test Problem 14	89
Table 19.	Solver Performance in Test Problem 15	93
Table 20.	Solver Performance in Test Problem 16	96

1. INTRODUCTION

1.1. Direct and Iterative Solvers

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. These equations are solved using either direct or iterative methods. The most reliable (and often the simplest) solvers are based on direct methods. The robustness of direct solvers comes at the expense of large storage requirements and execution times. Iterative techniques exhibit problem-specific performance and lack the generality, predictability and reliability of direct solvers. These disadvantages are outweighed by their low computer memory requirements and their substantial speed especially in the solution of very large matrices. With the advent of rapidly improving computing power, the array of tractable problems expands to include increasingly complex, large and three-dimensional systems. Iterative solvers are currently used almost exclusively for the solution of these large problems, while direct methods are usually limited to systems of a few hundred to at most a few thousand equations.

The matrices arising from Finite Difference and Finite Element simulations of fluid and heat flow and transport through porous and fractured media are sparse, i.e., typically only a few percent of the matrix elements are non-zero. If *standard* ordering [Aziz and Settari, 1979] of the gridblocks of a regularly discretized domain is employed, the resulting matrices are also banded. Defining *work* W as the number of multiplications and divisions necessary to reduce the matrix to upper triangular form and to perform back substitution, it was shown [Price and Coats, 1974] that for direct solvers (Gauss elimination and LU decomposition)

$$W \propto NM^2 \quad (1)$$

and the minimum storage requirement S is

$$S = NM \quad (2)$$

where N is the order of the matrix and M its half-bandwidth, the full bandwidth being

$$B = 2M + 1. \quad (3)$$

The form of the matrix depends upon the ordering of equations. It is evident from equations (1) and (2) that for a given problem size N , work and storage are minimized when M is minimized. If I , J , K are the number of subdivisions in the x -, y - and z -directions respectively, the shortest half-bandwidth $M = JK$ when $I > J > K$. As W increases with the square of M , it is obvious that the penalty for non-optimization of the ordering of equations may be substantial.

More efficient numbering schemes such as D2 and D4 [Price and Coats, 1974] can reduce W and S by factors of up to 5.85 and 2 respectively. Even if such numbering is possible, it is obvious that the matrix bandwidth and the corresponding work during direct solution increase dramatically with the dimensionality of the problem. This clearly demonstrates the reason for the limitations in the application of direct matrix methods to no more than a few thousand grid blocks in one- and two-dimensional problems, or a few hundred gridblocks in three-dimensional systems.

Iterative matrix methods are currently the most attractive (and often the only possible) practical means for the solution of large systems of linear equations. The bibliography on the subject is extensive. The previous generation of the most common iterative solvers included the *Successive Over-Relaxation* method [Varga, 1962; Young, 1971], the *Alternating Direction Implicit* method [Peaceman and Rachford, 1955], the *Strongly Implicit Procedure* [Stone, 1968], as well as their variants. While these methods had the advantage of speed and limited memory requirement over direct methods, they suffered from a number of shortcomings. They were problem-specific, required diagonal dominance for satisfactory convergence, were sensitive to directional and general heterogeneity, and relied on a number of parameters which were difficult to determine and critical to the performance of the methods. Despite their unpredictability, the approach to the numerical simulation of fluids in the subsurface (limited heterogeneity, IMPES or semi-implicit formulations, flow coefficient matrices and not Jacobians) at the time of their development was such that they were useful and satisfactory for a wide variety of applications.

The underlying concepts of the family of solvers termed *Conjugate Gradient (CG) Methods* were introduced in the early 1950s [Hestenes and Stiefel, 1952], and attracted considerable attention when first presented. Curiously, application of CG methods was very limited until the mid-1970s, when they enjoyed a strong resurgence. Currently CG methods are the most commonly used techniques for the iterative solution of large systems of simultaneous linear equations. CG methods, though iterative, converge to the true solution in a finite number of iterations in the absence of roundoff errors. Conventional iterative methods exhibit practically acceptable convergence only for diagonally dominant matrices, but conjugate gradient methods have no such limitation [Hestenes and Stiefel, 1952]. CG methods do not rely on parameter estimates (such as the overrelaxation factor in SOR) for a satisfactory performance. Because of these and other properties, CG methods have become the most attractive alternative for the solution of linear systems in recent years. In the present report we discuss the use and performance of a CG package in the TOUGH2 family of codes.

1.2. The TOUGH2 Family of Codes and its Solvers

The TOUGH2 family of codes is a descendant of MULKOM which first became operational in the early 1980s [Pruess, 1983]. MULKOM was built on the recognition that the governing equations for non-isothermal, multi-phase, multi-component flow are the same, regardless of the nature and number of fluid components present. Accordingly, MULKOM featured a modular architecture, in which a central module for flow and transport could be interfaced with several different fluid property modules to model systems with different fluid mixtures. The development of MULKOM was driven by research applications primarily in geothermal reservoir engineering, and later also in nuclear waste-related studies. The coding of MULKOM was never finalized; rather, it consisted of a continually evolving collection of program modules for specialized applications, with only sketchy documentation. The needs of nuclear-waste oriented applications then prompted an effort to finalize and document a specialized version of MULKOM for non-isothermal two-phase flows of water and air, which in 1987 was released under the name TOUGH (for Transport Of Unsaturated Ground-water and Heat; [Pruess, 1987]). A number of enhanced versions of TOUGH have subsequently been developed at the Lawrence Berkeley Laboratory (LBL), as well as in other places [Pruess, 1990]. The methodological choices made in the development of MULKOM have been discussed in a laboratory report [Pruess, 1988].

With its "hardwiring" of the two fluid components water and air, TOUGH traded-in some of MULKOM's flexibility against greater robustness and ease of use. The successor to TOUGH is the TOUGH2 code [Pruess, 1991; ESTSC, 1992], an upwardly-compatible code after which the current family of simulators is named. The name "TOUGH2" is now used to refer to both the particular modular architecture of these codes and the actual coding implementations referred to as TOUGH2. The TOUGH2 family of codes includes (a) TOUGH2 for two-phase multi-component fluid and heat flow with phase change, (b) T2DM, a TOUGH2 module for Fickian hydrodynamic dispersion [Oldenburg and Pruess, 1993], and (c) T2VOC [Falta et al., 1995] and M2NOTS [Adenekan, 1992; Adenekan et al., 1993] for three-phase multi-component fluid and heat flow.

TOUGH2 is actually subroutine-for-subroutine very similar to TOUGH, but offers added new features (internal mesh generation, embedded fractured-media capability with the "MINC" method of "Multiple *I*nteracting Continua", version control), and it provides the full multi-component, multi-phase flexibility previously only available in the undocumented MULKOM collection of research modules. It allows the use of several equation-of-state modules (called EOS n , $n=1,2,\dots$) describing the behavior of different fluid systems. All the members of the TOUGH2 family use a fully implicit approach for the solution of the coupled and extremely non-linear Partial Differential Equations (PDEs) describing the processes of fluid and heat flow and transport through porous and/or fractured media. The non-linear equations are linearized using a Newton-Raphson iteration, and the resulting Jacobian is solved to yield the changes in the primary variables in each element (of the discretized domain) from the previous iteration. The implicitness of the approach allows the solution of some of the most complex and challenging numerical problems of flow. In the original TOUGH2, the Jacobian is solved using MA28 [Duff, 1977], a direct solver using LU decomposition and sparse matrix storage techniques.

T2DM is a module for Fickian hydrodynamic dispersion that has been added to TOUGH2 [Oldenburg and Pruess, 1993]. T2DM has been used to solve flow problems which include seawater intrusion, solutal convection, mixed forced and free convection, as well as double-diffusive convection where heat and salinity affect fluid density. The dispersion module, intended for use with the EOS7 fluid properties module for water, brine, and air, accounts for the effects of hydrodynamic dispersion and molecular diffusion of solute in two-dimensional rectangular domains. The dispersive fluxes depend not only on the primary variables of the two connected grid blocks but on all primary variables of the neighbor grid blocks of each interface. This gives rise to added terms and increased bandwidth in the Jacobian matrix relative to standard TOUGH2 where fluxes depend only on primary variables in the two connected grid blocks. The additional non-zero terms in the Jacobian result in long execution times with the direct solver MA28.

Falta et al. [1991; 1992a; b] developed an enhanced version of TOUGH2 for three-phase three-component flow of water, air, and volatile organic compounds, known as "STMVOC". The process complexities of highly non-linear fully transient three-phase multi-component fluid and heat flow with phase change were rigorously accounted for, making this the most difficult solvable three-phase fluid and heat flow problem presently available. STMVOC was later modified to better conform to the TOUGH2 architecture and nomenclature, and code capabilities were significantly enhanced. This resulted in the T2VOC code [Falta et al., 1995].

Throughout the development of the TOUGH2 codes an effort was made to maintain continuity in numerical performance, i.e., different code variations were always exercised on several earlier benchmark problems to guard against occurrence of coding errors. Therefore, verification or validation exercises performed with any one member of the TOUGH2 family are significant in establishing credibility for all the other members as well.

Accordingly, our summary of verification and validation work will cover all of the TOUGH2 variations developed at LBL.

MA28 provided for a long time a robust solver capable of handling some of the toughest problems of non-linear flow and transport. However, for reasons already discussed, storage and execution time requirements of MA28 limited the use of TOUGH2 to a maximum of a few thousand equations. We have added T2CG1, a package of preconditioned conjugate gradient solvers to TOUGH2 to complement its direct solver and significantly increase the size of tractable problems. The conjugate gradient solvers decrease execution time and memory requirements substantially, and make possible the simulation of three-dimensional flow problems with tens of thousands of 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.

1.3. Purpose

The purpose of this study is to discuss the application and performance of the T2CG1 package of CG solvers in the TOUGH2 family of codes. We intend to

- (a) Provide a mathematical background of the CG methods in the T2CG1 package, as well as information on the origins and development of the T2CG1 routines.
- (b) Supply a User's Guide for T2CG1, and relevant information on the input data requirements for the use of the package.
- (c) Investigate the performance of the CG solvers in a variety of problems of fluid and heat flow and transport through porous and fractured media, and demonstrate the power of the T2CG1 package in terms of execution speed and size of problems and applications.
- (d) Discuss a set of sound principles and practices, and suggest a few "rule-of-thumb" guidelines, to be used in the application of T2CG1.
- (e) Demonstrate the capabilities of the TOUGH2 family of codes in a variety of large-scale complex applications, and how these are enhanced by the availability of the T2CG1 package.

1.4. Organization and Approach

This report presupposes knowledge of the structure and nomenclature used in the TOUGH2 family of codes. These can be found in *Pruess* [1987], *Pruess* [1991], *Falta et al.* [1992, 1995], and *Oldenburg and Pruess* [1993].

Section 1 provides the background and scope of this effort. The mathematical concepts of the CG methods in the T2CG1 package appear in Section 2, as well as a discussion on the origins and code development. Section 3 describes the data inputs and provides a User's Guide for the use of the T2CG1 package.

In Section 4 we discuss the application of the T2CG1 module to a series of 16 test problems, and critically evaluate the performance of the CG methods employed

compared to the direct solver MA28 (if possible) and to each other. Finally, Section 5 presents a summary and conclusions.

The test problems in Section 4 include verification problems (for which either analytical or numerical solutions exist), validation problems (for which experimental results are available), and synthetic numerical problems (for which neither analytical solutions nor experimental results are available). Verification is shown by the proximity of the TOUGH2-produced solutions to the known analytical and/or numerical solution. Validation is indicated by the ability of the TOUGH2 solutions to reproduce experimentally observed data. In the review and discussion of each of the 16 test problems the following approach is used:

- (a) A short description of the problem is presented.
- (b) The inputs are discussed.
- (c) Runs are made using all three CG methods in the T2CG1 package, as well as the MA28 direct solver (memory permitting).
- (d) Convergence, timing and memory requirements are discussed and compared.

The calculations reported here were performed on a variety of computing platforms: two IBM RS/6000 workstations (models 370 and 320H) using double-precision (64-bit) arithmetic, a Macintosh Quadra 800 microcomputer equipped with a 68040 processor, and an IBM-type PC powered by a 486 DX2-66 processor. Because of the variety of computational platforms, compilers and optimization options used, the reader is cautioned against unwarranted generalizations about their relative performance. What is essential is the relative performance of solvers, and this is provided using the same platform for each test problem.

A computer diskette (PC type of format) with several of the input files used in our simulations is included as an attachment to this report. We include the compressed input files from problems using the officially released standard versions of TOUGH2 and T2VOC codes. We do not include the input files of non-standard TOUGH2 versions (i.e. not generally available at the time of the release of this report). Problems requiring the T2DM module are not included since these codes had not been officially released by the date of the completion of this report. We clearly indicate in the narrative of each test problem whether the corresponding input data file is included in the diskette.

To make the input files more easily readable we have added column counters to the keywords. An explanation of the files and their content is found in the text file "ReadMe" also included in the same disk. These input files can serve as references and as building blocks for other TOUGH2 and T2VOC applications. The diskette also includes listings of the subroutines PCAP (the capillary pressure subroutine) and RELP (the relative permeability subroutine) of TOUGH2. The subroutines on the diskette are slightly expanded versions of the standard code, and include the additional capillary pressure and relative permeability functions needed to accommodate the needs of some of the test problems. The changes in the code amount to no more than 15 lines, and the effort to implement them is minimal. The added functions are clearly marked for easy identification.

2. THE CONJUGATE GRADIENT METHODS IN T2CG1

2.1. Origin and Development

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

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (4)$$

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, \mathbf{A} matrices arising in TOUGH2 simulations are non-symmetric \mathbf{M} -matrices with typically no diagonal dominance. Although \mathbf{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 \mathbf{A} is a Jacobian, the elements of \mathbf{A} in a single row often vary by several orders of magnitude. In TOUGH2 simulations there are often conditions which result in a large number of zeroes on the main diagonal of \mathbf{A} , making pivoting impossible and resulting in very ill-conditioned matrices. Such zeroes on the main diagonal commonly arise in multiphase flow problems in connection with different sets of primary variables used for describing different phase conditions. 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 \mathbf{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.

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 \mathbf{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 code uses a matrix storage scheme which is identical to the SLAP Triad Matrix Storage Format, and was maintained unaltered in T2CG1.

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 DSLUBC. 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 may occasionally fail to converge successfully, we feel that problems where all three solvers fail are extremely rare.

2.2. The Mathematical Concepts of Conjugate Gradients

The basic "ordinary" conjugate gradient (CG) algorithm solves equation (4) only when the matrix \mathbf{A} is Symmetric Positive Definite (SPD) by minimizing the function

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x} \bullet \mathbf{A} \bullet \mathbf{x} - \mathbf{b} \bullet \mathbf{x}, \quad (5)$$

which occurs when the gradient

$$\nabla f = \mathbf{A} \bullet \mathbf{x} - \mathbf{b} = 0. \quad (6)$$

This is equivalent to equation (4), and the minimization is carried out by generating a succession of search directions \mathbf{p}_n and improved minimizers \mathbf{x}_n [Press *et al.*, 1989]. For a $N \times N$ SPD matrix \mathbf{A} , if the initial guess of the solution is \mathbf{x}_0 , the "ordinary" CG algorithm is

```

r0 := b - Ax0;
p-1 := 0;
ρ-1 := 1;
n := 0;
while residual > tolerance do
begin
  ρn := rnTrn;
  βn :=  $\frac{\rho_n}{\rho_{n-1}}$ ;
  pn := rn + βnpn-1;
  σn := pnTApn;
  αn :=  $\frac{\rho_n}{\sigma_n}$ ;
  rn+1 := rn - αnApn;
  xn+1 := xn - αnpn;
  n := n + 1;
end

```

The vectors \mathbf{r}_n satisfy $\mathbf{r}_n = \mathbf{b} - \mathbf{A}\mathbf{x}_n$ and are called residuals. It has been proved that this algorithm, in the absence of rounding errors and barring divisions by zero, gives the exact answer in a maximum of N steps [Golub and van Loan, 1989].

2.3. The Bi-Conjugate Gradient (BCG) Method

When the \mathbf{A} matrices are not SPD, as the ones commonly arising from the use of upstream weighting to control numerical diffusion in the solution of multi-phase flow PDEs, the "ordinary" CG algorithm is no longer applicable. Fletcher [1976] proposed the BCG method for the solution of linear, but not necessarily positive definite or symmetric systems. The BCG algorithm is given by

```

 $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0;$ 
 $\mathbf{p}_{-1} := \tilde{\mathbf{p}}_{-1} := \mathbf{0};$ 
 $\rho_{-1} := 1;$ 
 $n := 0;$ 
while residual > tolerance do
begin
 $\rho_n := \mathbf{r}_n^T \mathbf{r}_n;$ 
 $\beta_n := \frac{\rho_n}{\rho_{n-1}};$ 
 $\mathbf{p}_n := \mathbf{r}_n + \beta_n \mathbf{p}_{n-1};$ 
 $\tilde{\mathbf{p}}_n := \tilde{\mathbf{r}}_n + \beta_n \tilde{\mathbf{p}}_{n-1};$ 
 $\sigma_n := \tilde{\mathbf{p}}_n^T \mathbf{A} \mathbf{p}_n;$ 
 $\alpha_n := \frac{\rho_n}{\sigma_n};$ 
 $\mathbf{r}_{n+1} := \mathbf{r}_n - \alpha_n \mathbf{A} \mathbf{p}_n;$ 
 $\tilde{\mathbf{r}}_{n+1} := \tilde{\mathbf{r}}_n - \alpha_n \mathbf{A}^T \tilde{\mathbf{p}}_n;$ 
 $\mathbf{x}_{n+1} := \mathbf{x}_n + \alpha_n \mathbf{p}_n;$ 
 $n := n + 1;$ 
end

```

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) if the \mathbf{A} matrix is non-symmetric, in practice this is rare [Press *et al.*, 1986]. If a good preconditioner is used, BCG is an effective method [Seager, 1988]. If the \mathbf{A} matrix is not too far from being SPD, BCG is expected to converge efficiently [Sonneveld, 1989].

2.4. The Bi-Conjugate Gradient Squared (BCGS) Method

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. If \mathbf{x}_0 is a suitably chosen initial estimate of the non-singular equation (4), the algorithm is

```

 $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0;$ 
 $\mathbf{q}_0 := \mathbf{p}_{-1} := \mathbf{0};$ 
 $\rho_{-1} := 1;$ 
 $n := 0;$ 
while residual > tolerance do
begin
 $\rho_n := \mathbf{r}_n^T \mathbf{r}_n;$ 
 $\beta_n := \frac{\rho_n}{\rho_{n-1}};$ 
 $\mathbf{u}_n := \mathbf{r}_n + \beta_n \mathbf{q}_n;$ 
 $\mathbf{p}_n := \mathbf{u}_n + \beta_n (\mathbf{q}_n + \beta_n \mathbf{p}_{n-1});$ 
 $\mathbf{v}_n := \mathbf{A}\mathbf{p}_n;$ 
 $\sigma_n := \mathbf{r}_n^T \mathbf{v}_n;$ 
 $\alpha_n := \frac{\rho_n}{\sigma_n};$ 
 $\mathbf{q}_{n+1} := \mathbf{u}_n - \alpha_n \mathbf{v}_n;$ 
 $\mathbf{r}_{n+1} := \mathbf{r}_n - \alpha_n \mathbf{A}(\mathbf{u}_n + \mathbf{q}_{n+1});$ 
 $\mathbf{x}_{n+1} := \mathbf{x}_n + \alpha_n (\mathbf{u}_n + \mathbf{q}_{n+1});$ 
 $n := n + 1;$ 
end

```

For a $N \times N$ problem, BCGS was theoretically shown to terminate in at most N steps. For solving convection-diffusion problems, [Sonneveld, 1989] showed that the average work in two dimensions is proportional to $N^{1.5}$. Although this is comparable to the SOR performance, BCGR is far more robust, the proportionality coefficient is much smaller, and there is no need for critically important parameters (i.e. the overrelaxation coefficient).

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.

2.5. The Generalized Minimum Residual (GMRES) Method

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

$$\mathbf{K}(m) = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \mathbf{A}^3\mathbf{r}_0, \dots, \mathbf{A}^{m-1}\mathbf{r}_0\} \quad (7)$$

where $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ is the initial residual and n is the number of iterations for a sufficiently accurate solution. The problem is then reduced to the solution of the n -dimensional least squares problem, which is usually significantly smaller than the original N -dimensional problem. GMRES is guaranteed to converge to the true solutions in $n < N$ iterations. If a full orthogonalization approach is used, the method becomes prohibitively expensive because storage requirements increase with n and the number of multiplications with Nn^2 . To avoid this difficulty, a maximum value of n , denoted by m , much smaller than N is chosen and the algorithm is applied iteratively. If the convergence criterion is not met within m iterations, the iteration can be restarted using as a starting value of \mathbf{x} the one obtained at the m -th iteration of the previous cycle. This algorithm is referred to as the GMRES(m) algorithm, and is described as follows:

```

 $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0;$ 
 $\mathbf{v}_1 := \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|};$ 
 $i = 0;$ 
while residual > tolerance do
begin
  for  $n = 1, 2, \dots, m$  do
     $h_{k,n} := \mathbf{A}\mathbf{v}_n \mathbf{v}_k, k = 1, 2, \dots, n;$ 
     $\tilde{\mathbf{v}}_{n+1} := \mathbf{A}\mathbf{v}_n - \sum_{k=1}^n h_{k,n} \mathbf{v}_k;$ 
     $h_{n+1,n} := \|\tilde{\mathbf{v}}_{n+1}\|;$ 
     $\mathbf{v}_{n+1} := \frac{\tilde{\mathbf{v}}_{n+1}}{h_{n+1,n}};$ 
  end do
   $\mathbf{x}_m := \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m;$ 
   $\mathbf{r}_m := \mathbf{b} - \mathbf{A}\mathbf{x}_m;$ 
   $\mathbf{x}_0 := \mathbf{x}_m; \mathbf{v}_1 := \frac{\mathbf{r}_m}{\|\mathbf{r}_m\|};$ 
   $i := i + 1;$ 
end

```

The term \mathbf{V}_n denotes the $N \times n$ matrix whose columns are the l_2 -orthonormal basis $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$. The vector \mathbf{y}_m minimizes $\|\beta \mathbf{e}_1 - \bar{\mathbf{H}}_m \mathbf{y}\|$, in which $\beta = \|\mathbf{r}_0\|$ and $\bar{\mathbf{H}}_m$ a $(n+1) \times n$ matrix whose only non-zero entries are the elements $h_{k,n}$ generated by the method.

Hence, in GMRES(m) we solve the least squares problem

$$\min_{\mathbf{y}} \|\beta \mathbf{e}_1 - \bar{\mathbf{H}}_m \mathbf{y}\|, \quad (8)$$

which is accomplished using QR factorization. This is simple to implement because of the special structure of $\bar{\mathbf{H}}_m$, and allows the updating of its factorization progressively as each column appears, thus making it possible to obtain the residual norm of the approximate solution without having to compute the intermediate solution or the residuals. When this "natural" stopping test is used, the algorithm enables the user to decide when to stop the process without wasting needless operations.

The residual norm of GMRES(m) is minimized at every iteration, thus resulting in a predictable monotonic decrease (or at least non-increase) of the residuals. An important property of GMRES(m) is that the algorithm does not break down, but it may stagnate (e.g. in the case of indefinite non-symmetric \mathbf{A} matrices). If the solution converges, this is a very reliable solver.

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). For $m = 30$, the memory requirements of DSLUGM are about four times larger than those of DSLUBC and DSLUCS.

2.6. Preconditioning

Ordinary CG methods have a satisfactory rate of convergence if they are well conditioned, i.e. they are arithmetically "close" to the identity matrix. However, ill-conditioning is the rule rather than the exception in most matrices arising from TOUGH2 simulations. The convergence of CG algorithms turns out to be superlinear [Axelsson, 1977], with a convergence rate depending on the distribution of smallest eigenvalues of \mathbf{A} rather than on the spectral condition number [Sonneveld, 1989].

In order to improve the distribution of the smallest eigenvalues of \mathbf{A} and/or decrease the spectral condition number, *preconditioning* techniques are commonly applied. This suggests applying the CG methods not to the original equations but to their preconditioned form

$$(\mathbf{P}^{-1} \bullet \mathbf{A}) \bullet \mathbf{x} = \mathbf{P}^{-1} \bullet \mathbf{b} \quad (9)$$

A good preconditioning matrix \mathbf{P} should have two properties: (a) it should approximate the inverse of \mathbf{A} , in the sense that

$$\mathbf{P}^{-1} \bullet \mathbf{A} \approx \mathbf{1}, \quad (10)$$

(\mathbf{I} is the identity matrix) and the matrix in equation (9) is better conditioned than the original matrix \mathbf{A} , and (b) the matrix \mathbf{P} should be much easier to solve using a CG method rather than the \mathbf{A} matrix. In developing a general description of preconditioners, we can define

$$\mathbf{P}^{-1} = \mathbf{P}_L \bullet \mathbf{P}_R \quad (11)$$

where \mathbf{P}_L and \mathbf{P}_R are often chosen to be inverses of sparse triangular matrices. Defining $\bar{\mathbf{x}}$ and $\bar{\mathbf{b}}$ by

$$\mathbf{P}_R \bullet \bar{\mathbf{x}} = \mathbf{x} \text{ and } \bar{\mathbf{b}} = \mathbf{P}_L \bullet \mathbf{b} \quad (12)$$

respectively, the preconditioned version of the system reads:

$$\mathbf{B} \bullet \bar{\mathbf{x}} = \bar{\mathbf{b}}, \quad (13)$$

where $\mathbf{B} = \mathbf{P}_L \mathbf{A} \mathbf{P}_R$. There are several possibilities for implementing the preconditioning matrices. If *diagonal scaling* is used for preconditioning, then

$$\mathbf{P}_L = \mathbf{P}_R = \mathbf{D}^{-1/2}, \quad (14)$$

where \mathbf{D} is the main diagonal of \mathbf{A} . As already discussed, this type of preconditioning invariably produced very poor results in all TOUGH2 tests. An important class of preconditioners is based on *incomplete LU (ILU) factorization* of the matrix \mathbf{A} , which can be obtained by a slightly modified Gaussian elimination procedure and results in

$$\mathbf{L} \bullet \mathbf{U} = \mathbf{A} + \mathbf{E}, \quad (15)$$

where \mathbf{L} is a lower triangular matrix, \mathbf{U} is an upper triangular matrix, and \mathbf{E} a very sparse deviation matrix. Several possibilities of right, center, and left preconditioning exist with ILU preconditioning. SLAP uses central ILU preconditioning, in which

$$\mathbf{P}_L = \mathbf{L}^{-1} \text{ and } \mathbf{P}_R = \mathbf{U}^{-1}. \quad (16)$$

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 \mathbf{A} matrix (determined by the integrated finite difference formulation of TOUGH2) to develop an optimized *Incomplete Line LU* factorization (ILLU) preconditioner [Sonneveld, 1989]. The ILLU 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. In this case

$$\mathbf{P}_L = (\mathbf{L} + \mathbf{D})^{-1} \text{ and } \mathbf{P}_R = (\mathbf{U} + \mathbf{D})^{-1} \mathbf{D} \quad (17)$$

In our tests we confirmed Sonneveld's [1986] observation that the ILLU factorization has the advantage of being less sensitive to special directions in the problem (e.g. the advection direction in the advection-diffusion equation, layering, etc.).

3. DATA REQUIREMENTS AND USE OF T2CG1

T2CG1 allows four solver options: the direct solver MA28 [Duff, 1977], the BCG routine called DSLUBC, the BCGS routine called DSLUCS, and the GMRES routine called DSLUGM. The data requirements for the selection of the solver and the relevant convergence parameters are minimal. T2CG1 makes use of the previously unused parameter MOP(21) to select the solution method.

3.1. Solver Selection

MOP(21) is used to select the solution method. The following options are available:

MOP(21) =	1:	MA28 direct solver
	2:	DSLUBC (BCG solver)
	3:	DSLUCS (BCGS solver) - default
	4:	DSLUGM (GMRES solver)

The default is MOP(21)=3, i.e. the Bi-Conjugate Gradient Squared routine DSLUCS. The choice of the default was based on the robustness and speed of DSLUCS, which seemed to be the best overall performer on a wide variety of TOUGH2 problems.

All solvers share the same memory for matrix operations. For the same amount of available memory, DSLUCS and DSLUBC have the smallest memory requirements (the same for both routines) and can solve the largest size problems, which are slightly over 4 times larger than the size solvable by DSLUGM. MA28 has the largest memory requirements, and solves the smallest problems for the same amount of available memory. Due to the structure of MA28, its memory requirements are not well defined and depend on the problem dimensionality. Generally speaking, in two dimensions and three-dimensions the maximum size of problems solvable by MA28 is roughly 1/10th-1/20th and 1/20th-1/50th, respectively, of problems solvable by DSLUCS and DSLUBC.

3.2. Maximum Number of Iterations

The maximum number of allowable iterations NMAXIT for the CG methods to converge is given by the formula

$$NMAXIT = \max \left\{ 20, \frac{N}{10} \right\} \quad (18)$$

where N is the order of the matrix, i.e. the total number of equations. This formula provides enough iterations for the efficient, and accurate solution of practically all problems we have simulated using the TOUGH2 family of codes. Although CG methods are guaranteed to converge within N iterations in the absence of roundoff errors, the above formula reflects the practical point that if the number of iterations exceeds 10% of N , the solver in use is not recommended for the type of problem under investigation.

Currently the user has no control over the value of NMAXIT through data inputs. In the rather unlikely event of extremely ill-conditioned matrices, the user may want to increase the number of allowable iterations by modifying the source code. In APPENDIX A we provide information on how to accomplish this. For most problems it is inadvisable

to use $NMAXIT > 0.25 N$. If convergence is not achieved within the $NMAXIT \leq 0.25 N$ iterations, this is a good indication that the CG method being used is not a good choice.

3.3. Convergence Criterion

The convergence criteria used are method-specific. For the BCG and BCGS methods (i.e. routines DSLUBC and DSLUCS) the iterations after n steps stop when

$$\frac{\|\mathbf{P}^{-1}\mathbf{r}_n\|_2}{\|\mathbf{P}^{-1}\mathbf{b}\|_2} \leq \text{CLOSUR}. \quad (19)$$

In DSLUGM the iterative process stops after the n -th iteration when

$$\frac{\|\mathbf{S}_B\mathbf{r}_n\|_2}{\|\mathbf{S}_B\mathbf{b}\|_2} \leq \text{CLOSUR} \text{ or } \frac{\|\mathbf{S}_B\mathbf{P}^{-1}\mathbf{r}_n\|_2}{\|\mathbf{S}_B\mathbf{P}^{-1}\mathbf{b}\|_2} \leq \text{CLOSUR} \quad (20)$$

for right and left preconditioning respectively, where \mathbf{S}_B is a diagonal scaling vector.

The convergence criterion of the CS solvers is set at $\text{CLOSUR} = 10^{-6}$. The default value of CLOSUR is adequate for most problems. In our experience with hundreds of TOUGH2 simulations, only once have we encountered a case where the convergence criterion of 10^{-6} was insufficient. This may occur when the \mathbf{A} matrix is ill-conditioned and the elements in each row differ by several orders of magnitude (e.g. in cases of extreme heterogeneity, strongly non-linear processes, etc.). In this case it may be necessary to tighten the convergence criterion if divergence of the outer Newtonian iterations is to be avoided, but the largest possible value of CLOSUR should be used in order to avoid wasteful extra iterations.

The closure criterion may also need to be changed if the convergence criterion of the Newtonian iterations (parameter RE1 in record PARAM.3) is reduced below its default value of 10^{-5} . In this case it is a good practice to maintain the relationship

$$\text{CLOSUR} = \frac{\text{RE1}}{10}.$$

Although the user has no control over the value of CLOSUR through the input data set, it is quite simple to modify the source code to allow for a tighter convergence criterion. In APPENDIX A we show how this can be accomplished. However, the change in the convergence criterion should not be attempted unless absolutely necessary. It must be kept in mind that under no circumstances should CLOSUR be set in the vicinity of machine accuracy for the computing platform being used. If the user sets $\text{CLOSUR} < 10^{-12}$, then CLOSUR is internally reset to 10^{-12} .

3.4. Solver-Related Output

TOUGH2 supplies information on the CG solver performance in the LINEQ output file. A typical output is shown below:

AT [1, 2] DELT= .900000E+03 IERR=0& ERR= .608102E-06 IT= 88 ITC= 175

This example shows that in the second Newtonian iteration (ITER = 2) of the 1st timestep (KCYC = 1) when the timestep DELT = 900 s, the CG solver successfully (IERR = 0) solved the matrix in IT = 88 iterations, and the error (as described in the convergence criterion in Subsection 3.3) $ERR = 6.08102 \times 10^{-7}$. The cumulative number of CG iterations up to that point was ITC = 175.

The error code IERR may have different meanings depending on the CG solver. If the DSLUBC routine is used, then the following outputs are possible:

IERR = 0: the matrix was solved successfully
 IERR = 2: the method failed to converge in NMAXIT steps
 IERR = 4: the convergence criterion CLOSUR is too tight, and is reset to 10^{-12}
 IERR = 5: the preconditioning matrix is not positive definite
 IERR = 6: the matrix **A** is not positive definite
 IERR = 7: the incomplete factorization was not successful, stop-gap measures were enacted, but the resulting preconditioning is sub-optimal.

If the DSLUCS routine is used, IERR may have the following values and meanings:

IERR = 0: the matrix was solved successfully
 IERR = 2: the method failed to converge in NMAXIT steps
 IERR = 4: the convergence criterion CLOSUR is too tight, and is reset to 10^{-12}
 IERR = 5: the solution method is breaking down
 IERR = 6: stagnation is detected
 IERR = 7: the incomplete factorization was not successful, stop-gap measures were enacted, but the resulting preconditioning is sub-optimal.

If the DSLUGM routine is used, the following IERR options are possible:

IERR = 0: the matrix was solved successfully
 IERR = 2: the method failed to reduce the norm of the current residual on its last call, and the method is stalled.

4. TEST PROBLEMS

We tested the performance of the T2CG1 package of conjugate gradient solvers in 16 problems covering the range from one-dimensional isothermal single phase flow to three-dimensional non-isothermal three-phase flow and transport with phase appearance and disappearance. In the following subsections we present a detailed discussion of each problem and of the performance of the three solvers. In all test problems the performance indicators of the various solvers are tabulated, and the results for the fastest routine are printed in boldface. A very good measure of the efficiency of a particular CG method (or any iterative method) is given by the number of CG iterations, N_{CG} , (i.e. the number of matrix-vector multiplications) vis-a-vis the number of equations, N , i.e. the order of the matrix. Although CG methods theoretically converge to the true solution within at most N iterations, in practice a CG method is deemed efficient if it can produce a solution within a prescribed tolerance for $N_{CG}N \leq 0.1$.

4.1. Test Problem 1: The Theis Problem (Radial 1-D, Single-Phase, Isothermal)

4.1.1. Problem description

The first problem represents the one-dimensional radial flow toward a well of radius $r_w \rightarrow 0$ in a homogeneous circular aquifer with infinite-acting boundaries. The analytical solution to this problem is the well known *Theis* [1935] solution. The aquifer is $Z = 50$ m thick, has a transmissivity of 1000 m²/day, a storage coefficient $S = 10^{-4}$, and water is pumped at a rate of 1000 m³/day. In terms of the TOUGH2 data requirements, in the aquifer the permeability is $k = 2.3671 \times 10^{-11}$ m², the porosity is $\phi = 0.35$, the soil compressibility is 6.922×10^{-11} Pa⁻¹ (assuming that the water compressibility is constant and equal to 4.5479×10^{-10} Pa⁻¹), and the temperature is 20 °C. The head distribution is sought at $t = t_{max} = 10$ days.

The EOS1 fluid property module with $NK = 1$ and $NEQ = 1$ was used for the TOUGH2 simulation. The well radius is $r_w = 10^{-3}$ m, and the outer boundary extends to $100,000$ m, which is sufficient for the aquifer to act as infinite. A total of $N = 104$ equations are solved, i.e. the domain is subdivided into 104 gridblocks. Figure 1 shows a schematic of the simulated aquifer, along with pertinent information. More information on this TOUGH2 simulation can be found in the input file "PROB1", which is stored on the diskette accompanying this report.

Figure 2 shows the analytical solution and the TOUGH2 solution obtained for an aquifer temperature of 20 °C. The two solutions are identical. It must be noted that in this and other similar problems of single-phase flow in confined aquifers (where the compressibility is small and the only variable changing is pressure) the convergence criterion for the relative error in the Newtonian iterations must be set to a sufficiently small number. More specifically, the parameter RE1 in Record PARAM.3 of the PARAM data block must be reset from its default value of 10^{-5} to a number several orders of magnitude smaller. The maximum RE1 value should not be larger than 10^{-8} , and the CG convergence criterion CLOSUR must be reset accordingly (see Subsection 3.3).

4.1.2. Solver performance

The solver performance is summarized in Table 1. The second column gives the number of timesteps Δt required to cover the desired simulation time $t_{max} = 10$ days. The total number of Newtonian iterations needed to complete the simulation appears in the third

column. The fourth and fifth columns show the maximum and minimum number of CG iterations needed for the convergence of the CG solutions, i.e. of each Newtonian iteration. The CPU time requirements are shown in the sixth column.

The term N_{CG} in Table 1 (as well as in all subsequent tables) denotes the number of conjugate gradient (CG) iterations needed to reach or exceed the convergence criterion (see Subsection 3.3). Inspection of Table 1 reveals that the CPU time is practically identical for all the CG solvers. A single CG iteration suffices for the solution, which was expected due to the simple tridiagonal matrix arising from this problem. This similar performance is attributed to the very small (almost trivial) size of the problem.

For this one-dimensional problem (i.e. a tridiagonal matrix) with only $N = 104$ equations to solve, the potential superiority of the iterative methods over the direct solver cannot be realized because of the additional overhead needed to set up the CG solutions. For the same reason it is not possible to identify a CG routine with a clear superiority over the other iterative solvers.

Table 1. Solver Performance in Test Problem 1					
Number of Equations: 104			Computing Platform: Apple Macintosh QUADRA 800		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	11	33	-	-	7.57
DSLUBC	11	33	1	1	7.87
DSLUCS	11	33	1	1	7.53
DSLUGM	11	33	1	1	7.55

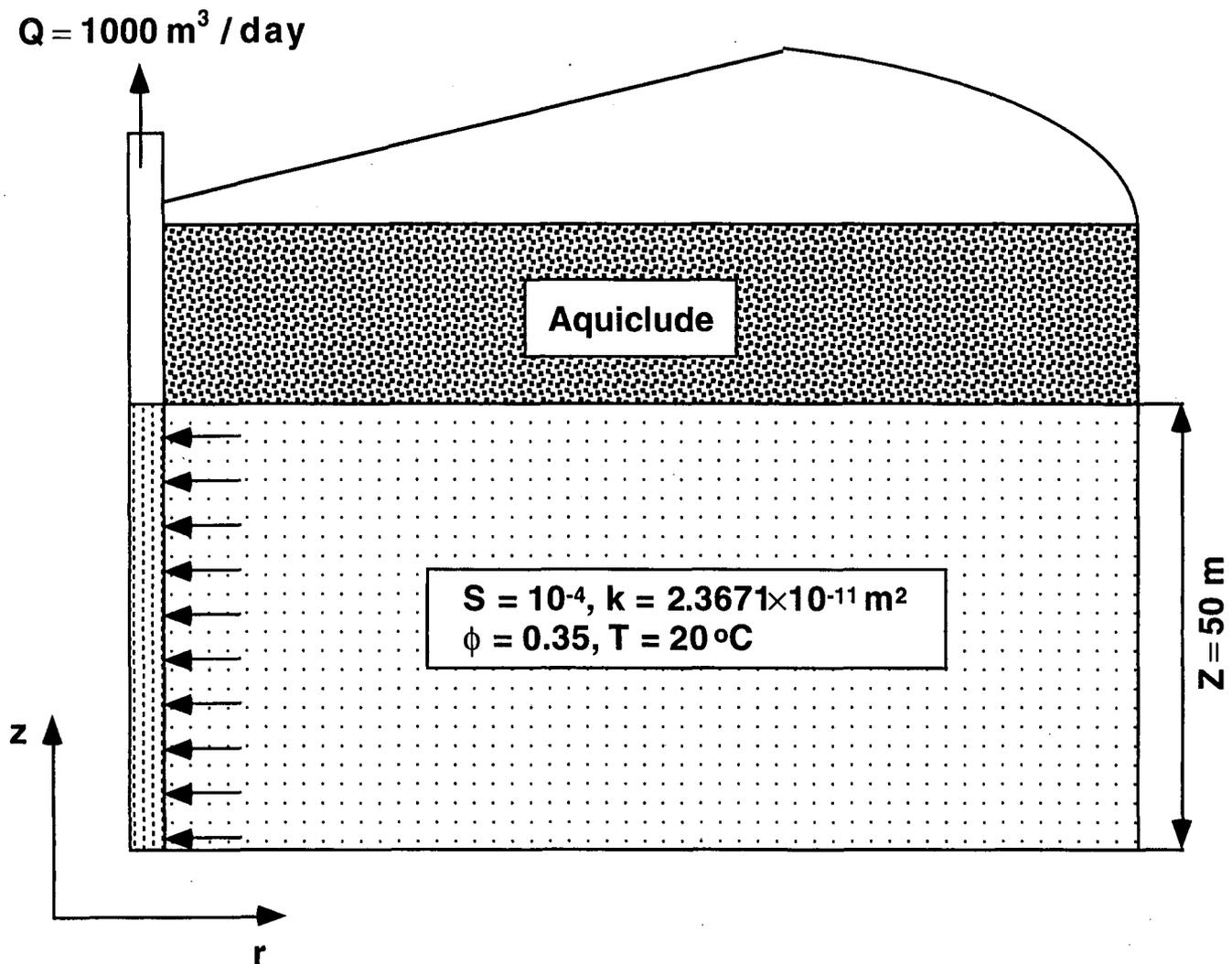


Figure 1. The Theis problem: radial flow to a single well.

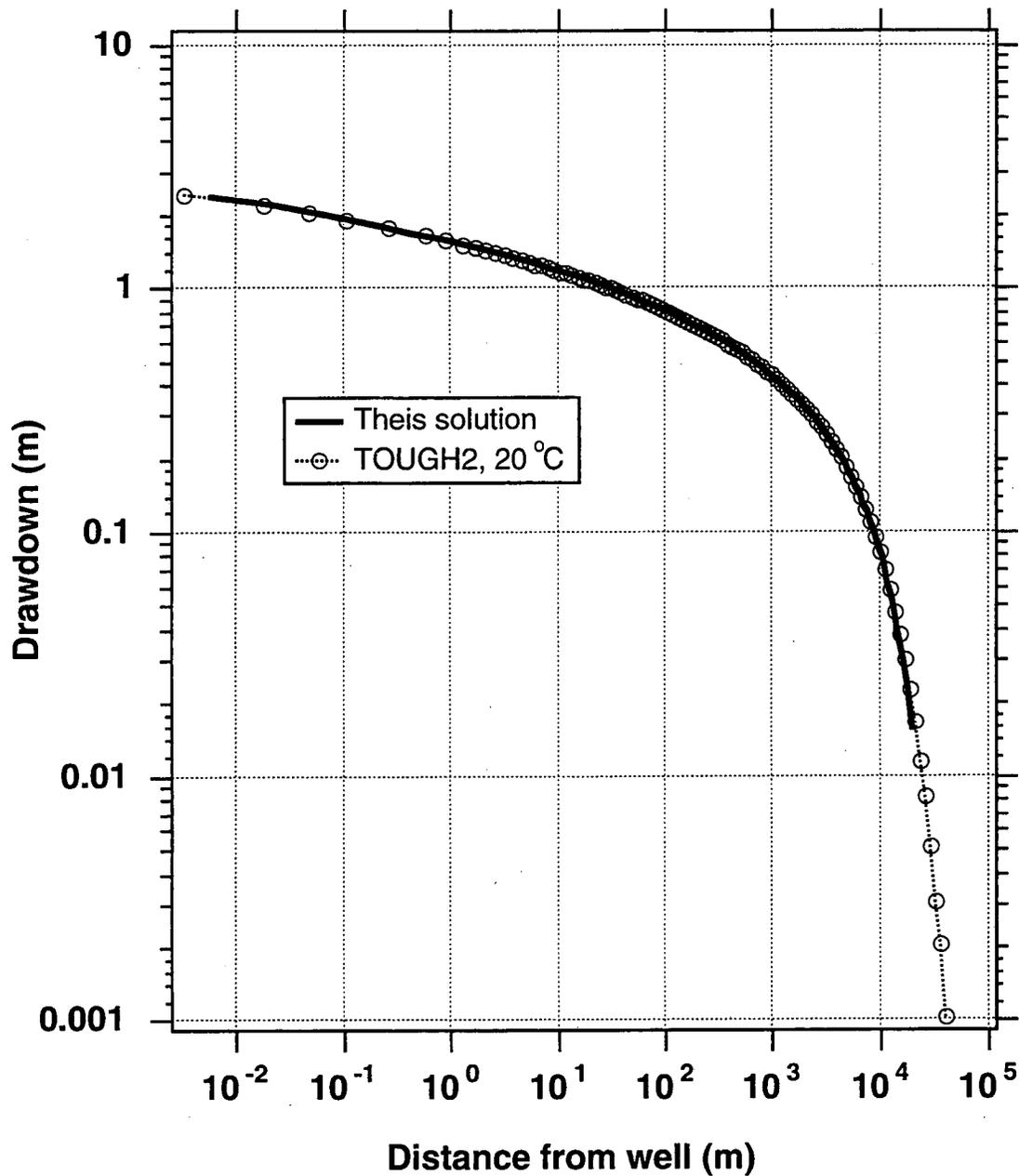


Figure 2. Comparison of the analytical and the TOUGH2 solutions to the Theis problem at $t = t_{max} = 10$ days.

4.2. Test Problem 2: Radial Heat Transport (Radial 1-D, Single-Phase, Non-Isothermal)

4.2.1. Problem description

The radial heat transport was originally solved analytically by Avdonin [1964], and was later described by Ross *et al.* [1982]. Moridis and Pruess [1992] discussed in detail the performance of TOUGH in the solution of this problem. Cold water is injected into a semi-infinite, high-temperature aquifer. The overburden and underburden are impermeable to mass and heat flow, acting as no-flow and adiabatic boundaries and reducing the governing heat transport equation to that of convection and diffusion. Figure 3 shows a schematic of the problem and other pertinent information.

The properties and conditions are shown in Fig. 3. The radius of the aquifer is 1000 m (sufficient to approximate the semi-infinite behavior), at which a constant temperature (equal to the initial temperature of 170 °C) is imposed. The TOUGH2 predictions of the temperature distribution are sought at $t = t_{max} = 10^9$ sec after the initiation of the cold water injection for comparison to the analytical solution.

This simulation was performed using the EOS1 fluid property module with $NK = 1$ and $NEQ = 2$. We used 127 gridblocks, resulting in a total of $N = 254$ equations. The input data file for this problem appears under the name "PROB2" on the diskette accompanying this document. The grid was generated with the MESHMAKER facility built into the TOUGH2 code [Pruess, 1991]. The data for MESHMAKER are attached to the bottom of the PROB2 file. A very large volume ($10^{38} m^3$) was assigned to the single boundary gridblock in the radial direction (127th), thus ensuring constant boundary pressures and temperatures during the duration of the simulation.

Water at a temperature of 160 °C and a rate of 10 kg/sec is injected directly into the first gridblock located next to the wellbore. With flow rate directly specified through a mass source, the problem becomes insensitive to the precise value of permeability; we used a value of $k = 6.6646 \times 10^{-13} m^2$. The soil density and soil specific heat were 2,899 kg/m³ and 740 J/kg-°C respectively. The maximum allowed time step size was $\Delta t = 10^6$ sec, i.e. 1/1000th of the maximum.

Figure 4 shows an excellent agreement between the analytical and the TOUGH2 solutions at $t = 10^9$ sec. The problem under discussion has a "similarity solution" in terms of the variable r^2/t [O'Sullivan, 1981; Doughty and Pruess, 1990; 1992]. The temperature changes in space and time are coupled in such a way that they depend solely on r^2/t , rather than on r and t separately. In Figure 5 we plot the analytical and the TOUGH2 solutions at $t = 10^9$ sec versus the similarity variable r^2/t . The TOUGH2 results are consistent with the r^2/t invariance that is known to exist in this problem. At $t_{max} = 10^9$ sec the TOUGH2 solution virtually coincides with the analytical solution.

4.2.2. Solver performance

Table 2 shows that there is very little (if any) practical difference between the performance of the MA28, DLUCS, and DSLUGM solvers, but the DSLUBC routine is measurably slower than the rest. It is interesting that the number of timesteps needed to cover t_{max} varies with the solver. All the CG solvers are very efficient, as demonstrated by the number of matrix solutions (i.e. matrix-vector multiplications) needed for convergence. With the exception of the first Newtonian iteration in the first, second, and third timestep, a single iteration suffices for the CG solution.

This is attributed to the simple penta-diagonal matrix arising from this problem. It must be kept in mind that Test Problem 2 represents a small matrix solved many (over 1300) times. As in Test Problem 1, the CG solvers do not exhibit any advantage over the direct solver because of the very small (almost trivial) size of the problem. With a penta-diagonal matrix of order $N = 254$ only, the potential superiority of the iterative methods over the direct solver cannot be clearly demonstrated because of the additional overhead needed to set up the CG solutions. For this reason, it is not possible to observe the theoretically predicted 2:1 relative speed of the DSLUCS routine over the DSLUBC solver.

It is interesting to notice that the MA28 solution may require more timesteps than some of the CG solvers. This is due to the fact that in one-dimensional problems with simple tri- and penta-diagonal matrices the CG solutions are not iterative since the preconditioning in essence produces the direct solutions (thus requiring a single iteration for convergence). These may be more accurate than the MA28 solutions due to fewer operations and smaller roundoff error. When this happens fewer Newtonian iterations are needed for convergence, which allows faster growth of the timestep size and thus reduces the total number of Δt 's to cover the simulation period. It appears that in this problem the DSLUBC algorithm produces a slightly less accurate solution than MA28, thus requiring a few extra timesteps.

Table 2. Solver Performance in Test Problem 2					
Number of Equations: 254			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	1,361	9,859	-	-	526
DSLUBC	1,367	9,865	4(*)	1	575
DSLUCS	1,327	9,298	3(†)	1	533
DSLUGM	1,334	9,420	3(+)	1	537

(*): 4, 3 and 2 CG iterations occur only once. In all other cases only 1 iteration is needed.

(†): Occurs only once. In all other cases only 1 CG iteration is needed.

(+): 3 and 2 CG iterations occur only once. In all other cases only 1 iteration is needed.

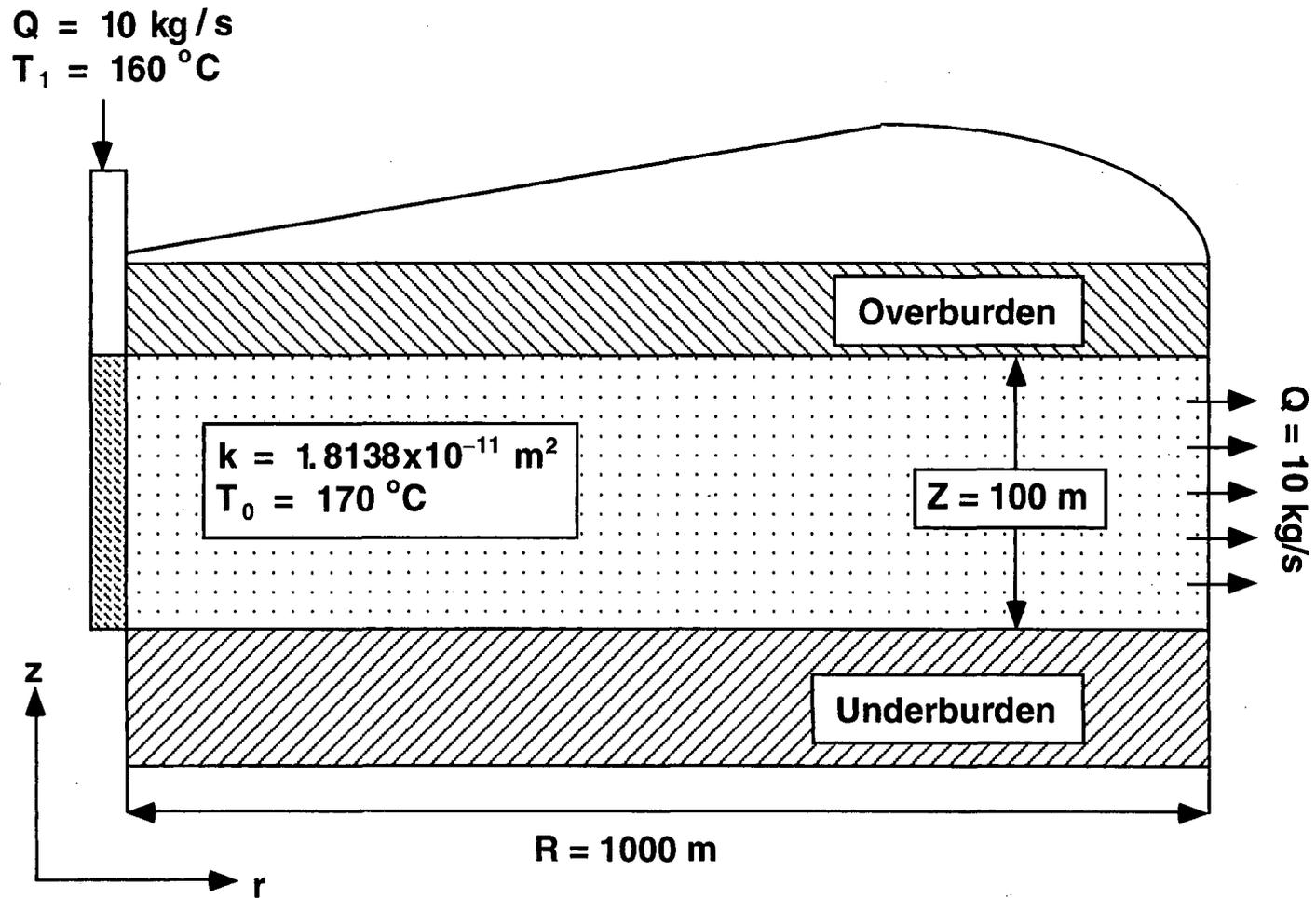


Figure 3. The radial heat transport system in Test Problem 2.

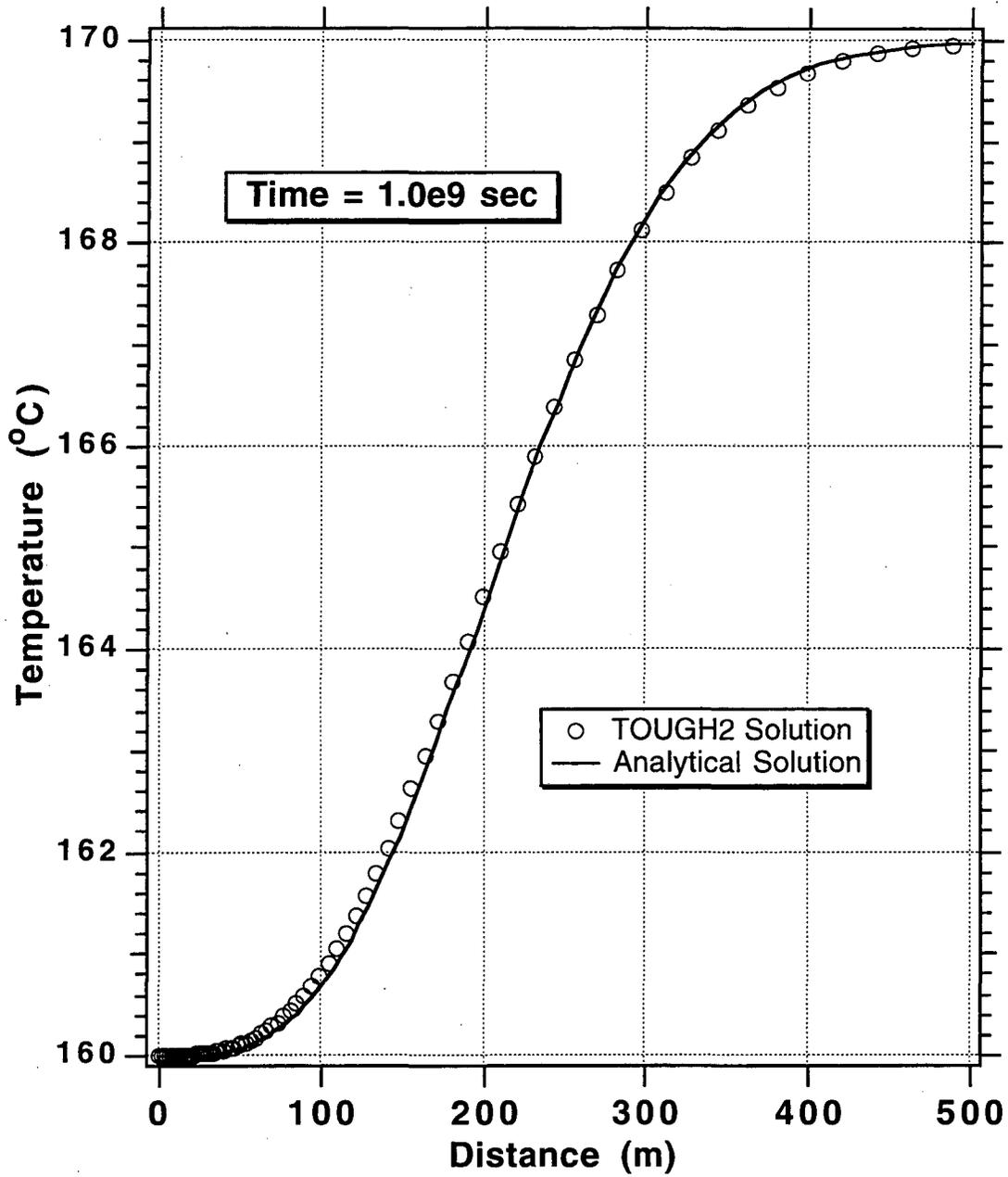


Figure 4. Comparison of the analytical and TOUGH2 solutions to the radial heat transport problem (Test Problem 2).

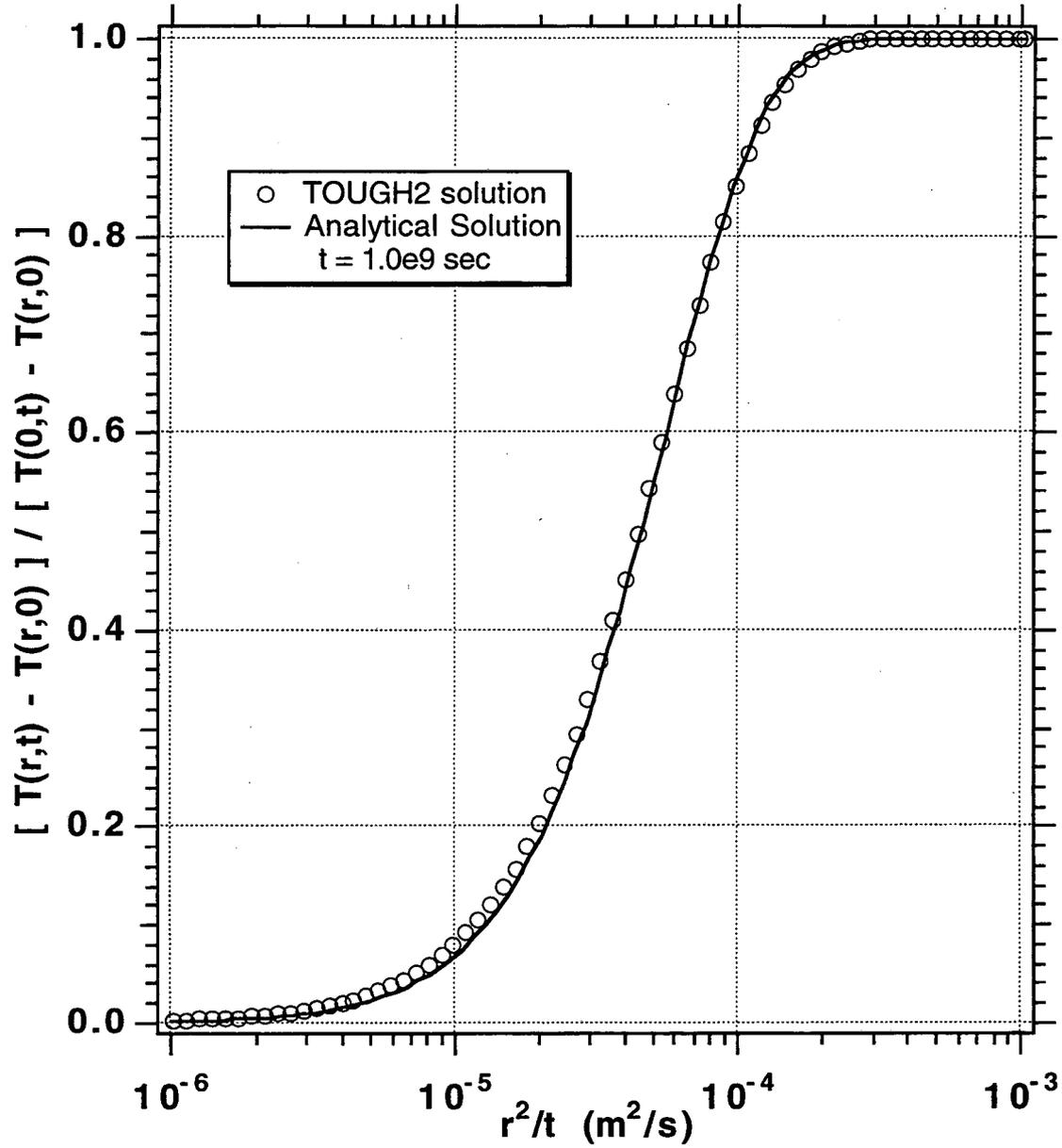


Figure 5. Analytical and TOUGH2 similarity solutions in Test Problem 2.

4.3. Test Problem 3: Convection Cell Experiment (Cylindrical 2-D, Single-Phase, Non-Isothermal)

4.3.1. Problem description

The third test problem involves a laboratory convection cell experiment described, performed, and modeled by Reda [1984]. Figure 6 shows a schematic of the experiment. A porous medium consisting of glass beads with an average diameter of 0.65 mm fills the annular region between the two vertical concentric cylinders. Application of heat generates a thermal buoyancy force, giving rise to the development of convection cells. Figure 7 provides information on the system geometry, properties, as well as initial and boundary conditions. This problem has been used to test the ability of TOUGH to simulate transient two-dimensional simultaneous heat and mass flow, and has been discussed in detail by Moridis and Pruess [1992].

To evaluate TOUGH2, numerical predictions are compared to measurements of the thermal response of the surface of the inner cylinder at the elevations $z = 0.4192, 0.6288, 0.7336, \text{ and } 0.7860 \text{ m}$ (i.e. $z/\Delta r = 2, 3, 3.5, \text{ and } 3.75$). These comparisons are made over a period of $t = t_{max} = 10^5 \text{ sec}$ for a power level of 278.3 W/m . The strong variation of temperature in the immediate vicinity of the heater, combined with the need to include permeability-enhancement effects in our simulations, necessitated a grid with a sufficiently fine discretization in the all-important region near the heater.

The first radial grid block extends to 9.5 mm , i.e. the radius of the heater (as well as of the nylon rods placed above and below). The next 5 radial gridblocks have a radial increment $\Delta r = 0.65 \text{ mm}$ each, corresponding to the mean glass bead size, d . Besides providing the desired fine spatial resolution near the heater, these blocks are also used to represent the enhanced permeability in the $5d$ zone affected by wall effects. The next 4 gridblocks have progressively larger Δr 's and reach $r = 0.05 \text{ m}$. These are followed by 5 more blocks with logarithmically increasing spacing, and a boundary gridblock with a $\Delta r = 10^{-5} \text{ m}$. This discretization results in 16 gridblocks in r . Layer thickness in the region of the heater was chosen as $\Delta z = 0.02096 \text{ m}$. Above and below the 12 grid layers representing the heater region there are 3 layers with the same $\Delta z = 0.02096 \text{ m}$, followed by a coarser discretization. The top layer with a $\Delta z = 10^{-5} \text{ m}$ is used to represent the top boundary conditions. The MESHMAKER facility of TOUGH2 was used to generate the grid. The specifications for the grid generation appear at the end of the input file "PROB3a" (provided on the accompanying diskette) in the MESHMAKER data block.

We ran two simulations. The first did not consider permeability enhancement, and the medium had the properties of the domain SOIL1 (see input data file "PROB3a"), which represents the unenhanced "background" permeability of $k_0 = 6.1 \times 10^{-10} \text{ m}^2$. In the second simulation (see input data file "PROB3b") we used nine different sub-domains of porous media in the simulation. Sub-domain "SOIL1" still represents the unenhanced "background" permeability. Soil types "SOIL2" through "SOIL6" represent the five sub-domains with channeling-induced enhanced permeability in the vicinity of the heater. Following Reda [1984], permeability in these domains is linearly interpolated between k_0 and $2k_0$ over the $\Delta r = 3.25 \text{ mm}$ region starting at the internal radius $r_i = 9.5 \text{ mm}$. The soil density and specific heat of the six domains "SOIL1" through "SOIL6" are 2500 kg/m^3 and $740 \text{ J/kg}\cdot^\circ\text{C}$ respectively.

Domain "BOUND" is used to represent the top permeable boundary of the model and has the same properties as "SOIL1", with the exception of specific heat which is $10^4 \text{ J/kg}\cdot^\circ\text{C}$ for this domain. This value is chosen to exclude the large volume blocks in domain BOUND from the mass and energy balances.

The gridblocks at the outer radial boundary belong to the domain "TCOND", which has a very low porosity (10^{-5}), zero permeability and a very high specific heat ($10^{54} \text{ J/kg-}^\circ\text{C}$). This forces strict "no mass-flow" boundaries while maintaining a constant temperature. The gridblocks assigned to the heater belong to the domain "HEATR", which is characterized by a very low porosity (10^{-5}), zero permeability to impose a "no-mass-flow" boundary, and the properties of cast iron, i.e. a density of 7400 kg/m^3 , a specific heat of $569.4 \text{ J/kg-}^\circ\text{C}$, and a thermal conductivity of $50.2 \text{ W/m-}^\circ\text{C}$ [Carlsaw and Jaeger, 1959]. However, the use of cast iron is not expected to have a significant effect on the simulation, and is implemented only in the spirit of conceptual consistency. An important aspect is the inclusion of a large compressibility (10^{-5} Pa^{-1}) for the tiny pore space in the "HEATR" domain. This feature prevents the build-up of excessive pore pressures in this domain due to thermal expansion of liquid water. The "HEATR" domain could have also been initialized as containing gas, which at ambient temperature has a large compressibility of approximately 10^{-5} Pa^{-1} .

The initial pressure and temperature conditions for the TOUGH2 run are $2.0 \times 10^5 \text{ Pa}$ and 21.5°C respectively. No separate initial step of gravity equilibration was deemed necessary because it was determined that this process occurs extremely fast. More specifically, the pressure diffusivity $D = k/(c\phi\mu)$ is of order $10^3 \text{ m}^2/\text{sec}$, so that pressure equilibration would require only a fraction of a second.

For this simulation we used the EOS1 fluid property module with $\text{NK} = 1$ and $\text{NEQ} = 2$. The 416 gridblocks resulted in a total of $N = 832$ equations. Complete data input files appear on the diskette under the names "PROB3a" and "PROB3b" (without and with channeling effects respectively).

We compare the simulation results to the experimental measurements made by Reda [1984] at the locations $r = r_i$ and $z/\Delta r = 2, 3, 3.5$, and 3.75 over time. The point at $z/\Delta r = 2$ marks the bottom of the heater and lies at the interface between layers of equal thickness; similarly, the point at $z/\Delta r = 3$ marks the top of the heater and lies at the interface between layers of equal thickness. The points at $z/\Delta r = 3.5$ and $z/\Delta r = 3.75$ lie above the top of the heater. Temperature predictions at these points are obtained by interpolation. The maximum simulation time is $t_{max} = 10^5 \text{ sec}$.

The evolution of temperature over time at the prescribed locations is presented in Figures 8 and 9. Figure 8 shows the experimental measurements and the numerical predictions of the temporal variation of temperature at the bottom and the top of the heater. The numerical results with and without permeability enhancement are shown here. A very good agreement between experiment and prediction is observed for the period of transient convection, as well as for the eventual steady state.

The numerical results in Figure 8 are obtained at the center of the second gridblock at $r = 9.825 \text{ mm}$, i.e. 0.325 mm away from the inner radius r_i . Although this distance appears to be very small, our simulation indicates an extreme sensitivity of temperature to the radial distance r in the immediate vicinity of the inner radius. The temperature at r_i is also shown in Figure 8 as the upper limit of the range of the temperature at the points in the graphs. It can be seen that within 1 mm the temperature drops by as much as 2°C . The very strong dependence of temperature on the radial distance suggests that the predicted surface temperature lies between the two temperatures at $r = 9.5 \text{ mm}$. and $r = 9.825 \text{ mm}$.

Comparison between predictions and observations and experiments at the points $r = r_i$ and $z/\Delta r = 3.5$, and 3.75 (i.e. above the top of the heater) in Figure 9 shows reasonable

agreement. A far less dramatic dependence of temperature on radial distance r is expected at these two elevations z because they do not lie immediately next to the heater.

From these results it is evident that permeability enhancement has a significant impact on the temperature distribution. Temperatures predicted without permeability enhancement are consistently higher both at the bottom and the top of the heater. This temperature differential is small initially, but keeps increasing during the transient period and stabilizes as the steady-state is approached at 3 - 3.1 °C. The temperature differential appears earlier and is more pronounced at the top of the heater. This indicates that without flow channeling there is a somewhat slower initiation of convection, and a weaker convection process at later times when steady-state is approached.

However, it must be pointed out that despite its apparent better agreement with measurements (at least near steady-state), it is simply inappropriate to state that neglecting channeling effects produces more accurate results because of extremely steep temperature gradients in the vicinity of the heater. This hyper-sensitivity has been demonstrated and discussed above. It is obvious that even minute (i.e. sub-millimeter) changes in the position of the measuring device (sampling point) can effect sizable temperature differentials. This observation from our simulations adds further perspective on the comparison between experimental and numerical results. A more detailed discussion can be found in *Moridis and Pruess* [1992].

4.3.2. Solver performance

Table 3 shows the performance of the solvers on two different computing platforms: a Macintosh QUADRA 800 and an IBM RS/6000 370 workstation. The single Macintosh run included permeability enhancement. A single output was provided at $t = 10^6$ sec, a fact which minimized the number of timesteps to cover t_{max} . Although only $N = 832$ equations are solved, the superiority of the CG solvers is becoming apparent. The CG routines are invariably faster than the direct solver, but the small size of the problem does not allow large gains in speed. The DSLUCS routine is the fastest of the CG solvers, followed by DSLUGM. DSLUCS is only 1.09 times faster than DSLUBC, significantly less than the theoretically estimated factor of 2. N_{CG} never exceeds 5% of N , demonstrating the efficiency of the CG routines.

In the workstation runs both cases (without and with channeling effects) were studied. In Table 3 the results are reported in the same order, and the results are separated by a '/'. Outputs were required at 14 times, thus increasing the number of timesteps to cover $t_{max} = 10^5$ sec. With channeling effects neglected, DSLUGM is still the fastest solver and DSLUCS the second fastest. With channeling effects included, the order is reversed and DSLUCS is the fastest routine. DSLUBC is the slowest routine, slower than the direct solver, and 33% slower than DSLUCS. The execution time requirements, CG iterations, and Newtonian iterations increase with the heterogeneity of the problem. When channeling effects are neglected, the number of Newtonian iterations is the same regardless of the solver, but vary when channeling effects are included. It is interesting to notice the relative performance of CG vs direct solvers on the two computing platforms. The gains in performance are more evident on the IBM RS/6000 370 workstation, where the fastest CG solvers realize higher gains in speed (versus the direct solver) than on the Macintosh.

In Figures 10 and 11 we demonstrate the evolution of the CG solutions, indicated by the relationship between the residuals (see subsection 3.3) and the number of iterations to convergence, in the first Newtonian iteration of the first timestep. In Figure 10 (no permeability enhancement) we see that the DSLUGM solution demonstrates a monotonic reduction in the residual, a performance consistent with the properties of the method.

DSLUCS combines an oscillating behavior with the fastest reduction in the residuals, and exhibits many (and sometimes significant) local peaks in the convergence performance. These local peaks are also observed in DSLUBC, but they are usually smaller in magnitude. DSLUCS demonstrates the steepest decline in error with the number of iterations, followed by DSLUGM, while DSLUBC is the slowest in error reduction.

Figure 11 (in which permeability enhancement is included) demonstrates a similar convergence performance. Although this is a more heterogeneous problem, at this timestep and Newtonian iteration fewer iterations are needed for convergence than in Figure 10, i.e. the problem is easier to solve. The error reduction in DSLUGM is smooth and monotonic. The evolution of the DSLUBC and DSLUCS solutions, while not monotonic, is significantly smoother and lacks the oscillations observed in Figure 10. The same pattern is observed: DSLUCS shows the strongest error reduction for the least number of iterations, and DSLUBC is the slowest.

Table 3. Solver Performance in Test Problem 3(*)					
Number of Equations: 832			Computing Platform: Apple Macintosh QUADRA 800		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	26	91	-	-	331
DSLUBC	26	91	31	2	321
DSLUCS	26	91	35	2	295
DSLUGM	26	91	41	11	299
Number of Equations: 832					
Computing Platform: IBM RS/6000 370					
SOLVER	Number of Δt 's (†)	Newtonian Iterations (†)	Maximum N_{CG} (†)	Minimum N_{CG} (†)	CPU Time (sec) (†)
MA28	37/38	127/137	-	-	48/52
DSLUBC	37/38	127/139	35/37	7/7	52/61
DSLUCS	37/38	127/135	30/30	7/7	39/42
DSLUGM	37/38	127/139	39/44	8/8	36/43

(*) The runs on the IBM RS/6000 370 required output at 14 times, while the Macintosh QUADRA 800 provided output only at $t_{max} = 10^6$ sec.

(†) The first number without, the second with channeling effects.

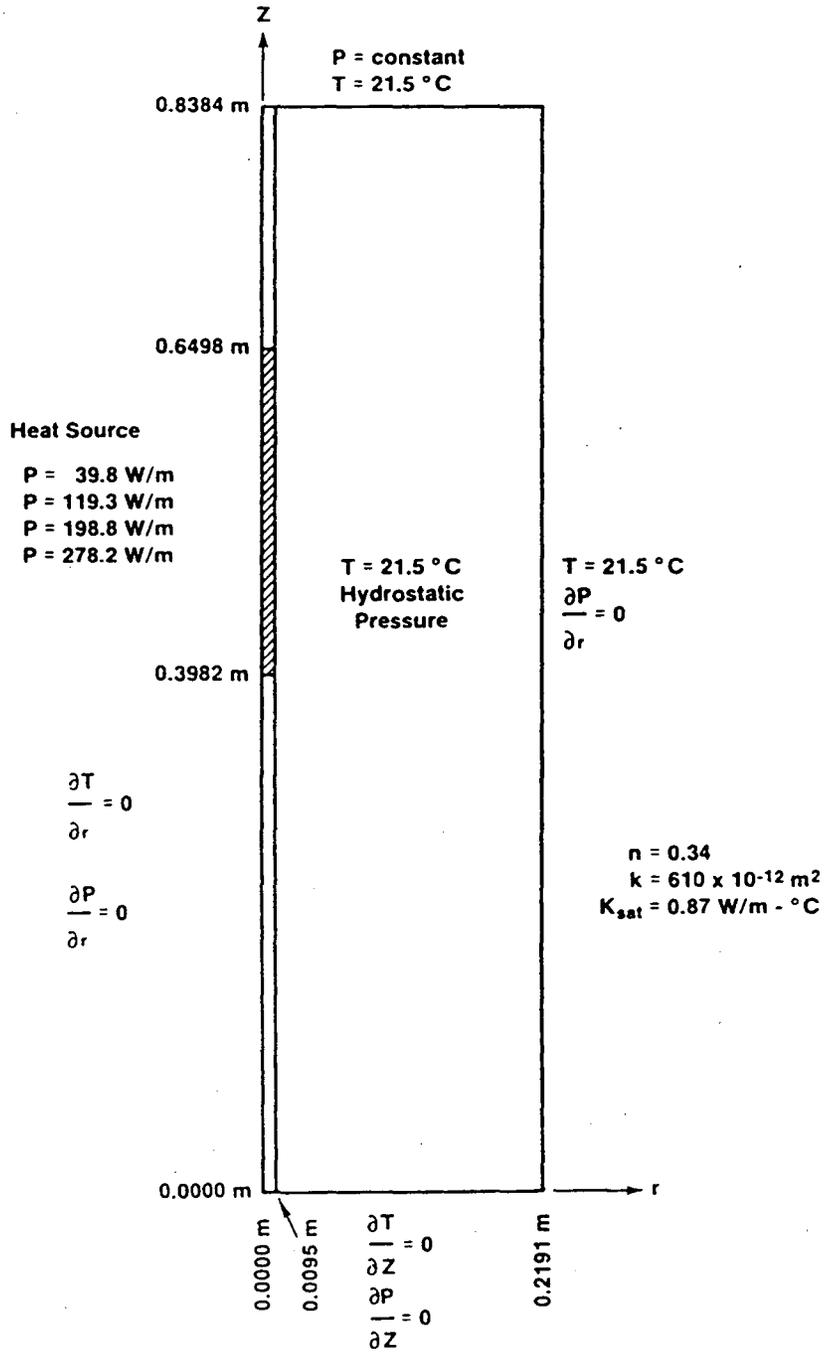


Figure 6. Schematic of the convection cell apparatus in Test Problem 3 (from Reda [1984]).

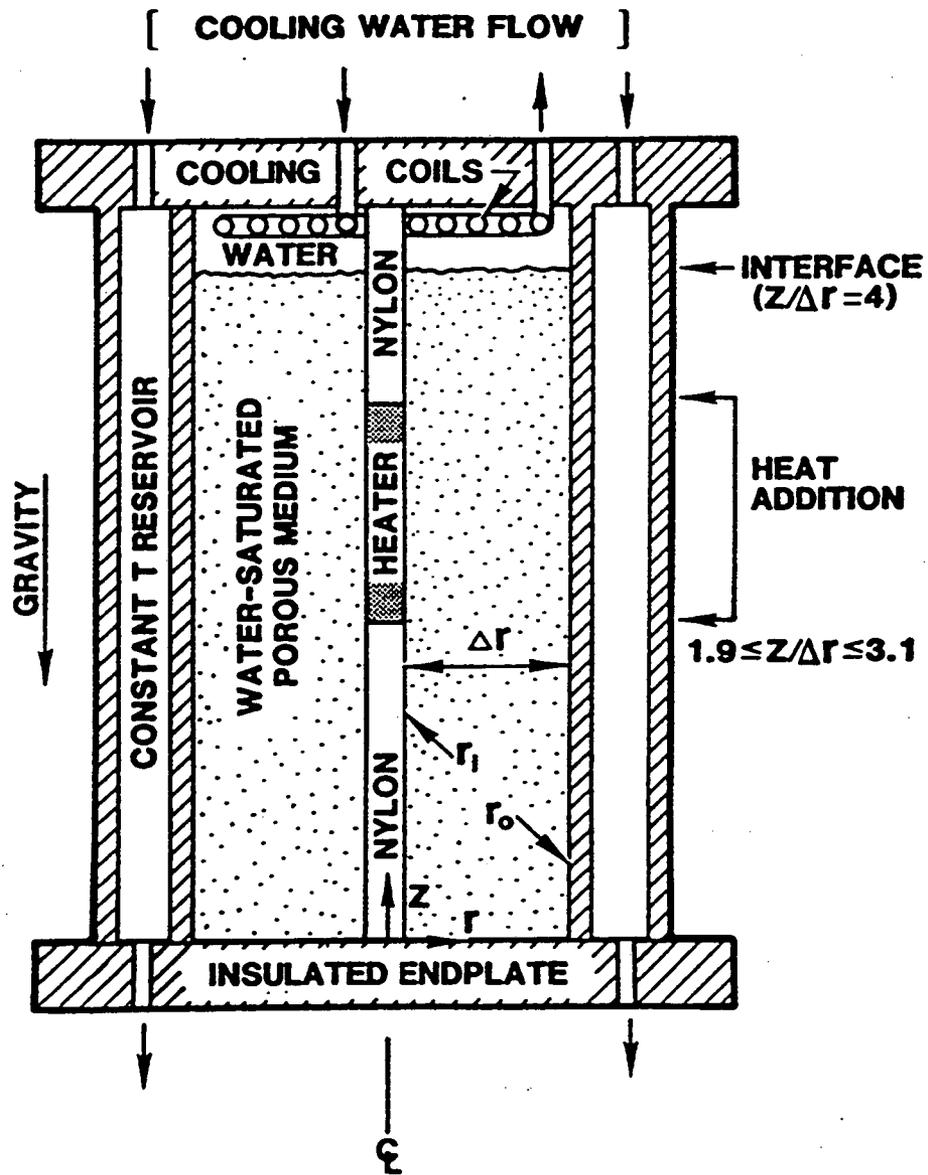


Figure 7. The convection cell experiment in Test Problem 3.

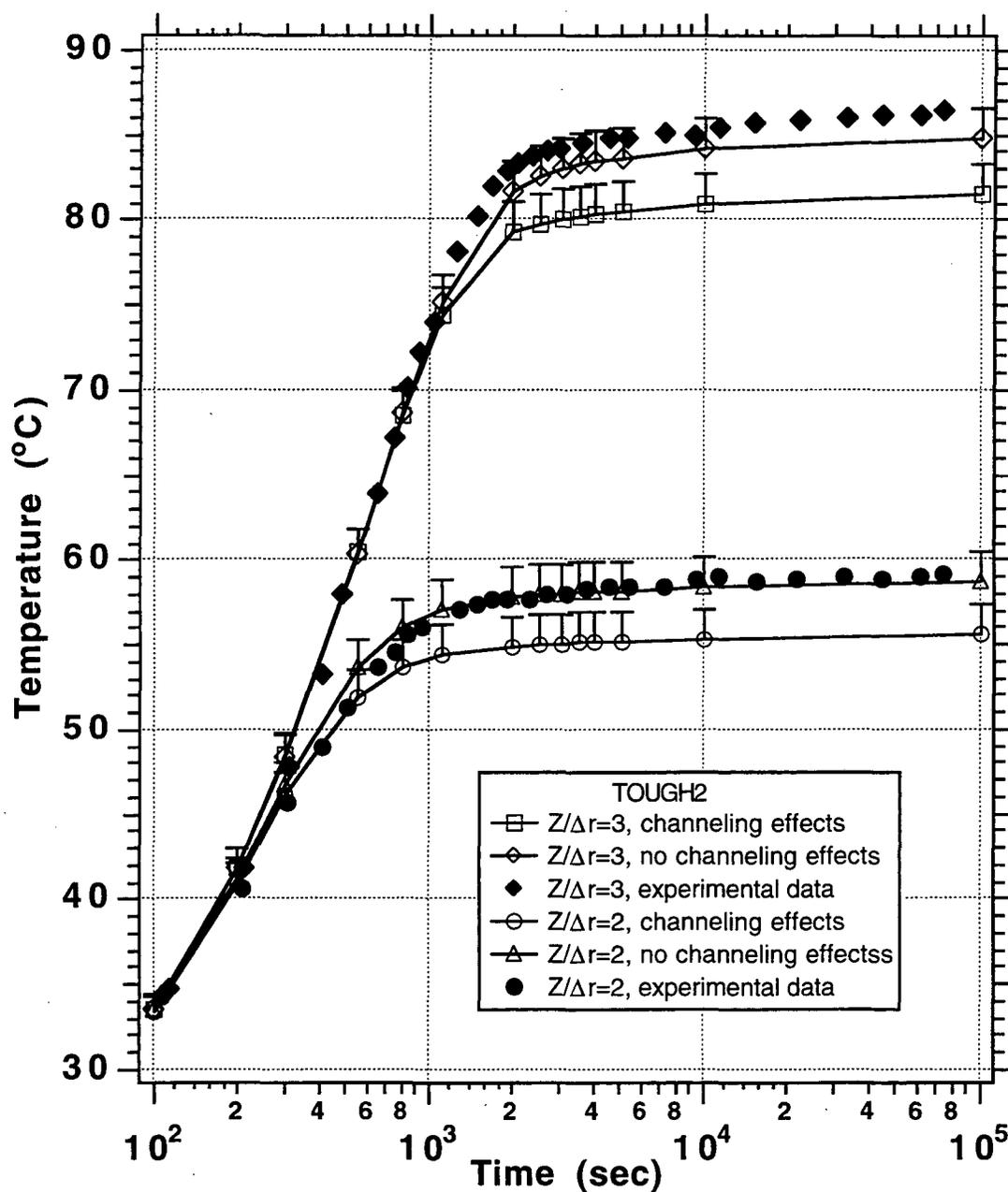


Figure 8. Comparison of TOUGH2 predictions with experimental data at the top and bottom ($z/\Delta r = 3$ and 2 respectively) of the heating element in Test Problem 3. The bars attached to the TOUGH2 curves indicate the predicted temperature on the surface of the heater.

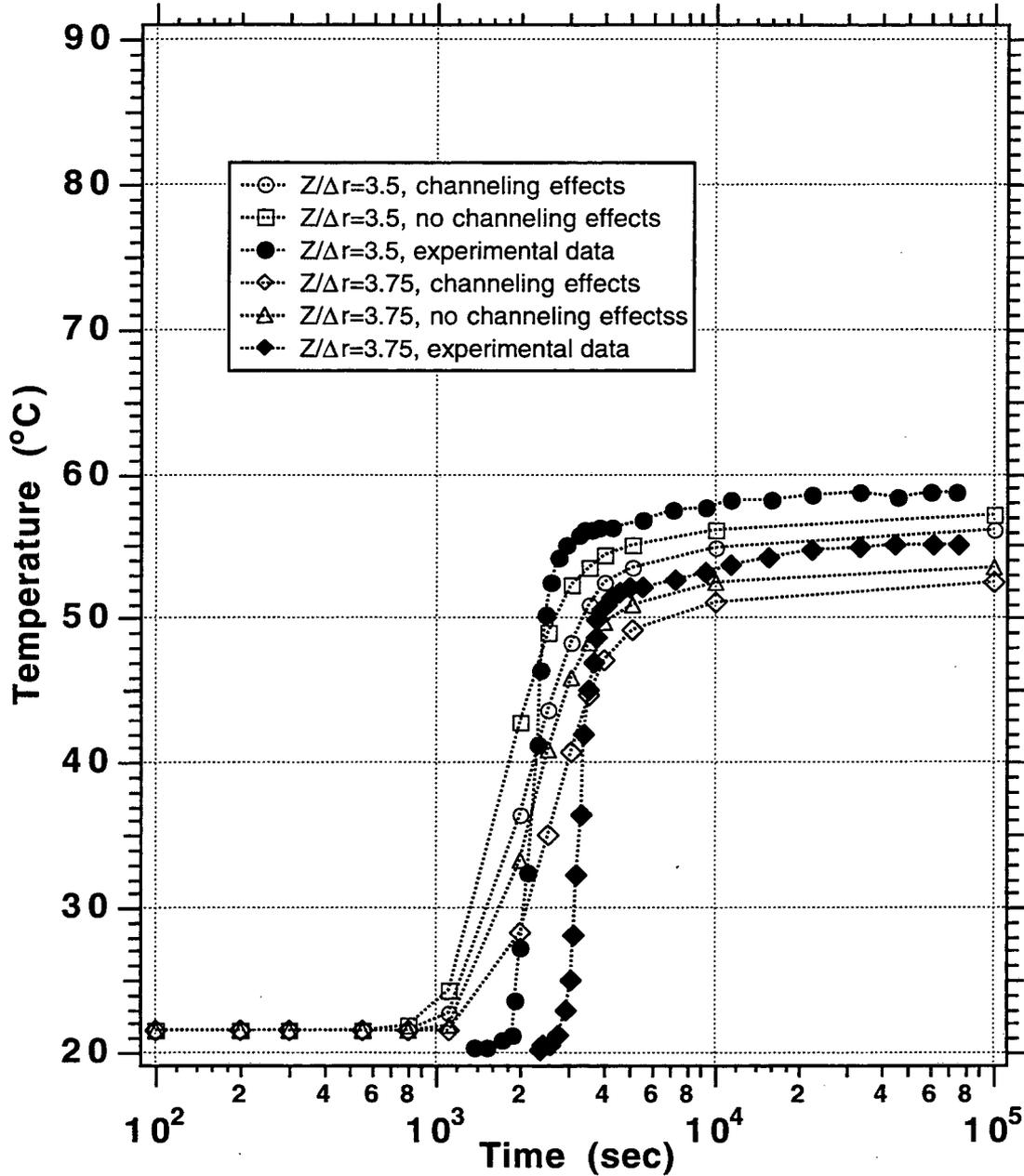


Figure 9. Comparison of TOUGH2 predictions with experimental data at $z/\Delta r = 3.5$ and $z/\Delta r = 3.75$.

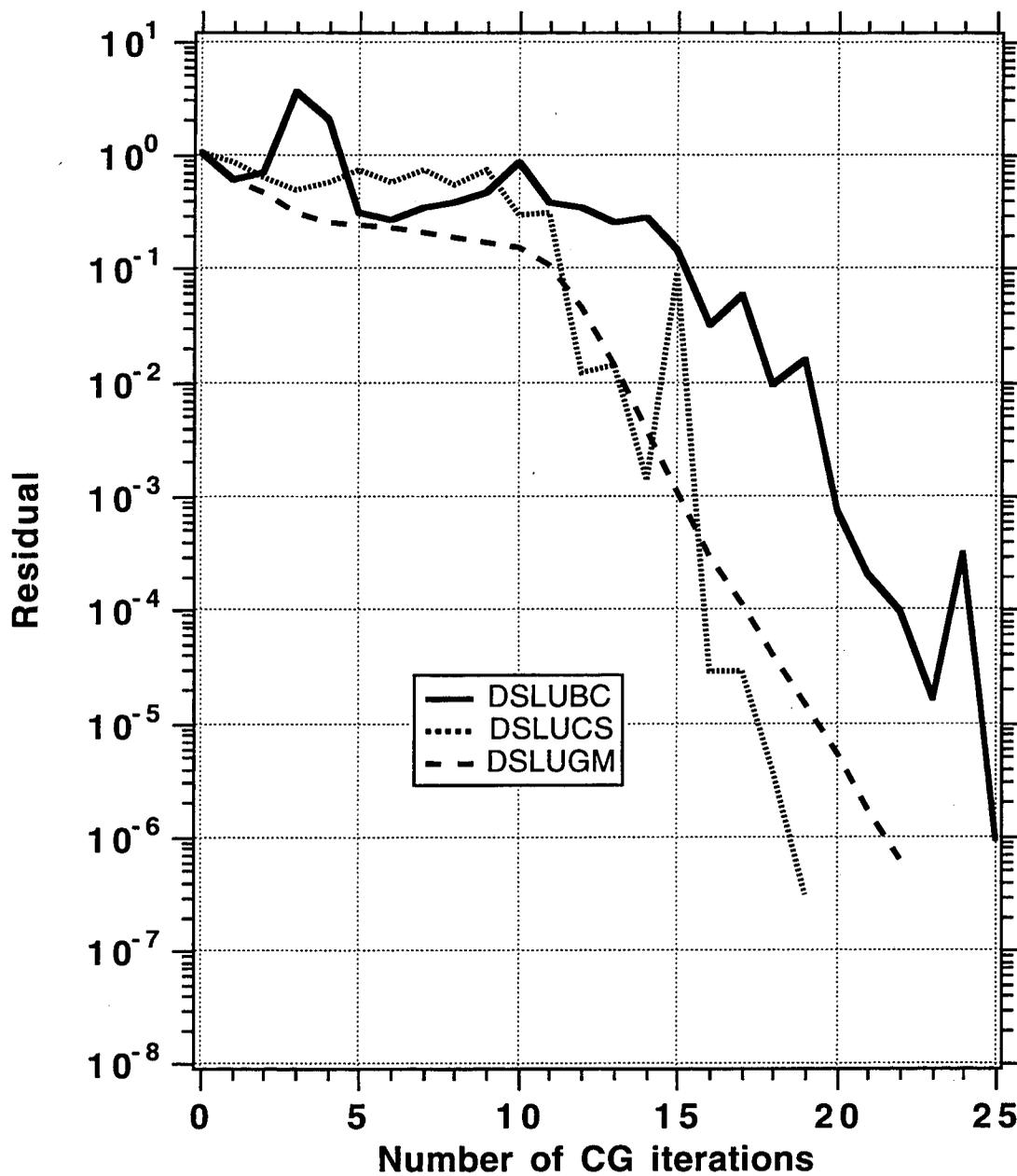


Figure 10. CG solver performance in the first Newtonian iteration of the first timestep in Test Problem 3 when flow channeling is neglected.

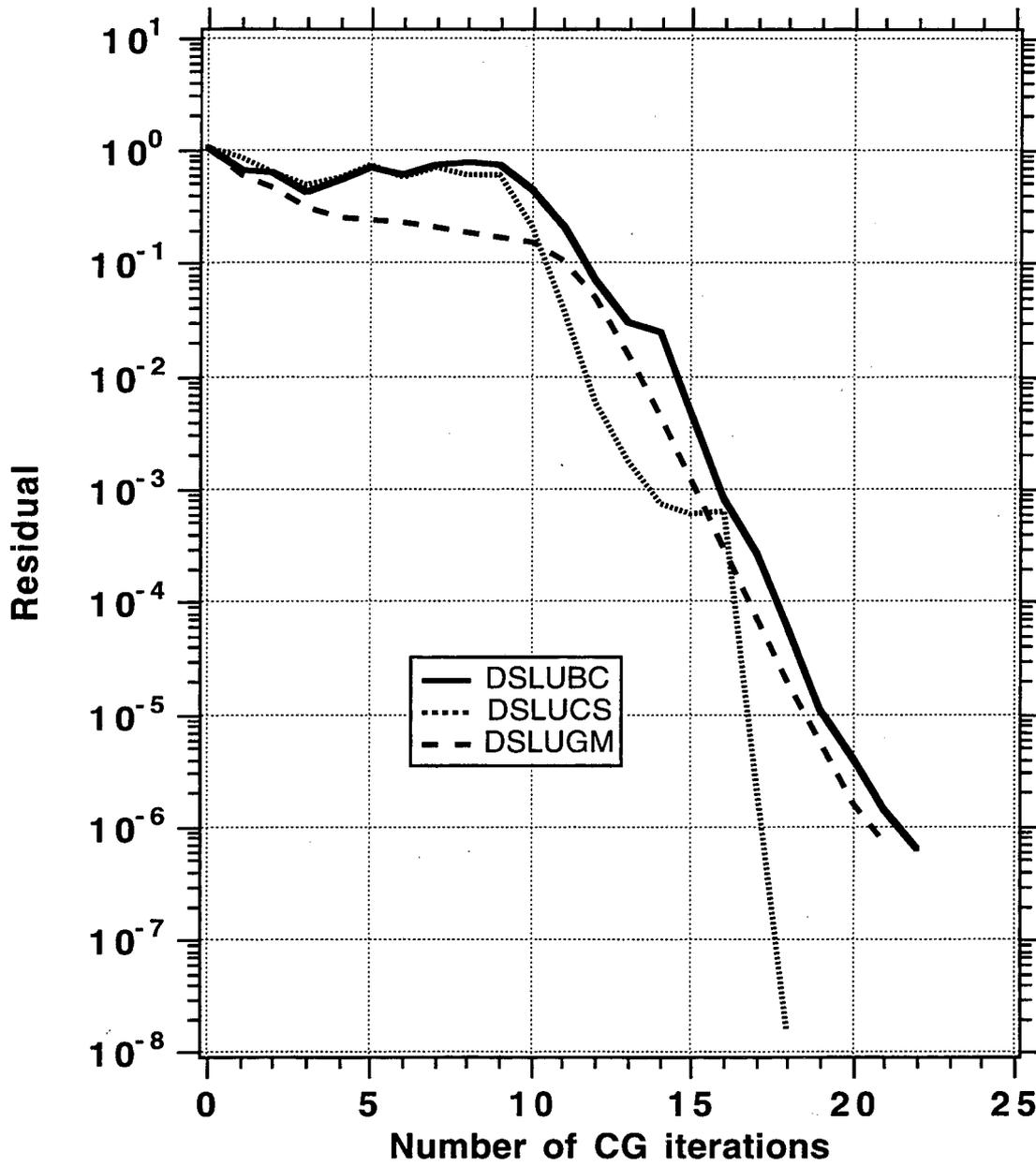


Figure 11. CG solver performance in the first Newtonian iteration of the first timestep in Test Problem 3 when permeability enhancement is considered.

4.4. Test Problem 4: Determination of Effective Permeability in a Homogeneous Medium with Embedded Impermeable Objects (2-D, Single-Phase, Isothermal and Non-Isothermal)

4.4.1. Problem description

This problem examines flow in a simple two-dimensional model of a heterogeneous porous medium. The basic computational grid is generated with the MESHMAKER facility of TOUGH2 as a 20 m by 15 m x-y section, with grid spacing of $\Delta x = 0.25$ m, $\Delta y = 0.125$ m, for a total of $80 \times 120 = 9600$ grid blocks (Figure 12). The y-axis is rotated 90° against the horizontal to make the section vertical. A mesh preprocessing program is then used to place impermeable obstacles with lengths uniformly distributed in the range of 2 - 4 m (Figure 13). For simplicity, rather than assigning the grid blocks belonging to the impermeable regions (shown black in Figure 13) a permeability of zero, these blocks are simply removed from the mesh, leaving a total of 8003 grid blocks. Problem parameters are chosen representative of typical alluvial soils and are given in Table 4. The input data file for this problem is found under the name "PROB4" on the accompanying diskette.

The heterogeneous medium described above has been used to study the behavior of liquid infiltration plumes [Pruess, 1994]. In this problem we present results of single-phase flow simulations in the medium for determining effective permeability. The MESH file was edited to attach a grid block of 20 m length across the top of the domain, and a similar block at the bottom. The entire domain is initialized in single-phase conditions, at a pressure of $P = 1.0 \times 10^5$ Pa, and a temperature of 20 °C. The simulation is performed with the EOS1 fluid property module, using $NK = NEQ = 1$ in block MULTI to solve just one water mass balance equation per grid block. This results in a total of $N = 8003$ equations. The top block is made inactive, gravity is turned off, and injection of water at a rate of 1 kg/s is specified in the bottom block.

With 10 time steps the system achieves a very accurate steady state, with pressure in the bottom block settling in at 7.395×10^5 Pa.. From Darcy's law, flow rate \mathbf{F} is given by

$$\mathbf{F} = A k \frac{\rho}{\mu} \nabla P \quad (21)$$

Inserting applicable parameters of $\mathbf{F} = 1$ kg/sec, $A = 20$ m², $\rho = 998.5$ kg/m³, $\mu = 1.002 \times 10^{-3}$ Pa-s, $\nabla P = (7.395 \times 10^5 - 1.0 \times 10^5)/15 = 4.263 \times 10^5$ Pa/m, we find that the effective vertical permeability is 1.177×10^{-12} m².

The problem was also run in non-isothermal mode (specifying $NK = 1$ and $NEQ = 2$ in block MULTI). Of course, because injection is made isothermally temperatures do not change at all, but the Jacobian matrix is very different in non-isothermal mode: it is twice as big (i.e. it has $N = 16,006$ equations), is less sparse, contains large cross-derivatives between mass- and energy-balance equations, and is substantially more challenging to solve. As expected, the two solutions (obtained for $NEQ = 1$ and $NEQ = 2$) were identical.

4.4.2. Solver performance

MA28 could not solve this problem in either isothermal or non-isothermal mode because of insufficient memory. In isothermal mode all three CG routines produced a solution, but the differences in performance and results are significant. The clear superiority of the DSLUCS routine is evident. It required 86 sec and no more than 173 CG

Table 4. Parameters for Test Problem 4	
Permeability Porosity	$k = 10^{-11} \text{ m}^2$ $\phi = 0.35$
Relative Permeability	
<i>van Genuchten</i> function [1980] $k_{rl} = \sqrt{S^*} \left\{ 1 - \left(1 - [S^*]^{1/\lambda} \right)^\lambda \right\}^2$ irreducible water saturation exponent	$S^* = (S_1 - S_{lr}) / (1 - S_{lr})$ $S_{lr} = 0.15$ $\lambda = 0.457$
Capillary Pressure	
<i>van Genuchten</i> function [1980] $P_{cap} = -(\rho_w g/a) \left([S^*]^{-1/\lambda} - 1 \right)^{1-\lambda}$ irreducible water saturation exponent strength coefficient	$S^* = (S_1 - S_{lr}) / (1 - S_{lr})$ $S_{lr} = 0.0 \text{ or } 0.15$ $\lambda = 0.457$ $a = 5 \text{ m}^{-1}$
Geometry of Flow Domain	
2-D vertical (x,z) section width (x) depth (z) gridding	20 m 15 m 80 x 120 = 9600 blocks $\Delta x = 0.25 \text{ m}$ $\Delta z = 0.125 \text{ m}$
heterogeneity: stochastic distribution of impermeable obstacles	

iterations to solve the problem, i.e. the maximum N_{CG} was less than 2.2% of the number of equations N , thus demonstrating the efficiency of this routine in this problem.

The DSLUBC solution, which required 158 sec, took 1.84 times longer than DSLUCS and was very close to the theoretically predicted ratio of 2 between the time requirements of the two methods. Though significantly slower than DSLUCS, DSLUBC provided an efficient solution which never required more than $N_{CG} = 225$ iterations, corresponding to $N_{CG}/N = 2.81\%$.

In contrast, convergence of the generalized minimum residual method (routine DSLUGM) is very slow, and in most cases the specified convergence tolerance of 10^{-6} cannot be achieved within the maximum number (810) of iterations. Nonetheless, the approximate results obtained from the generalized minimum residual method are "good enough" to permit the Newton-Raphson process to converge, albeit with a few more (outer) iterations, i.e. the inaccuracies in the CG solutions increase the needed number of Newtonian iterations.

The performance of the CG solvers is illustrated in Figure 14, which demonstrates the steady but extremely slow convergence of the DSLUGM solution in the first Newtonian iteration of the 10th timestep. The DSLUBC and DSLUCS solvers do converge, but demonstrate violent oscillations caused by the loss of orthogonality as the steady state is approached. This irregular convergence behavior afflicts most CG solvers in situations where the iterations are started close to the solution (e.g when approaching steady state), and may lead to severe numerical cancellation and errors in the solution.

In non-isothermal mode the performance of the solvers is very similar. The superiority and efficiency of the DSLUCS routine in the solution of this problem is clear. DSLUCS requires 971 CPU sec and a maximum $N_{CG} = 361$ iterations for the solution of $N = 16,006$ equations, i.e. $N_{CG}/N = 2.26\%$. It is 2.6 times faster than DSLUBC, which also produces the correct solution.

DSLUGM stagnates in the solution of the non-isothermal problem, and is incapable of producing a solution. The algorithm does not break down, but the residuals do not decrease after the first few iterations. The stagnation is due to the fact that the matrix is indefinite non-symmetric.

Table 5. Solver Performance in Test Problem 4					
Number of Equations: 8,003			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	Could not solve the problem due to insufficient memory.				
DSLUBC	10	17	225	172	158
DSLUCS	10	17	173	154	86
DSLUGM	10	19	316	810	379
Number of Equations: 16,006			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	Could not solve the problem due to insufficient memory.				
DSLUBC	10	38	1,602(†)	172	2,525
DSLUCS	10	37	361	164	971
DSLUGM	Stagnates				

(†): The maximum number of allowable iterations (1602) was reached only once.

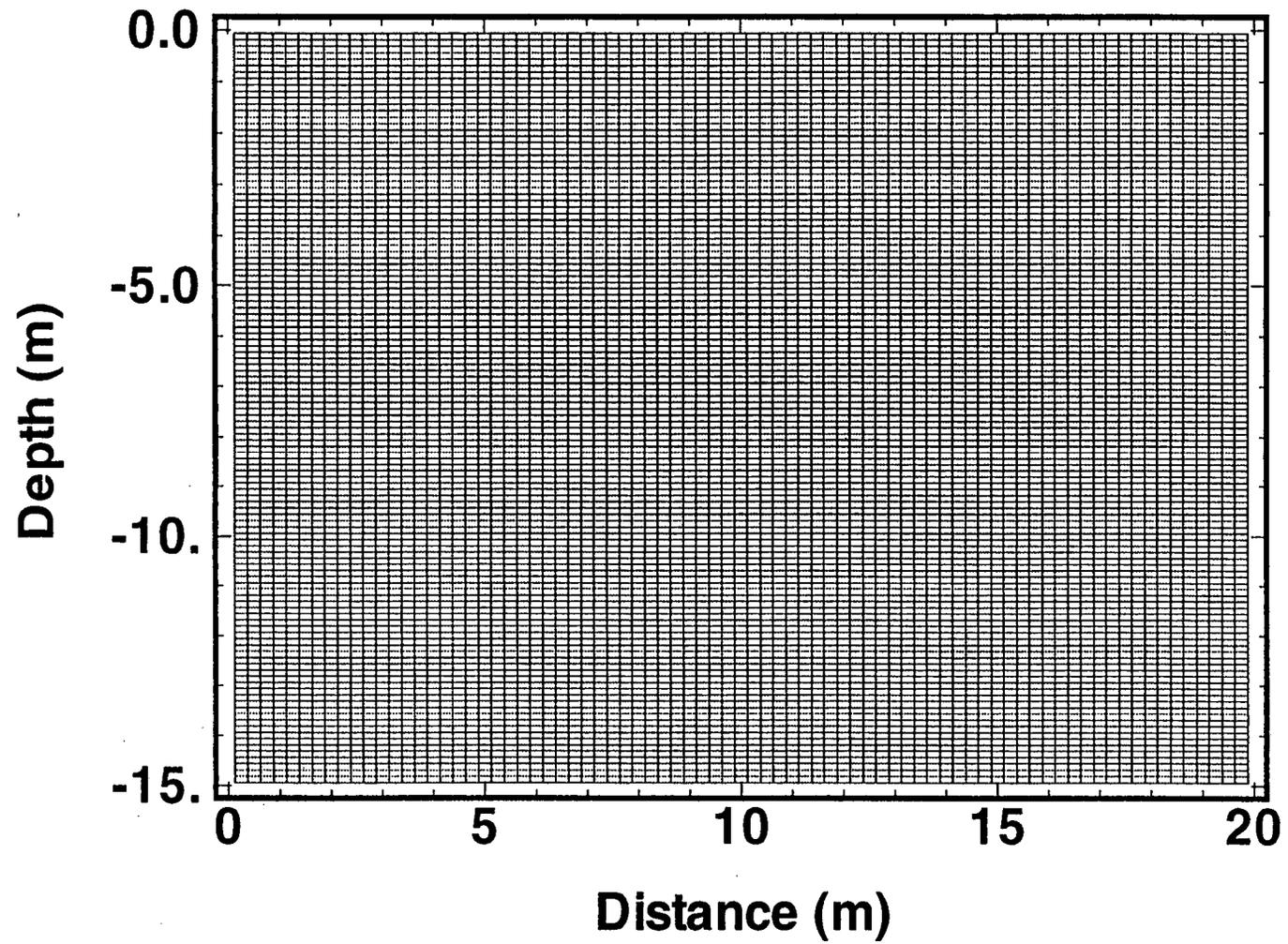


Figure 12. The computational grid in Test Problem 4.

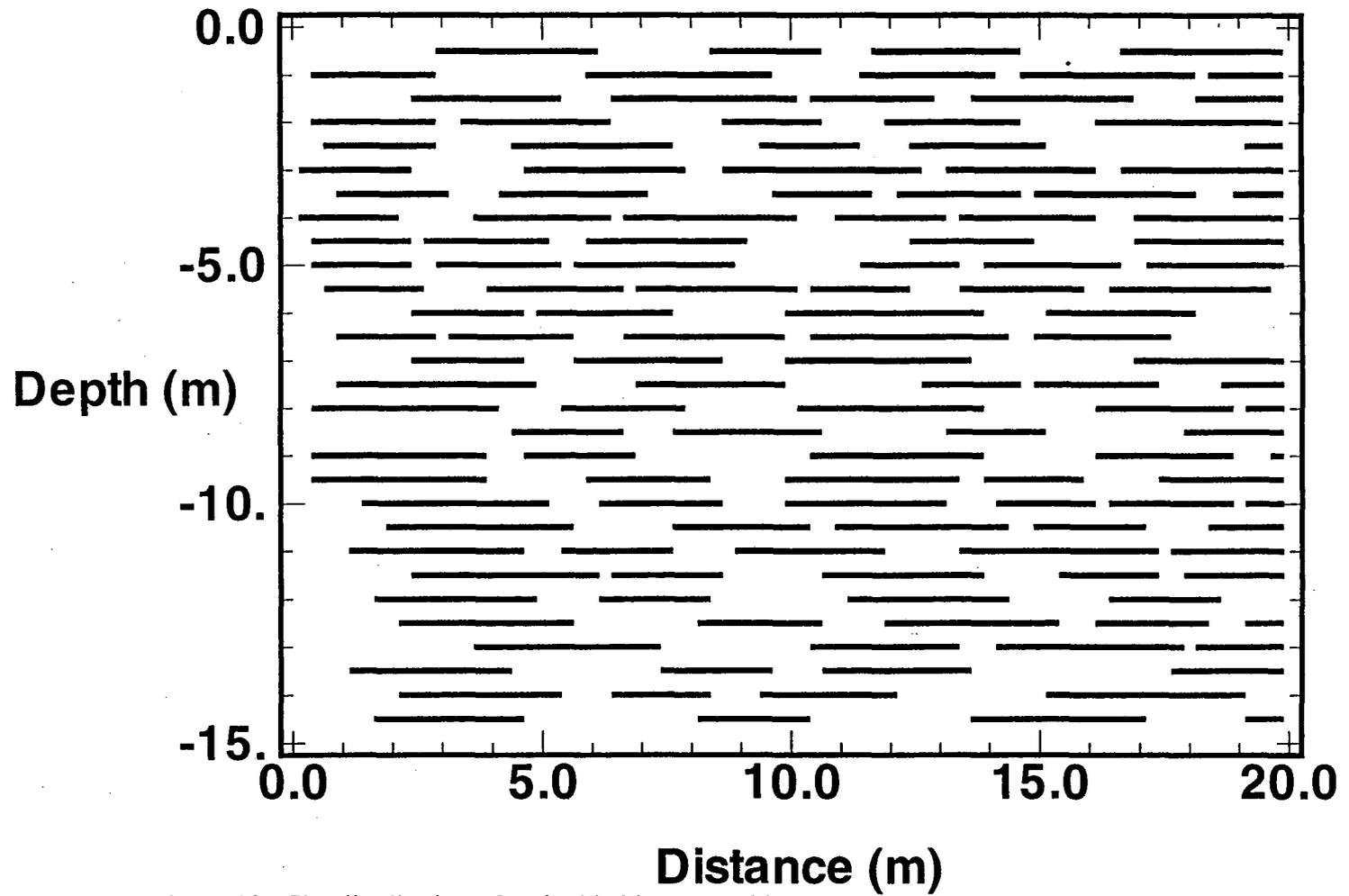


Figure 13. The distribution of embedded impermeable objects in Test Problem 4.

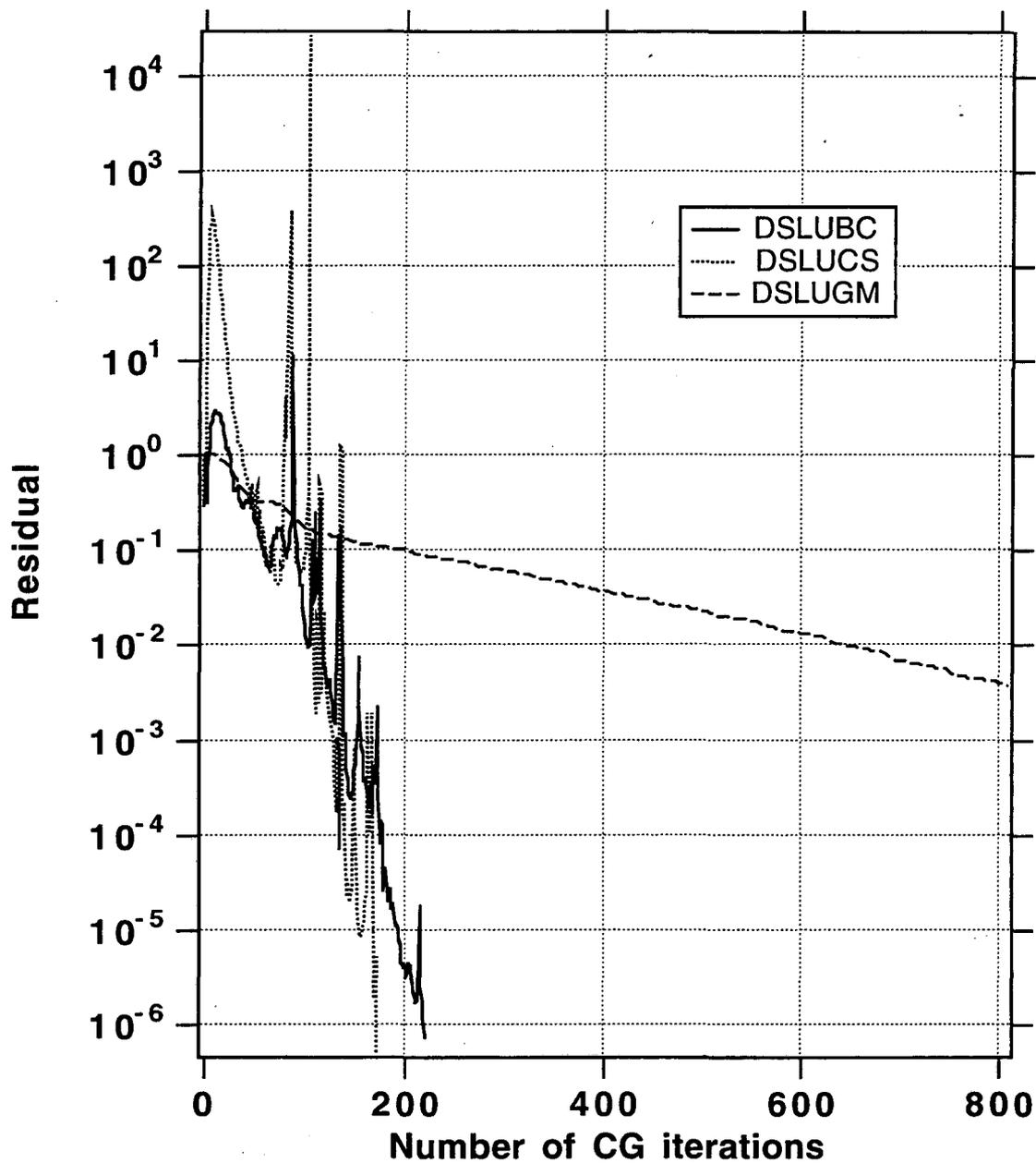


Figure 14. Performance of the CG routines in the first Newtonian iteration of the 10th timestep in Test Problem 4.

4.5. Test Problem 5: Determination of Effective Permeability in a Medium with Strong Spatially Correlated Heterogeneity (2-D, Pseudosingle-Phase, Isothermal)

4.5.1. Problem description

In this problem we examine flow in a vertical section grid with strong heterogeneity. The basic computational grid is generated with the MESHMAKER facility of TOUGH2 as a 20 m by 20 m x-y section, with grid spacing of $\Delta x = 0.2$ m, $\Delta y = 0.2$ m, for a total of $100 \times 100 = 10,000$ grid blocks. A spatially-correlated random field (Figure 14) was generated with the turning bands method [Mantoglou and Wilson, 1982], using a computer program written by A. Tompson [Tompson, 1989; Antunez and Pruess, 1994]. The data in Figure 15 are used as “permeability multipliers” (PM) in a TOUGH2 simulation, where in subroutine MULTI all permeabilities are scaled according to

$$k \rightarrow \tilde{k} = f_{PM} \cdot k, \quad (22)$$

where f_{PM} is the permeability multiplier. Capillary pressures are scaled consistently as [Leverett, 1941; Scheidegger, 1974]

$$P_{cap} \rightarrow \tilde{P}_{cap} = P_{cap} / \sqrt{f_{PM}} \quad (23)$$

As can be seen in Figure 15, $0.1 \leq f_{PM} \leq 100$, i.e. the scaling covers three orders of magnitude. This scaling results in a highly heterogeneous system where adjacent gridblocks have significantly different permeabilities (differing sometimes by over two orders of magnitude) and each its own capillary function. For permeability, porosity, and characteristic curves we again use the data given in Table 4. As in Test Problem 4, we attach top and bottom grid blocks across the domain, make the top block inactive and inject at a constant rate in the bottom block, to “measure” single-phase effective permeability.

A fluid property module “EOS9” for solving saturated/unsaturated flow by Richards’ equation is employed for the simulation. This module is not a part of the generally available “regular” TOUGH2 code, but is a research routine developed for the investigation of flow and transport in very heterogeneous media. The input data set for this problem is not included in the diskette accompanying this report because the EOS9 routine needed for execution is not generally available.

The simulation was run for a total of 10 timesteps. When EOS9 is used, a single equation per gridblock is solved, resulting in a total of $N = 10,000$ equations. Using equation (21) and inserting applicable parameters of $F = 1$ kg/sec, $A = 20$ m², $\rho = 998.5$ kg/m³, $\mu = 1.002 \times 10^{-3}$ Pa-s, $\nabla P = (1.6401 \times 10^5 - 1.01 \times 10^5)/20 = 3.1505 \times 10^3$ Pa/m, we find that the effective permeability is 1.589×10^{-11} m².

4.5.2. Solver performance

The solver performance is summarized in Table 6. The very large memory requirements of the MA28 routine essentially preclude the use of the direct solver in this problem, making the CG routines the only option for a problem of this size. The DSLUBC routine performs very well, and solves the problem in a maximum of $N_{CG} = 216$ iterations. This corresponds to $N_{CG}/N = 2.16\%$, indicating a very efficient solution.

The DSLUCS routine fails completely, diverges, and is unable to generate a solution. The DSLUGM routine generally fails to converge, but gives results that are “good enough” to permit the Newton-Raphson process to proceed to convergence. This

process (not uncommon in the application of CG solvers) in essence transfers error from the CG to the Newtonian iteration component of the solution, and does not pose any problems unless the error is significant. DSLUGM requires significantly more iterations and 394 CPU sec to convergence on an IBM RS/6000 370 workstation, i.e. 2.6 times longer than the DSLUBC solution.

Figure 16 shows the evolution of the residuals (see subsection 3.3) of DSLUBC and DSLUGM with the number of CG iterations in the first Newtonian iteration of the first timestep. Figure 17 presents the same information for the DSLUCS routine. The superiority of the DSLUBC solver and the unsuitability of the DSLUCS routine for the solution of this problem is evident from these figures.

Table 6. Solver Performance in Test Problem 5					
Number of Equations: 10,000			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	Could not solve the problem due to insufficient memory.				
DSLUBC	10	16	216	209	154
DSLUCS	Diverges				
DSLUGM(*)	10	16	1,020	739	394

(*): The CG convergence criterion was not satisfied, but the solution was "close".

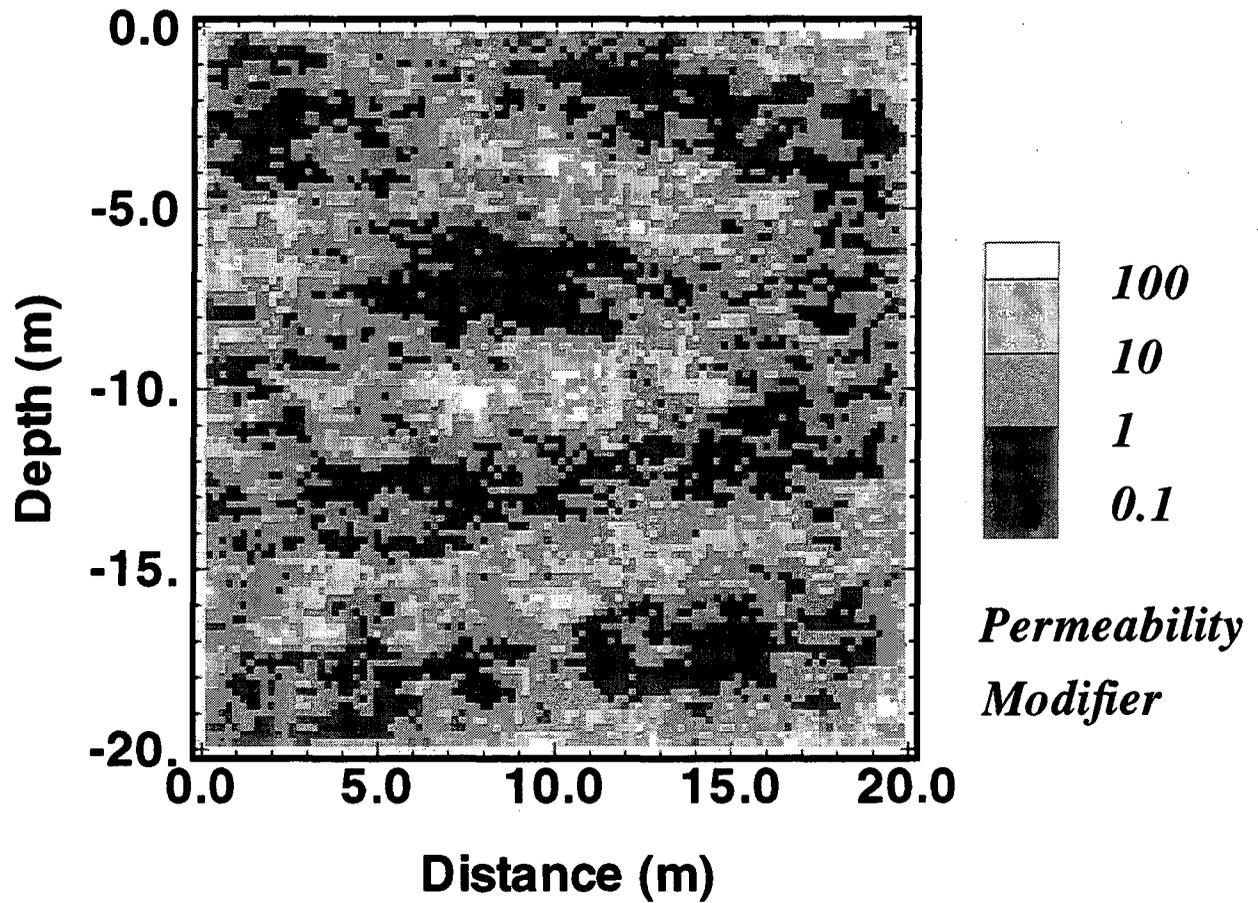


Figure 15. Distribution of the permeability modifier in Test Problem 5.

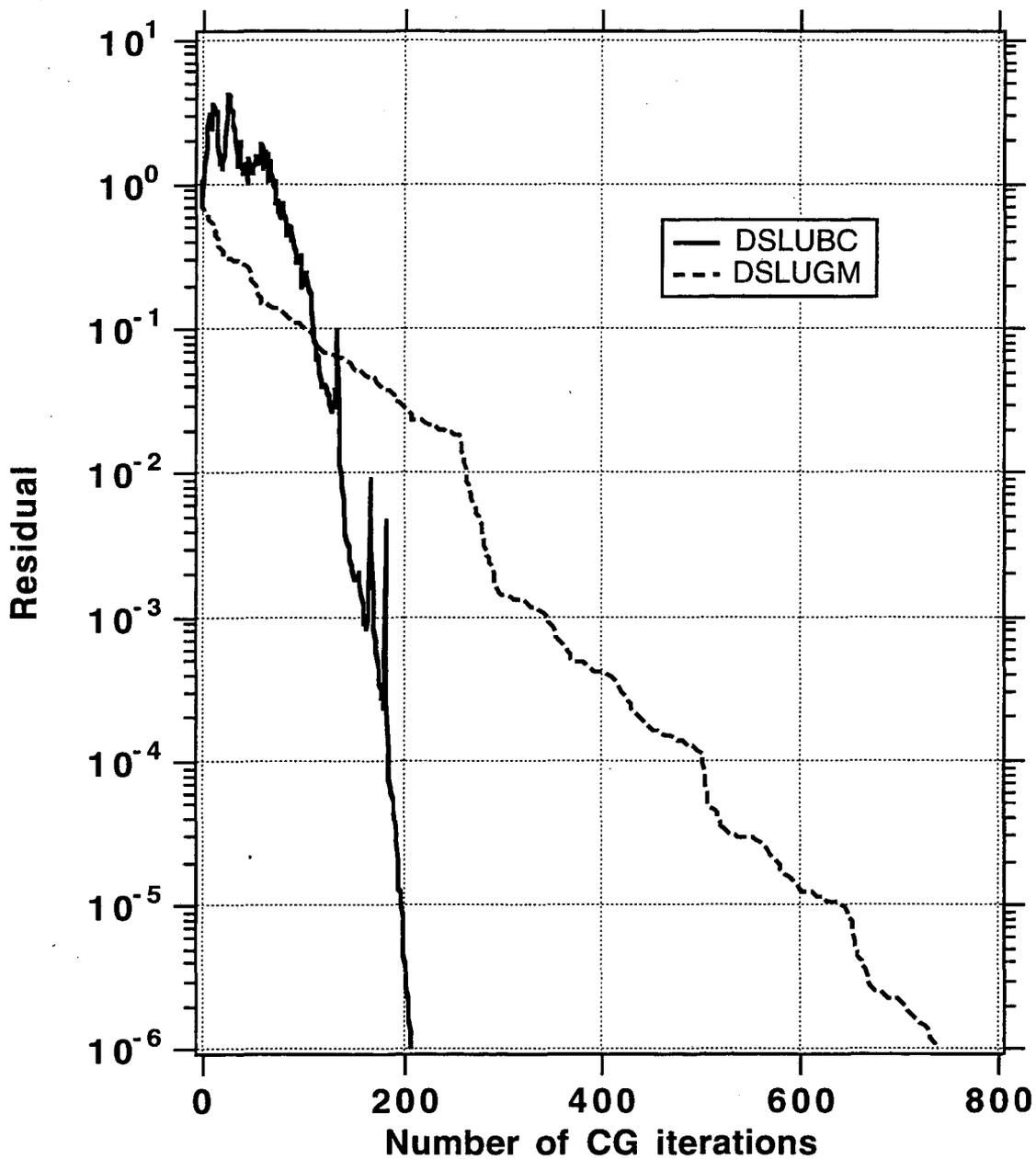


Figure 16. Performance of the DSLUBC and DSLUGM routines in the first Newtonian iteration of the first timestep in Test Problem 5.

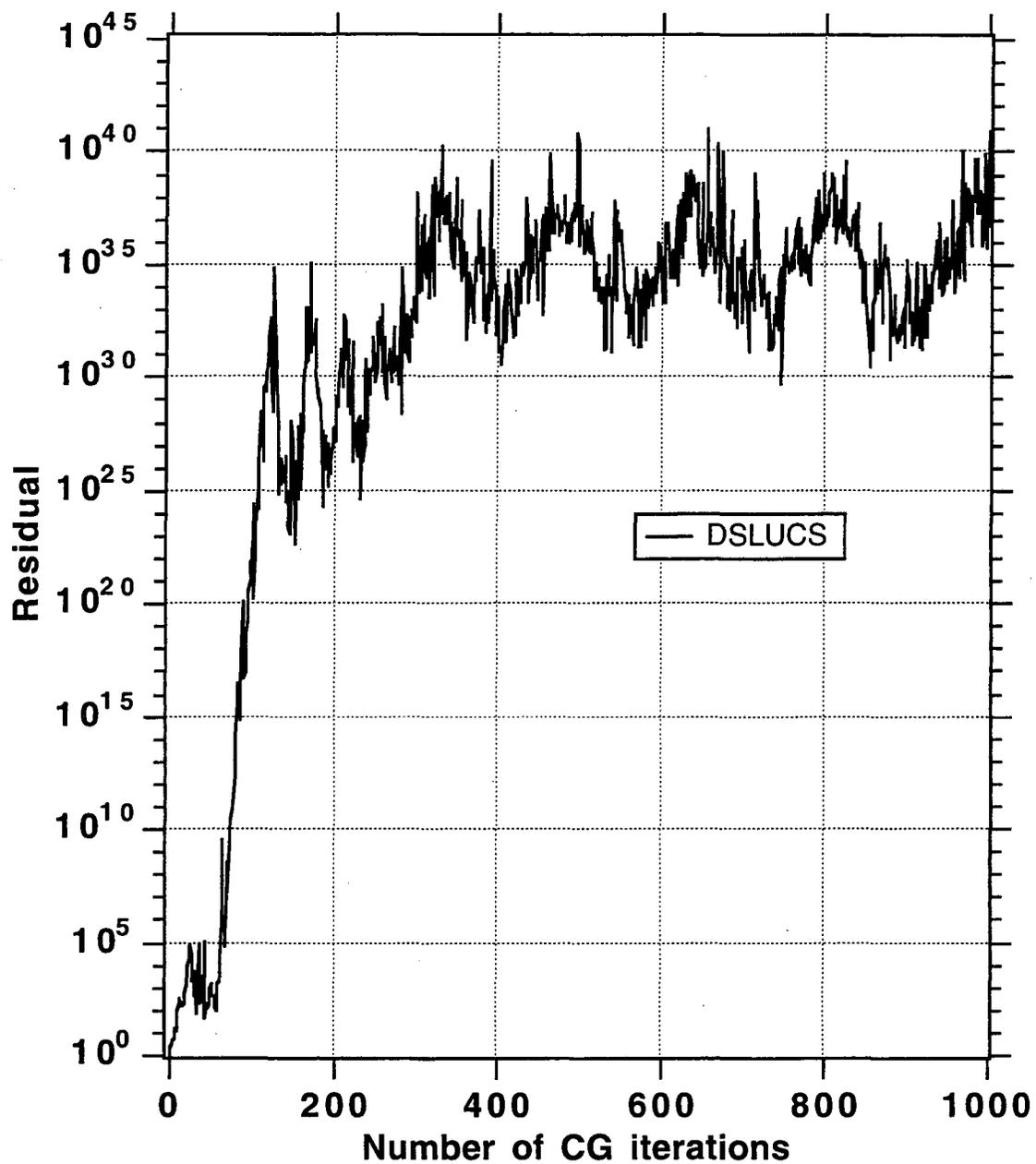


Figure 17. Divergence of the DSLUCS solution in the first Newtonian iteration of the first timestep in Test Problem 5.

4.6. Test Problem 6: 2-D Infiltration Experiment (2-D, Two-Phase, Isothermal)

4.6.1. Problem description

Test Problem 6 describes a two-dimensional infiltration laboratory experiment conducted by *Vauclin et al.* [1979], who provided measurement data and a numerical solution. This problem was discussed in detail by *Moridis and Pruess* [1992] in their validation study of TOUGH.

Water infiltrates at a rate of $4.111 \times 10^{-5} \text{ m/sec}$ over a length of 0.5 m of the vertical slab of soil depicted in Figure 18 which, due to symmetry, represents only the right half of the problem. The water table stands at 0.65 m from the bottom. The bottom boundary and the left boundary (line of symmetry) are considered impermeable to flow. Below $z = 0.65 \text{ m}$ the right boundary is a constant pressure boundary with a water saturation of 1 at the bottom; above $z = 0.65 \text{ m}$ it is a seepage surface, i.e. a mixed type of boundary condition which sets the water flux equal to zero when the medium is unsaturated and has a head equal to the hydraulic head when the medium is saturated (see below).

We compare experimental observations to numerical simulation results of the volumetric moisture content distributions with depth z at distances of $x = 0.19$ and 1.09 m from the line of symmetry. These comparisons are made at $t = 1800, 3600, 5400, 7200, 10800, 14400, 18000, 21000,$ and 28800 sec after the initiation of infiltration.

For this simulation the domain was subdivided in 378 (21×18 in x, z) gridblocks. We used the MESHMAKER facility of TOUGH2 [*Pruess, 1991*] to generate the grid (Figure 19). A very large volume (10^{29} m^3) was assigned to the right-hand boundary gridblocks (column 21), thus ensuring constant boundary pressures, saturations, and temperatures during the duration of the simulation. The simulation is performed with the EOS3 fluid property module, using $NK = NEQ = 2$ (i.e. isothermal conditions) in block MULTI to solve just two equations per grid block. This results in a total of $N = 756$ equations. The grid data, as well as all other pertinent information, can be found in the file "PROB6" on the accompanying diskette. A more complete discussion of the data inputs is provided by *Moridis and Pruess* [1992].

An upstream weighting scheme for mobilities (i.e. $MOP(11) = 2$) was used in this simulation. A no-flow top boundary was used because previous simulations had indicated that the mass transfer through this boundary was insignificant. Water was injected directly into the four shaded gridblocks highlighted in Figure 19 by specifying appropriate sources.

Initial pressures and saturations were hand-calculated and assigned using (i) the hydrostatic pressure distribution below the watertable, and (ii) the capillary pressure vs. water content relationship in *Vauclin et al.* [1979]. Coding the capillary pressure function into TOUGH2 requires a minimum of effort. For the user's convenience, we include the FORTRAN code for the *Vauclin et al.* [1979] capillary pressure function in the subroutine "PCAP" on the accompanying diskette.

The initial equilibrium state could have been also generated with a TOUGH2 run. A seepage face was assigned to the right-hand boundary gridblocks above the water table by (a) specifying these boundary blocks as single-phase gas in pressure equilibrium with the rest of the system and (b) assigning them to a domain with zero capillary pressure. The maximum allowed time step size was $\Delta t = 900 \text{ sec}$.

Figures 20 and 21 show the TOUGH2 predictions and the experimental observations at the desired locations and times. The agreement between experimental data

and numerical simulation results is good, but measurable local deviations are observed. The reason for these deviations is traced to the heterogeneity of the soil slab, and was extensively documented and discussed by *Moridis and Pruess* [1992].

4.6.2. Solver performance

As can be seen in Table 7, all three CG routines in the T2CG1 package are faster than the MA28 direct solver in the solution of this problem. The CG solutions are very efficient and achieve convergence in very few iterations. The solvers require a maximum of $N_{CG} = 40, 27,$ and 30 iterations to convergence for DSLUBC, DSLUCS, and DSLUGM respectively.

In terms of CPU time, the DSLUGM routine is the fastest in this problem (236 sec) and 2.36 times faster than MA28. However, there is practically no difference between the performances of DSLUGM and DSLUCS (241 CPU sec). MA28 was 1.88 times slower than DSLUBC, the slowest of the three CG routines (296 CPU sec).

Table 7. Solver Performance in Test Problem 6					
Number of Equations: 756			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	88	445	-	-	556
DSLUBC	88	445	40	16	296
DSLUCS	88	445	27	9	241
DSLUGM	88	445	30	14	236

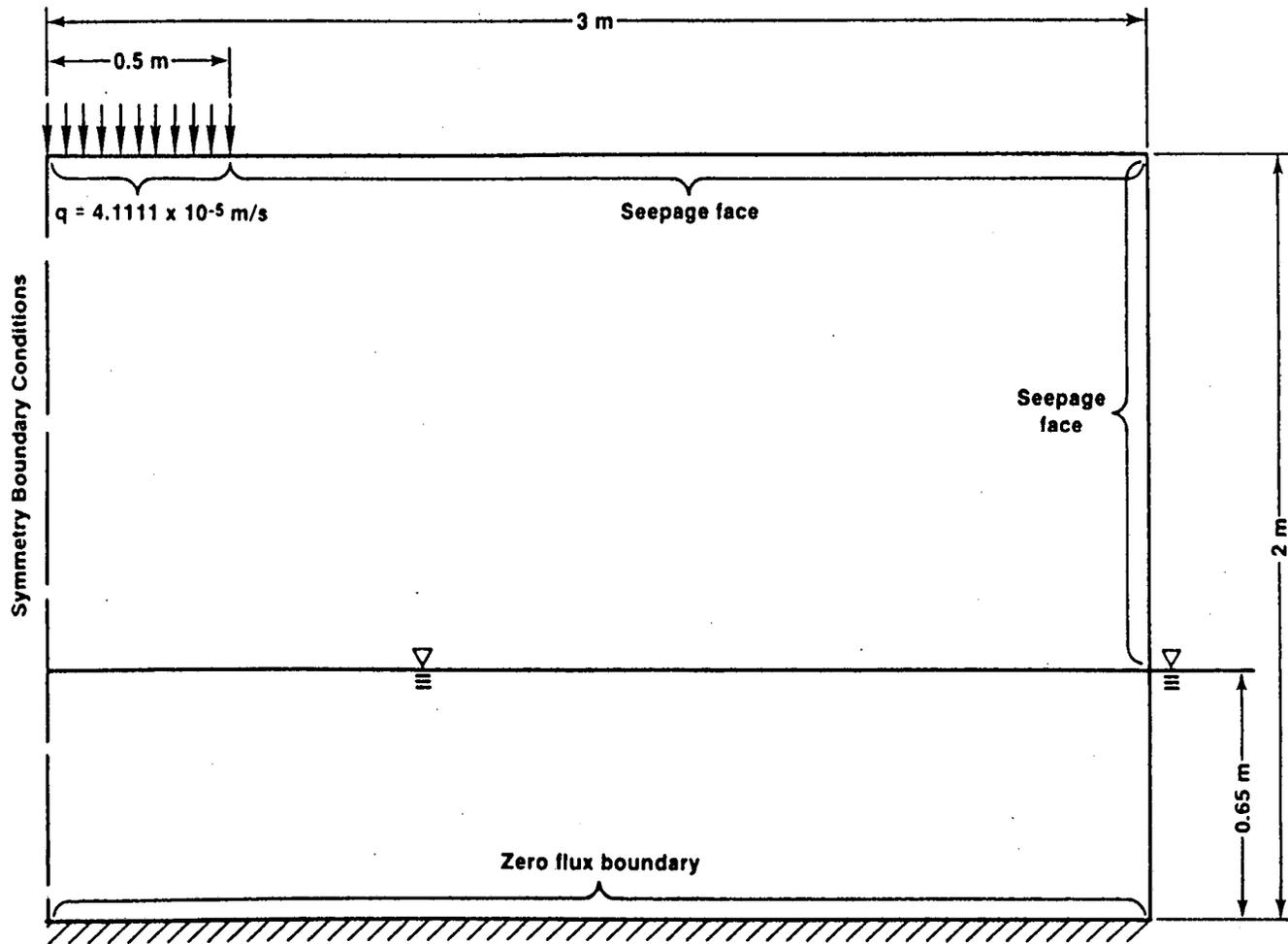


Figure 18. The two-dimensional infiltration system of *Vauclin's* [1977] experiment in Test Problem 6.

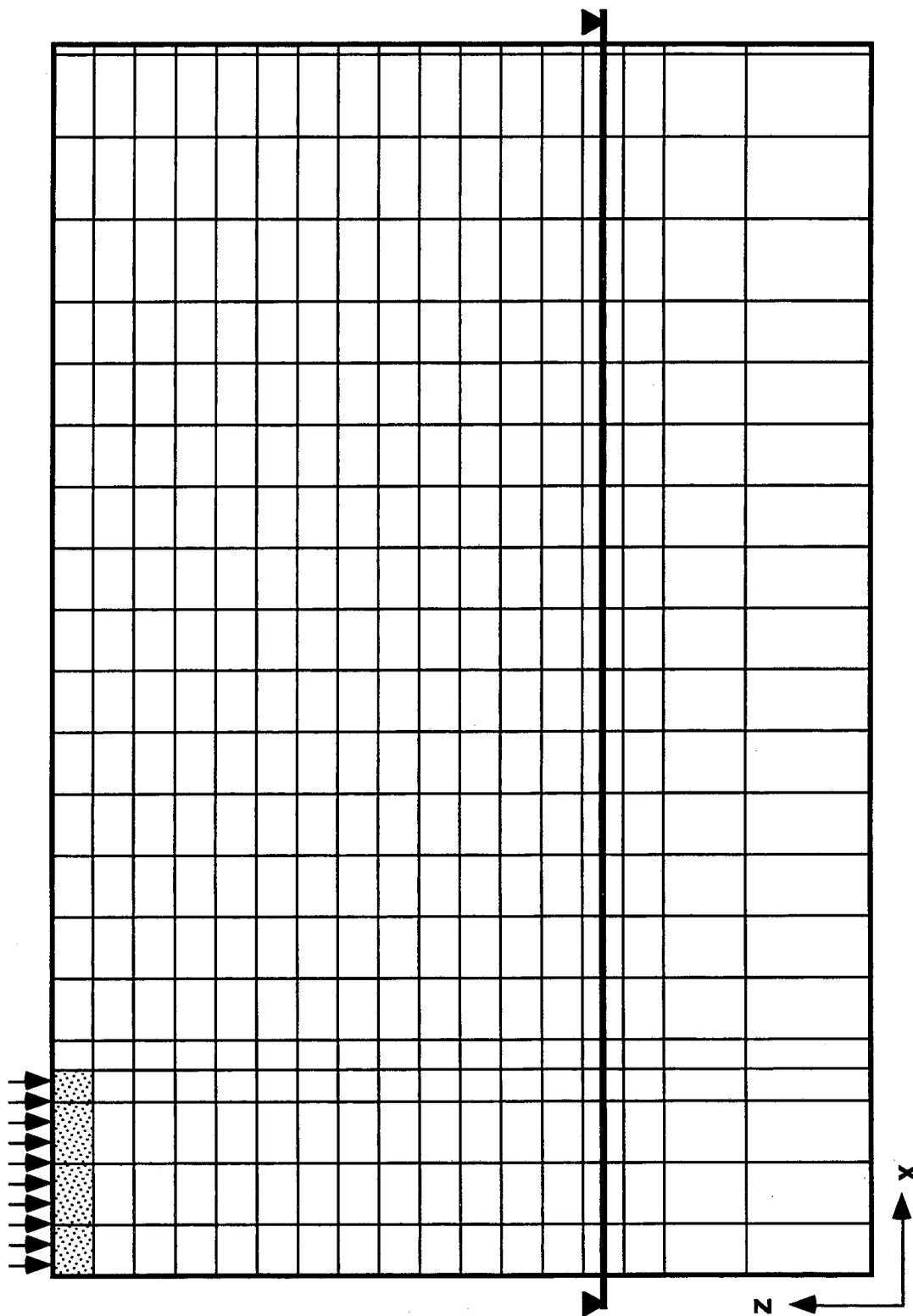


Figure 19. The TOUGH2 grid for the simulation of Test Problem 6.

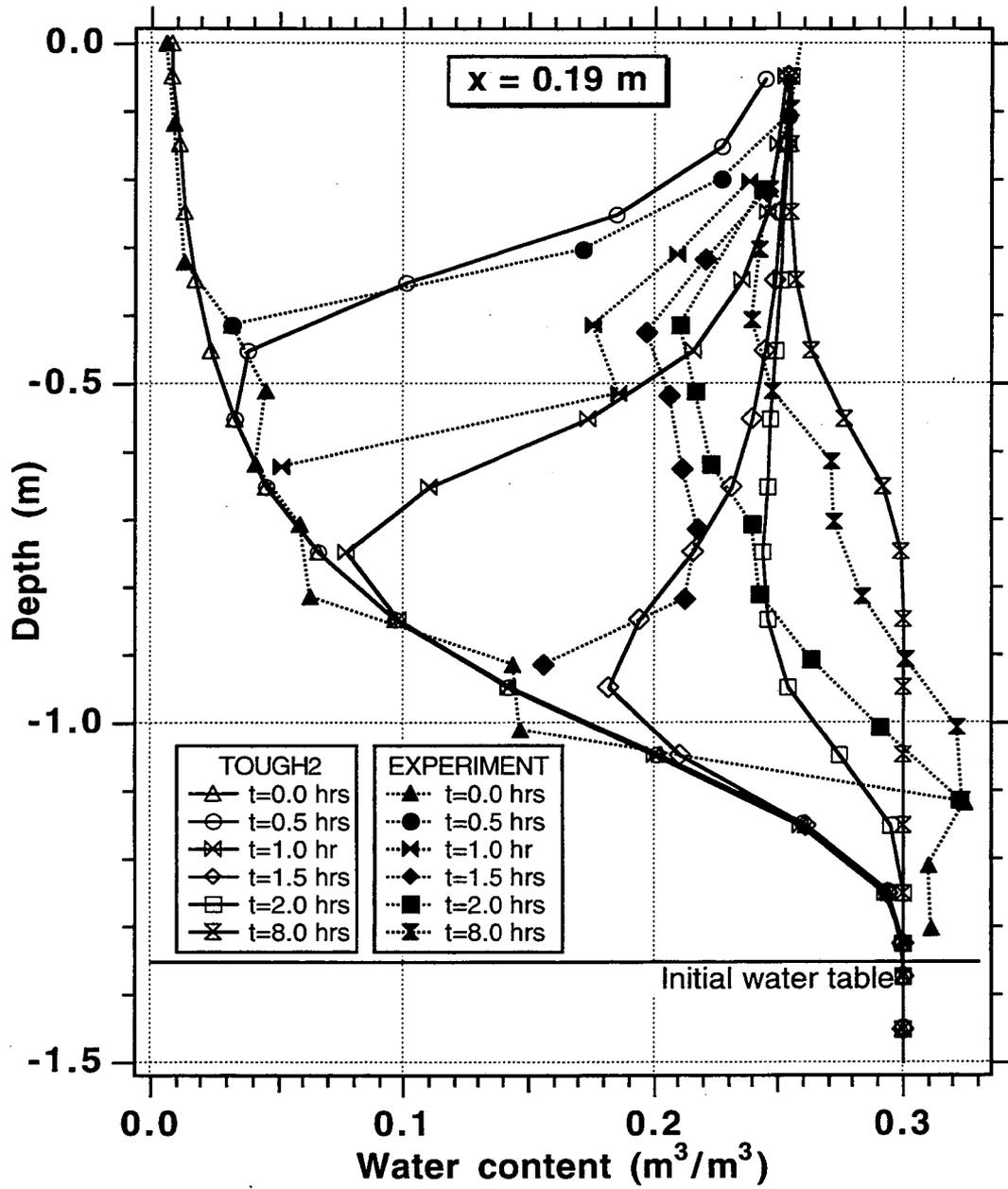


Figure 20. TOUGH2 predictions and experimental measurements at $x = 0.19 \text{ m}$ in Test Problem 6.

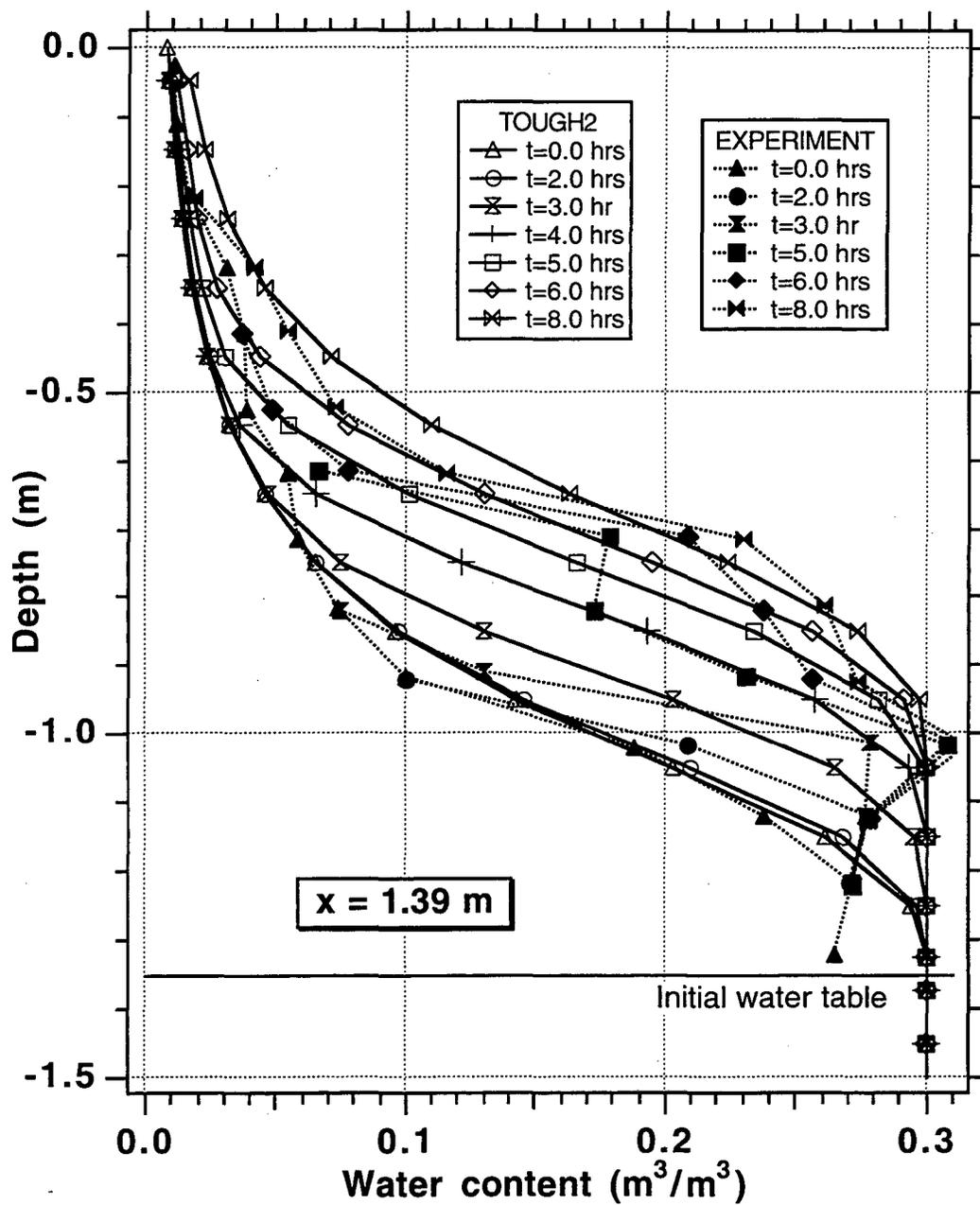


Figure 21. TOUGH2 predictions and experimental measurements at $x = 1.39 \text{ m}$ in Test Problem 6.

4.7. Test Problem 7: Vertical Section of WIPP (2-D, Two-Phase, Isothermal with High Permeability Contrasts and Phase Appearance and Disappearance)

4.7.1. Problem description

The WIPP (Waste Isolation Pilot Plant) is a U.S. Department of Energy (DOE) research and development facility for the underground disposal of transuranic waste from U.S. defense-related activities. The WIPP repository is located in a bedded salt formation near Carlsbad, New Mexico, and is 655 m underground within the Salado formation, which is brine saturated and consists of a large number of beds of relatively pure halite and impure halite containing interspersed clays and polyhalite. Thin interbeds of anhydrite, with associated underlying clay seams, are present in laterally extensive areas. Several of the more prominent anhydrite interbeds have been designated as Marker Beds. The repository horizon is separated by a few meters of halite from the overlying Marker Bed 138 and the underlying Marker Bed 139. A stratigraphic section on the Salado formation in the vicinity of the repository is shown in Figure 22. The layer permeabilities vary by four to five orders of magnitude.

The present simulation problem as designed by *Webb and Phelan* [1995] 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.

At the WIPP, waste that is emplaced in the repository will generate gas due to microbial degradation and anoxic corrosion. This gas generation may increase the pressure sufficiently to drive gas and brine into the surrounding Salado Formation, which is brine-saturated with a pore pressure of approximately 12 Mpa. The present simulation models the gas generation rate by specification of gas sources. At the end of the simulation (≈ 1000 years), gas generation has increased the room pressure sufficiently to drive gas out of the room and into the Salado Formation through the more permeable layers indicated in Figure 23. These results are based on a preliminary model which has evolved considerably since this problem was specified. The final version of this study is discussed briefly by *Webb et al.* [1995].

The simulated domain consists of 1200 elements in a two-dimensional vertical section grid. This is an isothermal two-phase flow problem, and is run with the EOS3 (water, air) module using $NK = NEQ = 2$ in block MULTI. This results in a total of $N = 2400$ equations. The file "PROB7" on the accompanying diskette contains the data inputs for this run. Permeabilities in the problem were stratified, generally very low (ranging between 10^{-23} and $10^{-15} m^2$), with extremely high permeability contrast in the vicinity of the more permeable repository. Porosities of the various strata were very low and all equal to $\phi = 0.01$. The initial pressures varied in the different strata. Capillary pressures and relative permeabilities followed the *Brooks and Corey* [1966] relationships. These are not available in the original TOUGH2 code, but coding them requires a minimum of effort. For the user's convenience, we include the FORTRAN code for the P_c and k_r relationships (in the subroutines "PCAP" and "RELP" respectively) in the accompanying disk. The added functions are clearly marked for easy identification.

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. Figure 23 shows the saturation distribution in the simulated domain at the end of the simulation period ($t = 3.16 \times 10^{10}$ sec).

4.7.2. Solver performance

We tested the performance of the various solvers on two different platforms (IBM RS/6000 370 and Apple Macintosh QUADRA 800) and different convergence criteria ($CLOSUR = 10^{-6}$ and 10^{-8} on the RS/6000, 10^{-8} and 10^{-10} on the Macintosh). The results are presented in Table 8. We notice that (a) significantly different performances are observed on the two computing platforms, (b) tightening the convergence criteria increases the execution times only marginally while (c) decreasing the number of timesteps and Newtonian iterations.

At the 10^{-6} convergence level, DSLUCS is the fastest solver on the RS/6000, requiring 408 sec to cover the desired simulation period. A maximum of 45 CG iterations (i.e. $N_{CG}/N < 2\%$) suffices for an accurate solution. It is interesting to note that the direct solver is the next fastest routine. This is possible because the size of the matrix is relatively small ($N = 2400$) and the problem is only two-dimensional. While DSLUGM is 5% slower than the direct solver and can still be considered competitive with MA28, DSLUBC is an unsuitable choice for this problem as it requires 31.7% more time than the direct solver. The MA28 solution requires the least number of timesteps and Newtonian iterations. This is expected because of roundoff errors in the CG solutions which may permeate the Newtonian iterations, increasing their number and reducing the rate of growth of the timestep size. By tightening the convergence criterion to 10^{-8} the roundoff error from the CG iterations is significantly reduced, resulting in the same number of timesteps and marginally more Newtonian iterations (527 vs. 521) than in the MA28 solution.

Contrary to expectations, decreasing the convergence criterion to 10^{-8} on the RS/6000 does not lead to substantially larger execution times. This is due to (a) more accurate CG solutions, decreasing the need for more Newtonian iterations to satisfy the criterion for Newtonian convergence, as well as to (b) the fact that the residuals decrease very rapidly once below 10^{-6} . The tighter CG criterion has practically no effect on the DSLUCS solution (409 vs. 408 sec), very slightly increases the DSLUBC execution time (614 vs. 593 sec), and slightly decreases the DSLUGM time to 450 from 478 sec.

We also tested the various routines in 10 time step runs on a Macintosh Quadra 800 microcomputer. A different performance picture emerged on this platform. MA28 required 316 sec. All the CG methods significantly outperformed the direct solver. DSLUCS and DSLUGM required practically identical execution times. DSLUCS concluded the 10-time step simulation in 191 sec, 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} , 185 seconds were needed and the number of iterations per matrix solution varied between 6 and 10.

For a closure criterion of 10^{-10} DSLUGM required 191 sec, and between 10 and 14 iterations for convergence. For a closure criterion of 10^{-8} DSLUGM required 272.25 sec and 7 to 12 iterations per matrix solution. DSLUCS and DSLUGM were about 1.7 times faster than the direct solver on the Macintosh. DSLUBC was the slowest of the CG solvers but was still quite competitive (since it only required 7% more time), and significantly outperformed the MA28 routine.

The observation that the execution times do not necessarily increase with tighter convergence criteria should not mislead the reader into believing that this constitutes the normal and predictable behavior of the CG solvers. The reader is cautioned that such

behavior is atypical and more the exception rather than the rule. In this problem such behavior is encountered due to well-behaved matrices and a rapid decline in the residuals. In less well-posed problems one should expect a noticeable increase in execution times with a tighter convergence tolerance.

Table 8. Solver Performance in Test Problem 7					
Number of Equations: 2,400 Tolerance: $10^{-6}/10^{-8}$			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's (*)	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	97	521	-	-	450
DSLUBC	102/97	554/527	45/51	9/10	593/614
DSLUCS	110/97	594/527	45/49	6/8	408/409
DSLUGM	125/97	673/527	179/113	8/9	478/450
Number of Equations: 2,400 Tolerance: $10^{-8}/10^{-10}$			Computing Platform: Apple Macintosh QUADRA 800		
SOLVER	Number of Δt 's (†)	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	10	35	-	-	316
DSLUBC	10	35/35	14/16	9/11	198/209
DSLUCS	10	35/35	10/11	6/7	185/191
DSLUGM	10	35/35	12/14	7/8	186/191

(*): For a total simulation time of 3.16×10^{10} sec.

(†): Only 10 timesteps were run.

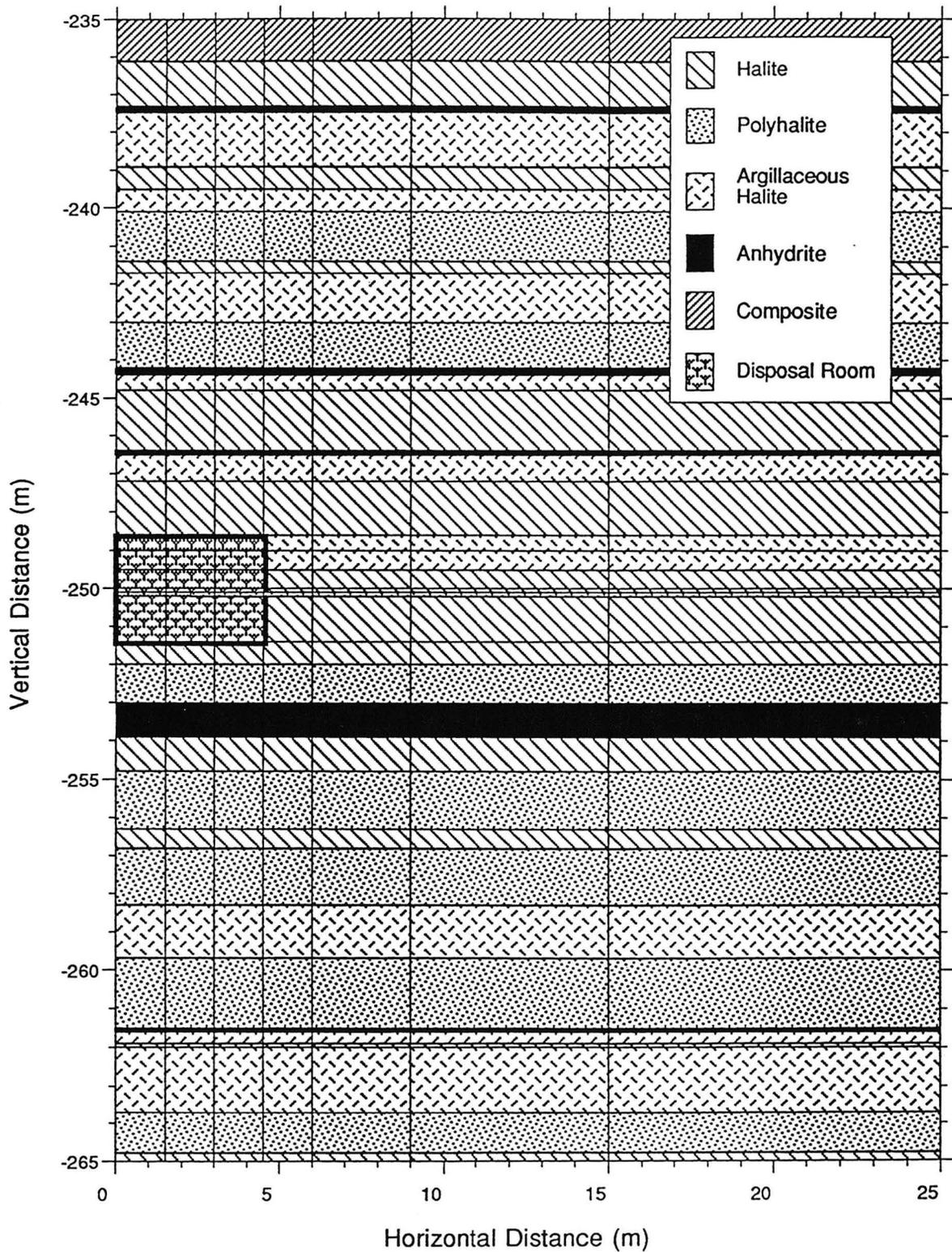


Figure 22. Schematic of typical WIPP underground rooms showing stratigraphic positions in Test Problem 7.

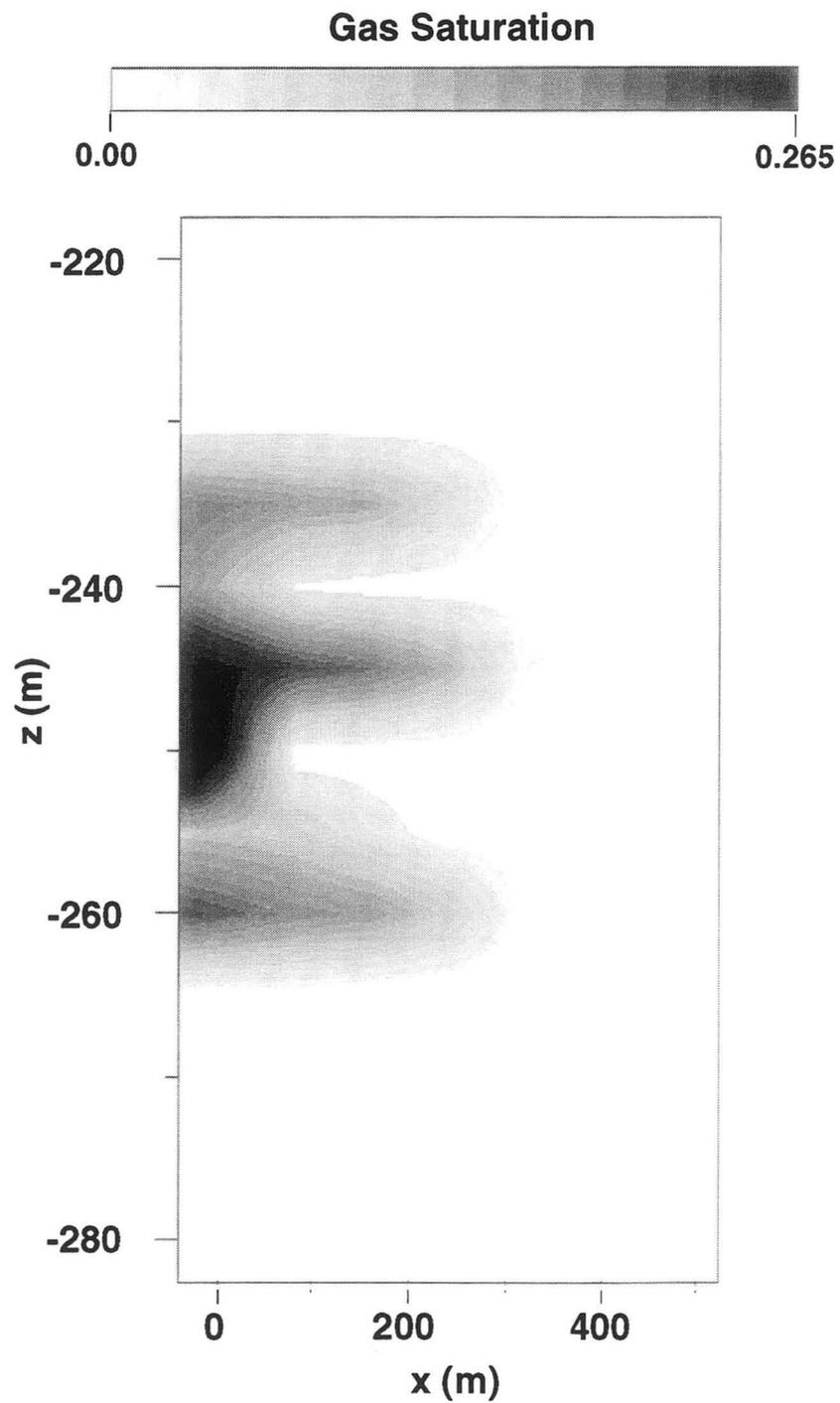


Figure 23. Gas saturation distribution in the subsurface in the vicinity of a WIPP underground room in Test Problem 7.

4.8. Test Problem 8: Saturated-Unsaturated Flow in a 3-D Medium (3-D, Two-Phase, Isothermal, Homogeneous and Heterogeneous)

4.8.1. Problem description

In Test Problem 8 we calculate gravity-capillary equilibrium in a three-dimensional domain with dimensions of $15\text{ m} \times 15\text{ m} \times 10\text{ m}$ in (x,y,z) . The domain was discretized into a $15 \times 15 \times 20$ grid in (x,y,z) , resulting in a total of 4500 gridblocks (Figure 24). The gridblocks below $z = -6\text{ m}$ were assigned to domain 'SATUR', and were initialized as single phase liquid. The rest of the domain was initialized at a saturation $S_l = 0.85$, corresponding to the irreducible gas saturation of $S_{gir} = 0.15$. A large 'top' block, assigned to domain 'ATMOS', with $P_{cap} = 0$ was attached to the top of the domain to simulate communication with the atmosphere.

We developed two data sets. The first had a homogeneous distribution of permeability (as in Table 4), and appears on the accompanying diskette under the name "PROB8". In the second data set a random field was generated using the method discussed in Test Problem 5. All permeabilities and capillary pressures were scaled according to Equations (22) and (23), i.e.

$$k \rightarrow \tilde{k} = k \cdot f_{PM} \text{ and } P_{cap} \rightarrow \tilde{P}_{cap} = P_{cap} / \sqrt{f_{PM}},$$

where the permeability multiplier f_{PM} is obtained from the logarithmic distribution

$$f_{PM} = \exp(-\beta s). \quad (24)$$

The term β is a scaling factor (a constant) and s is a random number between 0 and 1. In our case $\beta = 2.303$, leading to $0.1 \leq f_{PM} \leq 1$. As in Test Problem 5, the scaling results in a highly heterogeneous system where adjacent gridblocks may have significantly different permeabilities and capillary functions. We again use the data given in Table 4 for permeability, porosity, and characteristic curves.

The simulation of the homogeneous case (hereafter referred to as Problem 8) is performed with the EOS3 fluid property module, using $NK = NEQ = 2$ (i.e. isothermal conditions) in block MULTI to solve just two equations per grid block. This results in a total of $N = 9000$ equations (see file "PROB8" on the accompanying diskette). The same values of NK and NEQ were used in block MULTI in the heterogeneous case (hereafter referred to as Problem 8a). A special routine was used for the scaling of permeabilities and capillary pressures. This routine is not included in the generally available "regular" version of the TOUGH2 code, as it is a research routine developed for the investigation of flow and transport in very heterogeneous media. The input data set for Problem 8a is not included in the diskette accompanying this report.

The distribution of water saturation S_w in Problem 8 is shown in Figure 25. Due to the homogeneity of the domain, the same vertical distribution of S_w is observed at any vertical slice. The situation is different in the heterogeneous Problem 8a depicted in Figure 26, where the variability in capillary pressures results in locally different vertical S_w profiles. Problem 8a is much harder to solve than Problem 8 because of heterogeneity, and requires about 300 timesteps to reach steady-state at $t = 2.6667 \times 10^{10}\text{ sec}$ (845 years). Due to homogeneity, Problem 8 requires only 25 timesteps to reach steady-state at $t = 1.6667 \times 10^9\text{ sec}$ (52.8 years). This raises some interesting questions about the time-scale of such processes and the concept of steady-state.

4.8.2. Solver performance

The size and three-dimensional nature of the problem was prohibitive for the MA28 solver. The superiority of the DSLUCS solver in the solution of both problems (8 and 8a) is evident from Table 9. DSLUCS is 38% and 46% faster than DSLUBC (in the homogeneous and heterogeneous cases respectively), the second fastest solver. This is considerably less than the theoretically predicted maximum of a 2:1 speed advantage. DSLUGM is the slowest of the three routines, requiring 2.08 and 1.58 times the execution time of DSLUCS in Problems 8 and 8a respectively.

In Figure 27 the behavior of the residuals as a function of the number of matrix-vector multiplications (i.e. the CG iterations) follows the familiar patterns for the three routines. DSLUCS exhibits the sharpest decline in the residuals for the least number of iterations, and also shows non-monotonic behavior along with the largest and sharpest oscillations. DSLUBC shows a similar pattern, requires more CG iterations to convergence, and is characterized by more (although of lesser magnitude) oscillations. The monotonic decline in the residuals is evident in the DSLUGM solution. However, more iterations are required for the same residual level, and the DSLUGM solution begins to exhibit signs of stagnation, as evidenced by the almost flat portion of the tail end of the curve. In subsequent timesteps DSLUGM stagnated three times, reaching the maximum N_{CG} without achieving an acceptable convergence level.

Table 9. Solver Performance in Test Problem 8					
Number of Equations: 9,000 Homogeneous (a)			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	Could not solve the problem due to insufficient memory.				
DSLUBC	25	95	57	25	607
DSLUCS	25	95	77	15	439
DSLUGM	25	95	930(†)	23	913
Number of Equations: 9,000 Highly Heterogeneous (b)			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	Could not solve the problem due to insufficient memory.				
DSLUBC	300	1508	88	13	13,737
DSLUCS	300	1508	72	8	9,420
DSLUGM	300	1508	122	9	14,858

(†): The maximum number of allowable iterations (930) was reached only three times.

Discretization: 15x1m in x, 15x1m in y, 20x0.5 m in z

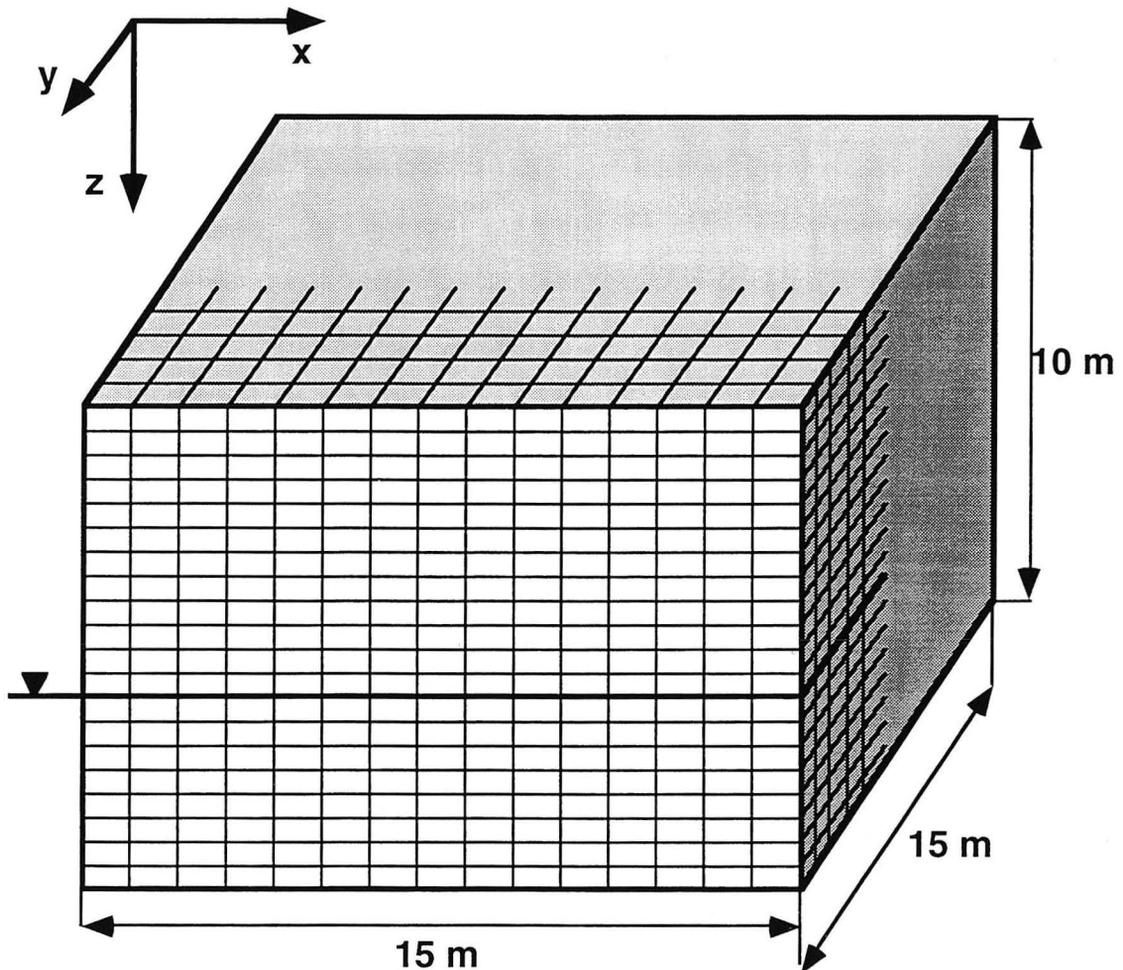


Figure 24. Discretization in Test Problems 8 and 8a.

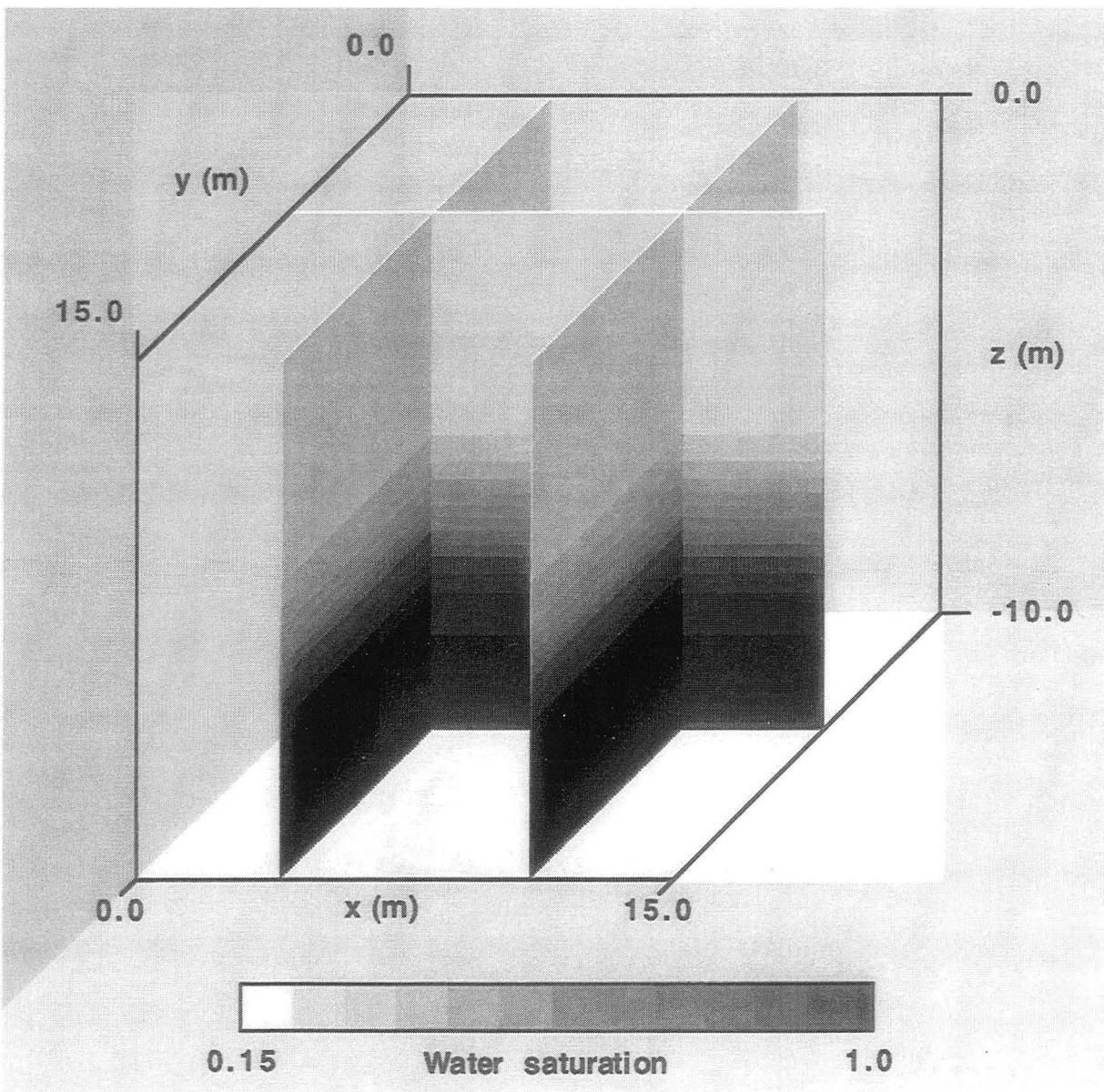


Figure 25. Water saturation distribution in the homogeneous three-dimensional domain of Test Problem 8.

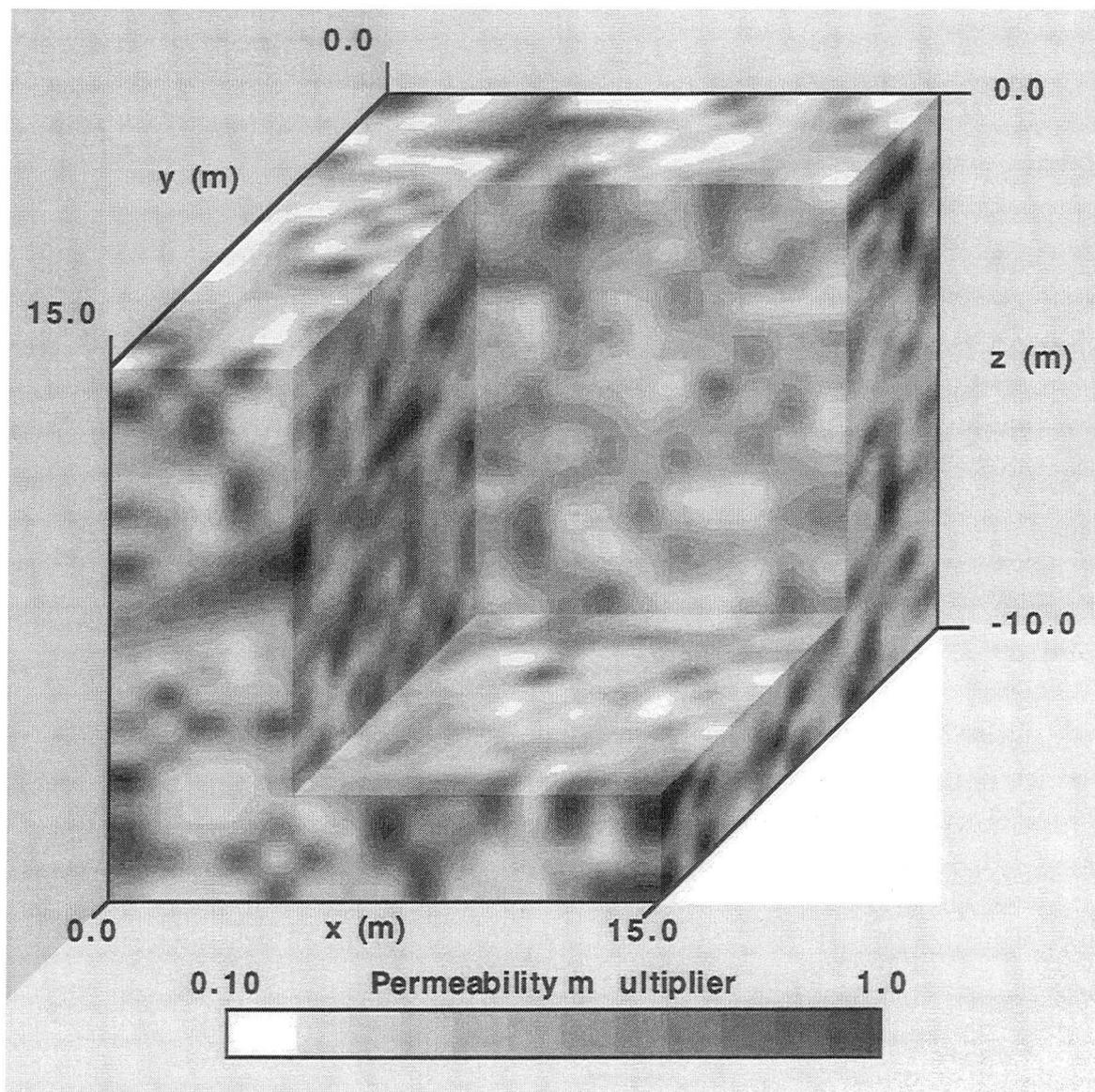


Figure 26. Permeability multiplier distribution in the highly heterogeneous three-dimensional domain of Test Problem 8a.

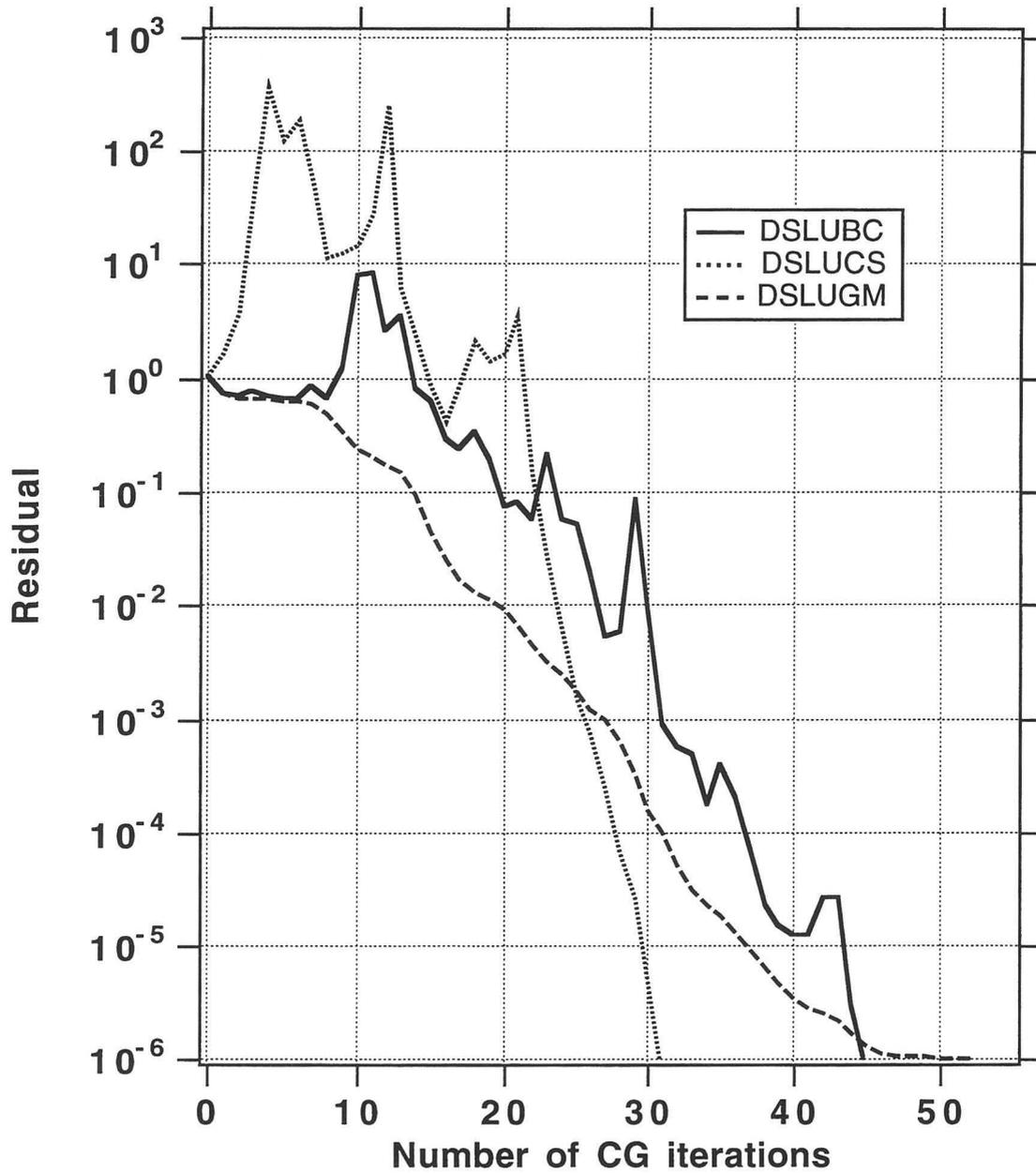


Figure 27. Relative performance of the CG solvers in the first Newtonian iteration of the 15th timestep in Test Problem 8a.

4.9. Test Problem 9: The Sandia Thermal Enhanced Vapor Extraction System - TEVES (3-D, Two-Phase, Non-Isothermal with Phase Changes)

4.9.1. Problem description

This is a modified version of a problem developed by *Webb and Phelan* [1995] to study the TEVES (Thermal Enhanced Vapor Extraction System) process being designed and built at Sandia National Laboratories. This is a demonstration of a process designed to extract solvents and chemicals contained in the Chemical Waste Landfill at Sandia National Laboratories. Figure 28 shows the general TEVES process setup. 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. A field demonstration of the process was conducted in the summer of 1994.

The behavior of the contaminants may be influenced by a number of factors including the soil heating rate and local temperature gradients, the evaporation rate of the liquid water and contaminants, the air flow rate, and the location of the vapor extraction borehole. If the air flow rate is too small, the heating and evaporation process could drive the contaminant vapors out of the heated zone and into the cooler unheated soil where they could condense, and contaminant migration into previously uncontaminated areas could occur. The soil is assumed to be initially at ambient conditions of 83.5 kPa and $20 \text{ }^\circ\text{C}$, and the heating and venting occur simultaneously. As time proceeds, the soil heats up and liquid water in the heated zone is vaporized and transported toward the borehole due to air sweep.

In order to address contaminant behavior issues, detailed two-phase calculations have been performed using a single-component NAPL version of TOUGH2. Variations in the borehole vacuum, vapor extraction locations, and soil permeability have been studied to evaluate the effect on system performance. The presence of NAPL required use of the recently released T2VOC module [*Falta et al.*, 1995].

In the variation described here the problem is simplified in that no NAPL is present in the system. In essence two-phase flow is assumed, and TOUGH2 with the EOS3 fluid property module for water, air, and heat is used. The simulated domain consists of 1300 elements in a three-dimensional grid. The simulation is performed using $NK = 2$ and $NEQ = 3$ (i.e. non-isothermal conditions) in block MULTI to solve three equations per grid block. This results in a total of $N = 3900$ simultaneous equations. We allowed a maximum of 50 time steps in the simulation. The data file for this problem can be found in the accompanying diskette under the name "PROB9". Figure 29 shows the increase in the temperature in the heated zone and surrounding soil at the end of the 50-timestep simulation ($t = 1.897 \times 10^6 \text{ sec}$).

4.9.2. Solver performance

A summary of the solver performance is provided in Table 10. Repeated attempts to solve this problem using the MA28 direct solver failed because of insufficient memory. All three CG solvers were able to provide an accurate solution, albeit at a different level of speed and efficiency.

In terms of numbers of CG iterations to convergence N_{CG} , all three CG solvers performed remarkably well and needed less than 1% of N . This indicates that this type of

problem creates matrices especially well suited to the CG solvers in the T2CG1 package. DSLUCS requires the least number of CG iterations (as well as the fewest Newtonian iterations), and DSLUBC the highest. In terms of execution speed, DSLUGM was the fastest (451 CPU sec), followed closely by DSLUCS (462 CPU sec). DSLUBC was significantly slower (619 CPU sec), and also required the largest N_{CG} .

Figure 30 shows the evolution of the residuals with the number of matrix-vector multiplications (i.e. the CG iterations) for the three CG routines in the first Newtonian iteration of the 50th timestep. The DSLUGM routine delivers a monotonic non-increasing evolution of the residuals in accordance with the properties of the underlying algorithm. The behavior of DSLUBC and DSLUCS follow the already established pattern of oscillatory decrease in the residuals. The DSLUBC residuals in this problem are uncharacteristically more oscillatory than the ones from DSLUCS. The superiority of DSLUCS over the DSLUBC routine is also evident.

Table 10. Solver Performance in Test Problem 9					
Number of Equations: 3,900			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	Could not solve the problem due to insufficient memory.				
DSLUBC	50	249	36	8	619
DSLUCS	50	239	20	4	462
DSLUGM	50	250	28	7	451

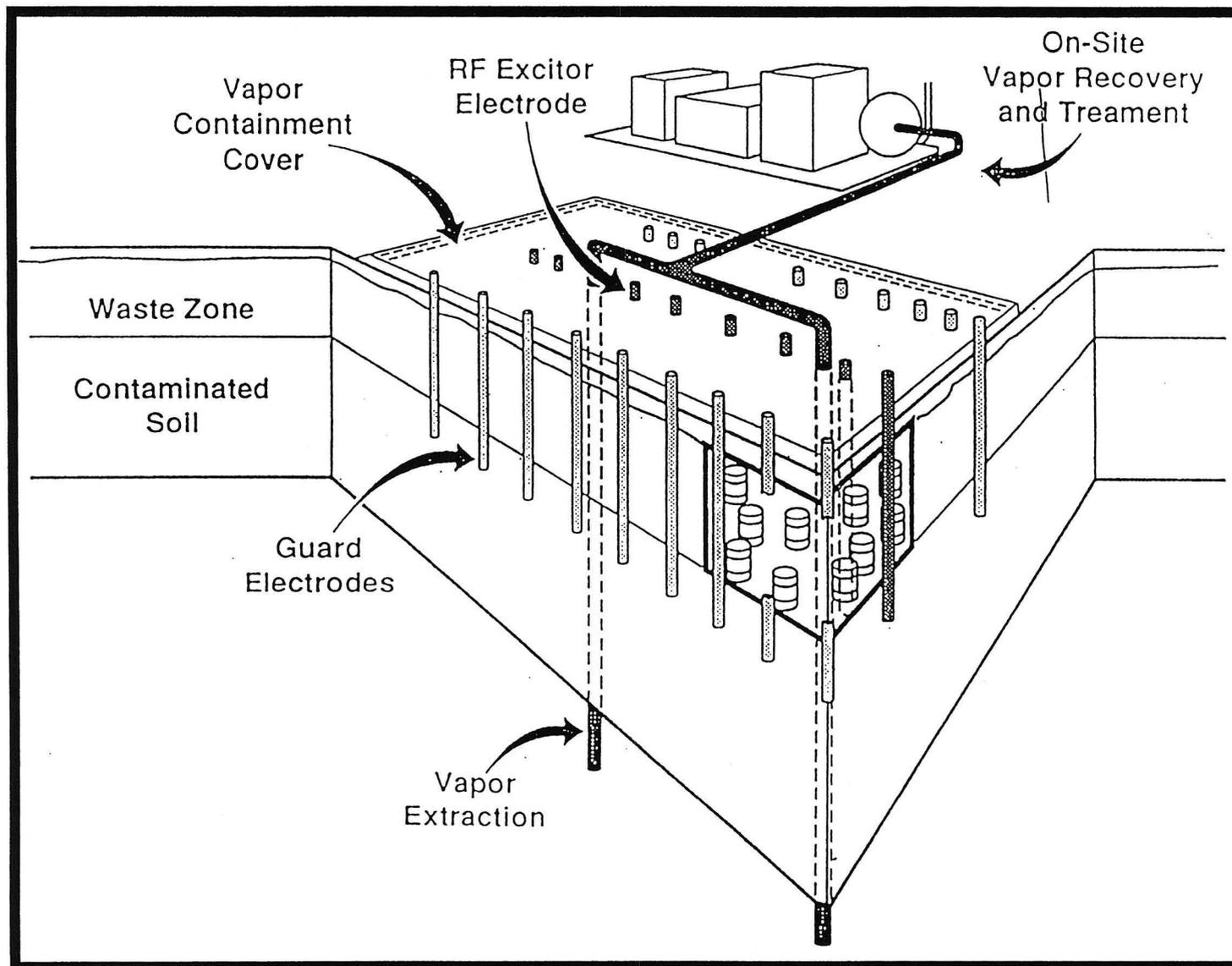


Figure 28. Schematic of the Thermal Enhanced Vapor Extraction System (TEVES) in Test Problem 9.

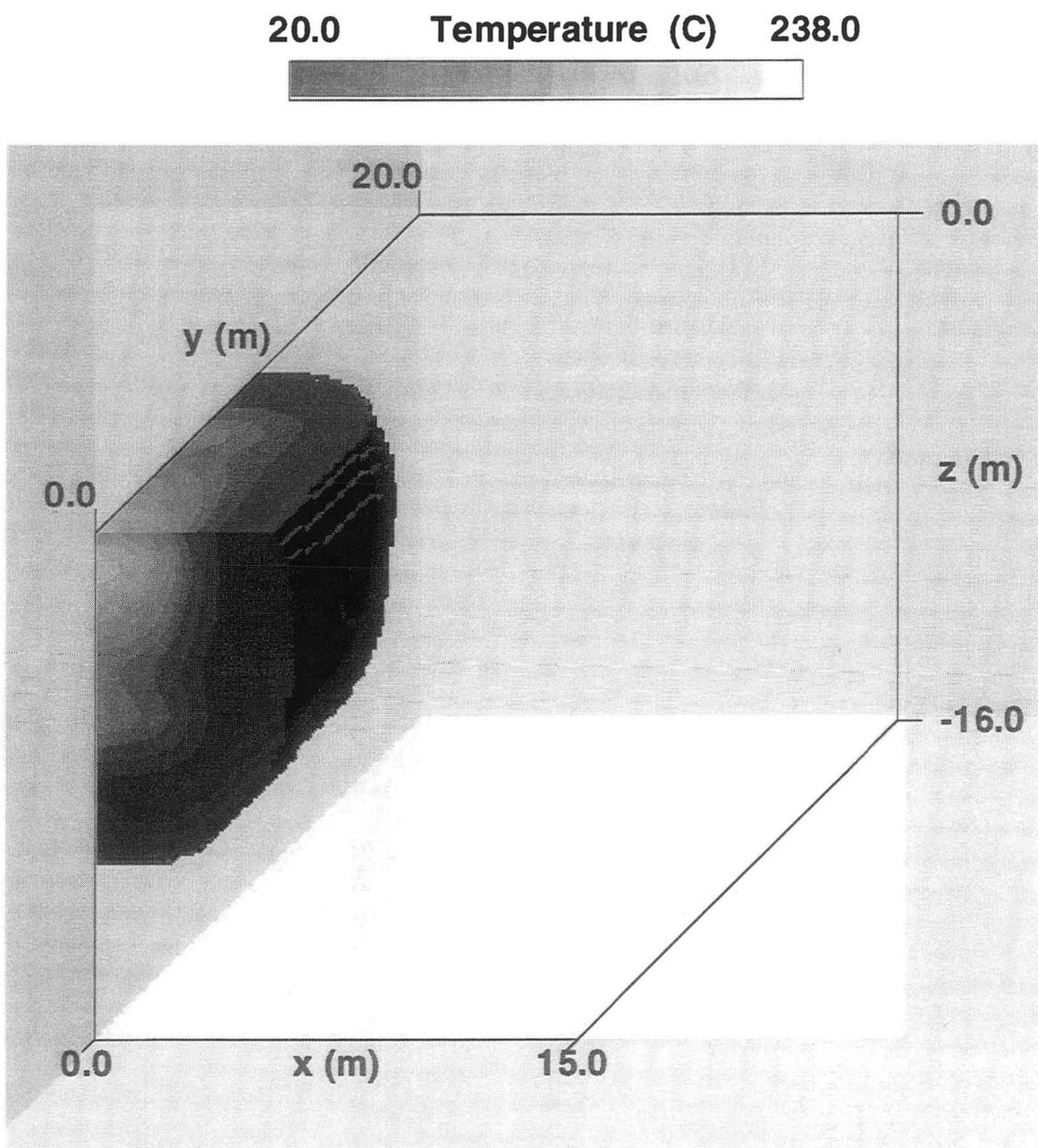


Figure 29. Temperature distribution at $t = 1.897 \times 10^6$ sec in the TEVES system simulated in Test Problem 9.

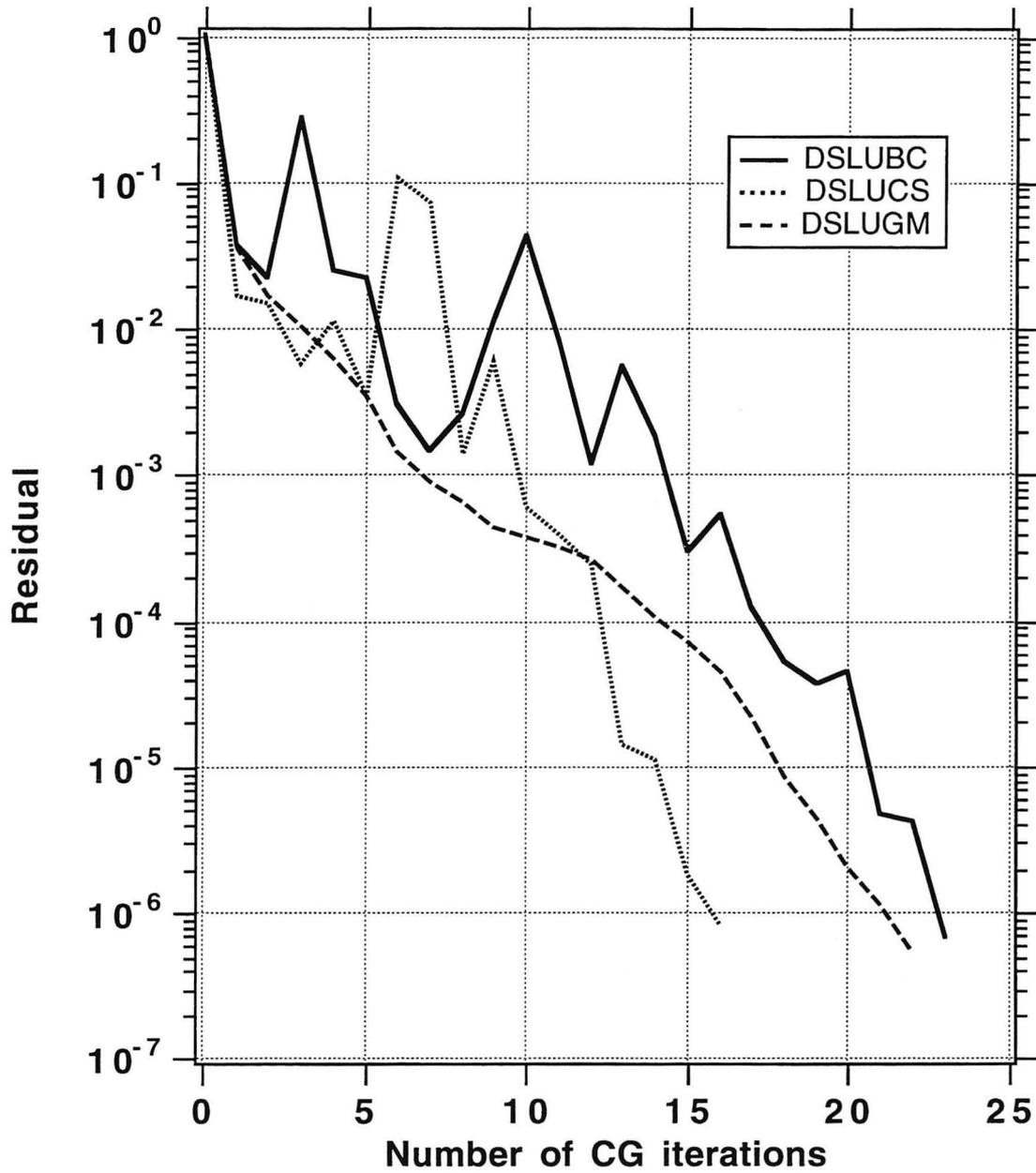


Figure 30. Relative performance of the CG solvers in the first Newtonian iteration of the 50th timestep in Test Problem 9.

4.10. Test Problem 10: 3-D Geothermal Reservoir Model (3-D, Single- and Two-Phase, Non-Isothermal with Phase Changes)

4.10.1. Problem description

Five three-dimensional simulation models with different discretizations were constructed [Antúnez *et al.*, 1994]. The different discretizations ranged between 500 and 10,000 elements and resulted in 1,000 to 20,000 equations in single- and two-phase systems. The simulation models have an areal extent of $5 \times 4 \text{ km}$ (20 km^2) and a thickness of 1000 m, divided in ten layers of 100 m each (Figure 31). The same discretization of the vertical reservoir dimension was maintained in all cases, but finer discretizations were used in the x and y directions. The various discretizations are given in Table 11.

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 m, 500 m, 550 m) for the producer and (4500 m, 3500 m, 250 m) for the injector. The heat source is distributed among the required elements to cover an area of $4 \times 10^5 \text{ m}^2$ (1000 m in x and 400 m in y) at the center of bottom layer (Figure 31).

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 $280 \text{ }^\circ\text{C}$ in all blocks; for the two-phase cases, 10 MPa and a gas saturation $S_g = 0.20$ 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.

TOUGH2 was used with the EOS1 fluid property modules with $NK = 1$ and $NEQ = 2$. The simulations were run for a total of 25 timesteps. A model input data file (corresponding to the 5,000 element problem) for this problem appears under the name "PROB10" on the diskette accompanying this document. All relevant reservoir properties can be found in the same file. The grid was generated with the MESHMAKER facility built into the TOUGH2 code [Pruess, 1991]. In Figure 32 we present the three-dimensional distribution of temperature (expressed as the deviation from the original $280 \text{ }^\circ\text{C}$) in the most active zones of the domain, i.e. in the vicinity of the heater and the producing well, at $t = 1.33 \times 10^9 \text{ sec}$. Figure 33 shows the three-dimensional liquid saturation of the two-phase flow problem at $t = 6.57 \times 10^8 \text{ sec}$ in the same areas.

4.10.2. Solver performance

The performance of the various solvers is described in Tables 12 and 13. Table 12 presents exclusively the 5,000-cell problem solved on the IBM RS/6000 370 computer for both single and two-phase flow conditions ($N = 10,000$). MA28 could not tackle this problem due to insufficient memory. The maximum number of allowable CG iterations in Test Problem 9 is $N_{CG} = 0.2N$.

The unequivocal superiority of the DSLUCS routine is obvious in both the single- and the two-phase case. Under single-phase conditions DSLUCS is 4.02 times faster than DSLUBC and 6.30 times faster than DSLUGM, and it also covers more simulation time due to more accurate solutions which, in turn, allow larger timesteps. Under two-phase conditions DSLUCS is 3.42 times faster than DSLUBC and 1.59 times faster than DSLUGM. The maximum number of CG iterations in DSLUCS and DSLUGM is significantly lower than the maximum allowable (while in the single-phase case it is often

reached). The maximum simulation time is the same for DSLUCS and DSLUGM, but is somewhat smaller for DSLUBC.

It is interesting to note that when using DSLUGM the two-phase problem requires significantly lower execution times. One would expect that the execution times would be comparable (since N is the same), as is invariably the case in direct solvers. This is not always the case in CG (as well as in other iterative) methods, where such case-specific and often counterintuitive behavior is frequently encountered. In this case, this behavior of DSLUGM is due to the fact that the matrix is not well behaved (because of the low water compressibility under single-phase conditions) and this routine is not well-suited for the problem.

In Table 13 we present the solver evaluation results obtained from runs on a 486-DX2-66 MHz PC for grid sizes ranging between 500 and 10,000 cells (N varying between 1,000 and 20,000). The performance of the solvers is very similar to that described in Table 13. DSLUCS is clearly superior to any other alternative in this problem. It invariably needs the least execution time (often less than a third of the execution times of the other routines), while covering the largest simulation time. We conducted the single-phase simulations on the PC using all four solvers. For the two-phase simulations we only used the DSLUCS solver.

Another important conclusion from Tables 12 and 13 is the effect of the computing platform and microprocessor architecture on the relative performance of the solvers. Although DSLUCS is the fastest among the solvers on both platforms, its relative speed is significantly higher on the IBM RS/6000 370 than on the PC in the 5,000-cell problem. On the PC the execution times under two-phase conditions are consistently lower than the CPU times for single-phase flow; on the IBM RS/6000 370 such behavior occurs only occasionally. These observations underscore the problem-, compiler- and machine-specificity which characterize the performance of the CG methods, as well as any iterative methods.

Figures 34 and 35 show the error reduction with the number of CG iterations for the single- and two-phase cases respectively. These figures demonstrate the evolution of the residuals in the first Newtonian iteration of the 11th timestep in the 5,000-cell problem. The superiority of DSLUCS is evident. In two-phase flow far fewer iterations are needed to achieve the 10^{-6} convergence criterion than in the single-phase case, indicating a better behaved matrix. DSLUGM produces its usual monotonic non-increasing evolution of the residuals in accordance with the properties of the underlying algorithm. Note that in the single-phase case the residuals are reduced in a series of steps characterized by local plateaus which is not very satisfactory. The DSLUBC and DSLUCS residuals in single-phase demonstrate pronounced oscillatory behavior, and the DSLUBC residuals are uncharacteristically more oscillatory than the ones from DSLUCS. The same oscillatory pattern is also present in Figure 34, but the magnitude of the oscillations is far smaller.

Table 11. Domain Discretizations in Test Problem 10

Case Number	Subdivisions in x	Subdivisions in y	Subdivisions in z	Total number of cells
1	5	10	10	500
2	10	10	10	1,000
3	10	20	10	2,000
4	25	20	10	5,000
5	50	20	10	10,000

Table 12. Solver Performance in Test Problem 10 (5000 cells)

Number of Equations: 10,000 Single-phase			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	Could not solve the problem due to insufficient memory.				
DSLUBC	25	105	2001(†)	6	5,191
DSLUCS	25	102	2001(*)	3	1,291
DSLUGM	25	104	2001(+)	6	8,127
Number of Equations: 10,000 Two-phase			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	Could not solve the problem due to insufficient memory.				
DSLUBC	25	122	3001(^)	5	6,542
DSLUCS	25	132	125(#)	3	1,909
DSLUGM	25	132	540(#)	5	3,035

(†): The maximum number of allowable iterations (2001) was reached 13 times - $t_{max} = 1.33 \times 10^9$ sec.

(*): The maximum number of allowable iterations (2001) was reached once - $t_{max} = 4.99 \times 10^8$ sec.

(+): The maximum number of allowable iterations (2001) was reached 38 times - $t_{max} = 1.32 \times 10^9$ sec.

(#): $t_{max} = 6.57 \times 10^8$ sec.

(^): The maximum number of allowable iterations (3001) was reached 5 times - $t_{max} = 5.55 \times 10^8$ sec.

**Table 13. Solver Performance on a PC in Test Problem 10
(Variable Grid Size)**

Number of equations (Phase No.)	Grid size	SOLVER	Number of Δt 's	Simulated time (sec)	Newtonian Iterations	CPU Time (sec)
1,000 (1)	500	MA28	25	7.1677E9	96	791
		DSLUGM	25	4.5053E9	108	871
		DSLUBC	25	4.7101E9	104	611
		DSLUCS	25	4.7101E9	97	518
2,000 (1)	1,000	MA28	25	19.660E9	96	5,552
		DSLUGM	25	4.0957E9	126	4,593
		DSLUBC	25	6.9629E9	103	2,247
		DSLUCS	25	4.3005E9	98	1,726
4,000 (1)	2,000	MA28	25	4.3005E9	100	17,041
		DSLUGM	25	1.2285E9	144	25,878
		DSLUBC	25	1.3821E9	112	11,948
		DSLUCS	25	1.9453E9	99	6,656
10,000 (1)	5,000	MA28	Insufficient memory			
		DSLUGM	25	1.7250E8	101	94,437
		DSLUBC	25	5.5501E8	106	48,964
		DSLUCS	25	7.1650E8	102	31,500
20,000 (1)	10,000	MA28	Insufficient memory			
		DSLUGM	15	4.1110E7	57	187,876
		DSLUBC	15	2.1730E8	109	163,327
		DSLUCS	15	3.3250E8	97	106,602
1,000 (2)	500	DSLUCS	25	9.0900E7	140	485
2,000 (2)	1,000	"	25	9.0900E7	122	1,017
4,000 (2)	2,000	"	25	9.0900E7	134	2,910
10,000 (2)	5,000	"	25	3.8100E7	132	7,733
20,000 (2)	10,000	"	25	2.5300E7	134	18,203

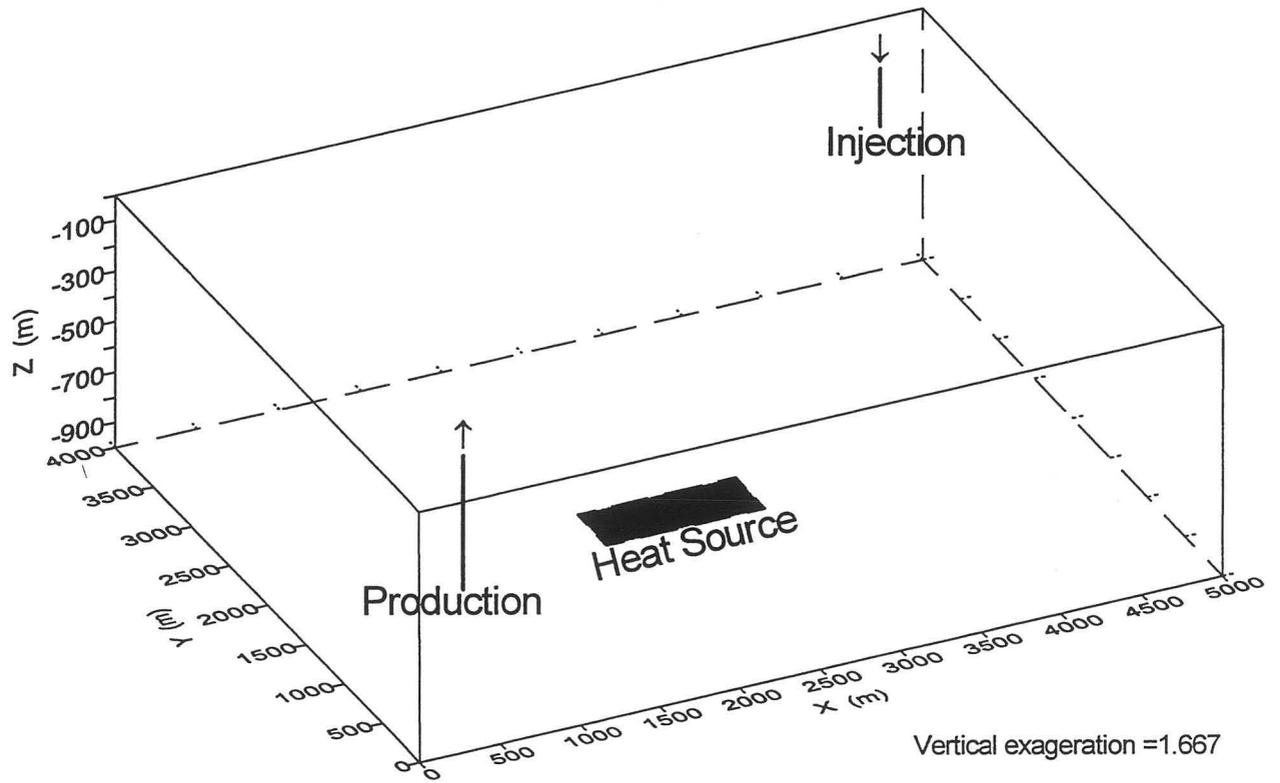


Figure 31. The simulation domain in Test Problem 10.

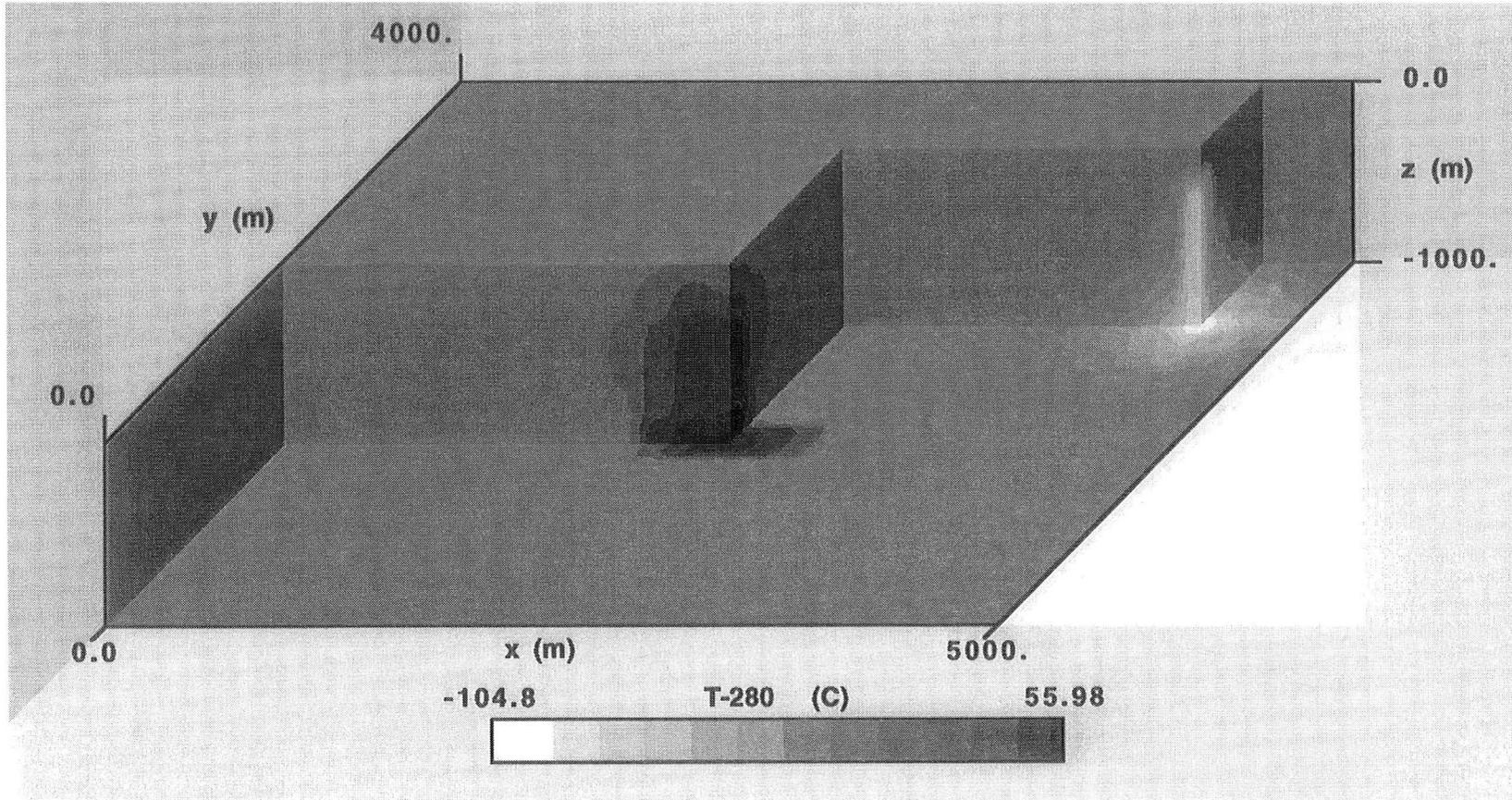


Figure 32. Test Problem 10: temperature distribution at $t_{max} = 1.33 \times 10^9$ sec (single-phase geothermal reservoir, 5000 cells).

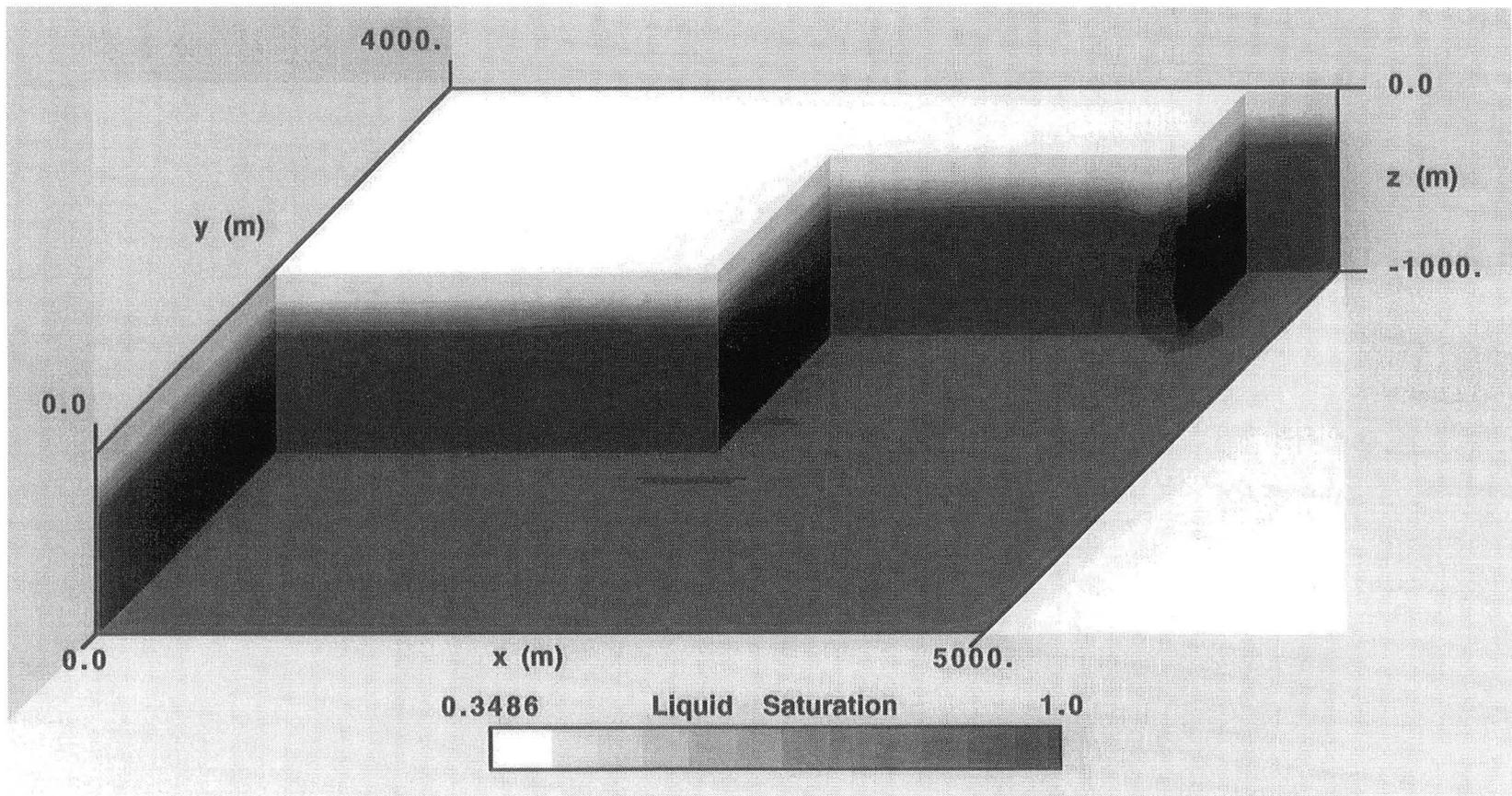


Figure 33. Test Problem 10: liquid saturation distribution at $t_{max} = 6.57 \times 10^8$ sec (two-phase geothermal reservoir, 5000 cells).

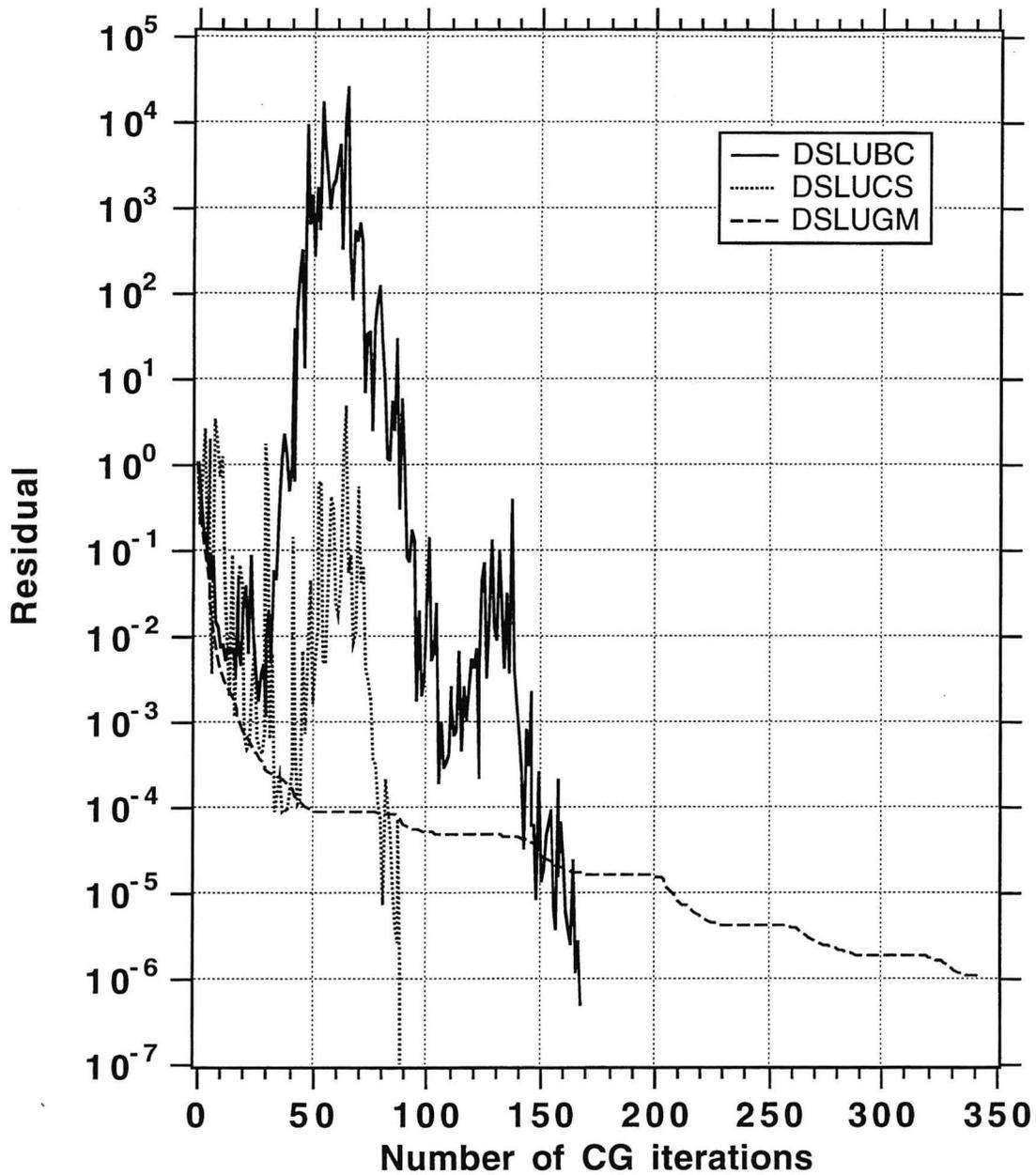


Figure 34. Performance of the CG solvers in the first Newtonian iteration of the 11th timestep in Test Problem 10 (5000 gridblocks, $N = 10,000$ equations, single-phase conditions).

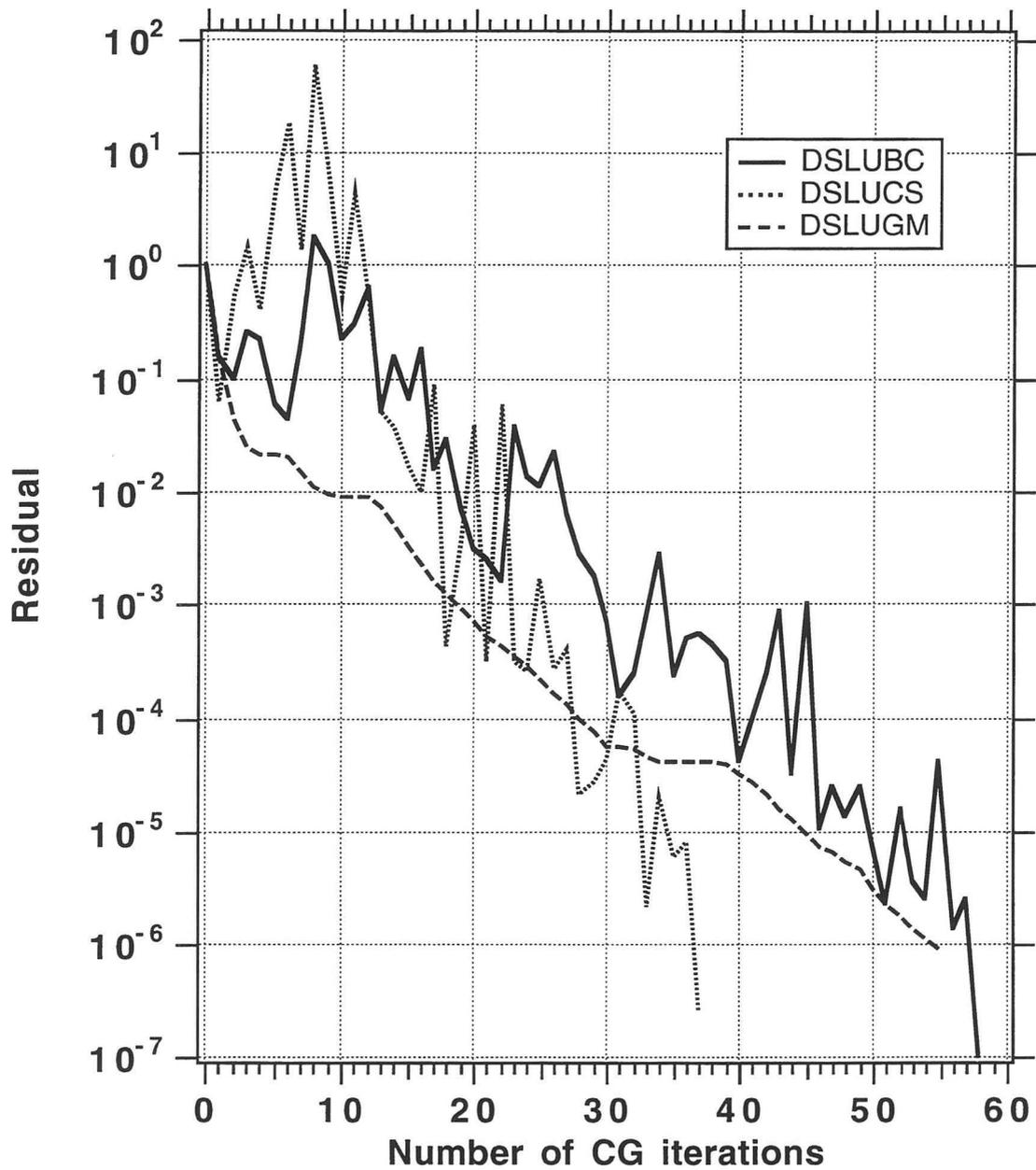


Figure 35. Performance of the CG solvers in the first Newtonian iteration of the 11th timestep in Test Problem 10 (5000 gridblocks, $N = 10,000$ equations, two-phase conditions).

4.11. Test Problem 11: Cerro Prieto Geothermal Field (3-D with Irregular Grid, Two-Phase, Non-Isothermal with Phase Changes and Matrix/Fracture Interactions)

4.11.1. Problem description

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, 1992]. Currently most of the brine is sent to evaporation ponds that cover an area of 18.6 km^2 (Figure 36). 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 $20 \text{ }^\circ\text{C}$) 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 (Figure 36).

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, 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 and was developed as a single porosity model [Antúnez and Lippmann, 1993]. Using 3 and 5 MINC shells resulted in discretizations of 5,644 to 8,466 elements and 11,288 to 16,932 equations. TOUGH2 was compiled using the EOS1 fluid property module for the two-phase simulations with $\text{NK} = 1$ and $\text{NEQ} = 2$. The MINC facility in MESHMAKER was used to describe the matrix/fracture interactions. The model was calibrated with production and piezometric data, and was used to test several injection strategies.

In the Cerro Prieto model the following schedule was followed: Inject $3.5 \times 10^6 \text{ kg/h}$ of $20 \text{ }^\circ\text{C}$ 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 $5.459 \times 10^6 \text{ kg/h}$ of steam and $6.394 \times 10^6 \text{ kg/h}$ of separated brine).

4.11.2. Solver performance

We made three sets of runs on a 486-DX2-66 MHz PC, corresponding to the three different grid systems. The number of simultaneous equations varied from 2,822 to 16,932. The results in Table 14 follow the observation made in test Problem 10. DSLUCS is the fastest of the three CG solvers by a wide margin (from 1.2 to 2.1 times faster), and needs the smallest number of Newtonian iterations to cover the desired simulation period. DSLUGM is consistently the slowest routine and does not appear a good choice for this type of problem.

**Table 14. Solver Performance on a PC in Test Problem 11
(Variable Grid Size, Irregular Grid)**

Number of equations	Grid size	SOLVER	Number of Δt 's	Simulated time (sec)	Newtonian Iterations	CPU Time (sec)
2,822	1,411	MA28	Insufficient memory			
		DSLUBC	25	4.8061E8	151	3,920
		DSLUCS	25	4.8061E8	145	2,596
		DSLUGM	25	4.8061E8	156	5,486
11,288	5,644	MA28	Insufficient memory			
		DSLUBC	25	4.1003E8	203	13,245
		DSLUCS	25	4.1003E8	194	10,462
		DSLUGM	25	4.1003E8	226	19,996
16,932	8,466	MA28	Insufficient memory			
		DSLUBC	25	4.1003E8	186	21,818
		DSLUCS	25	4.1003E8	179	16,987
		DSLUGM	25	4.1003E8	194	31,703

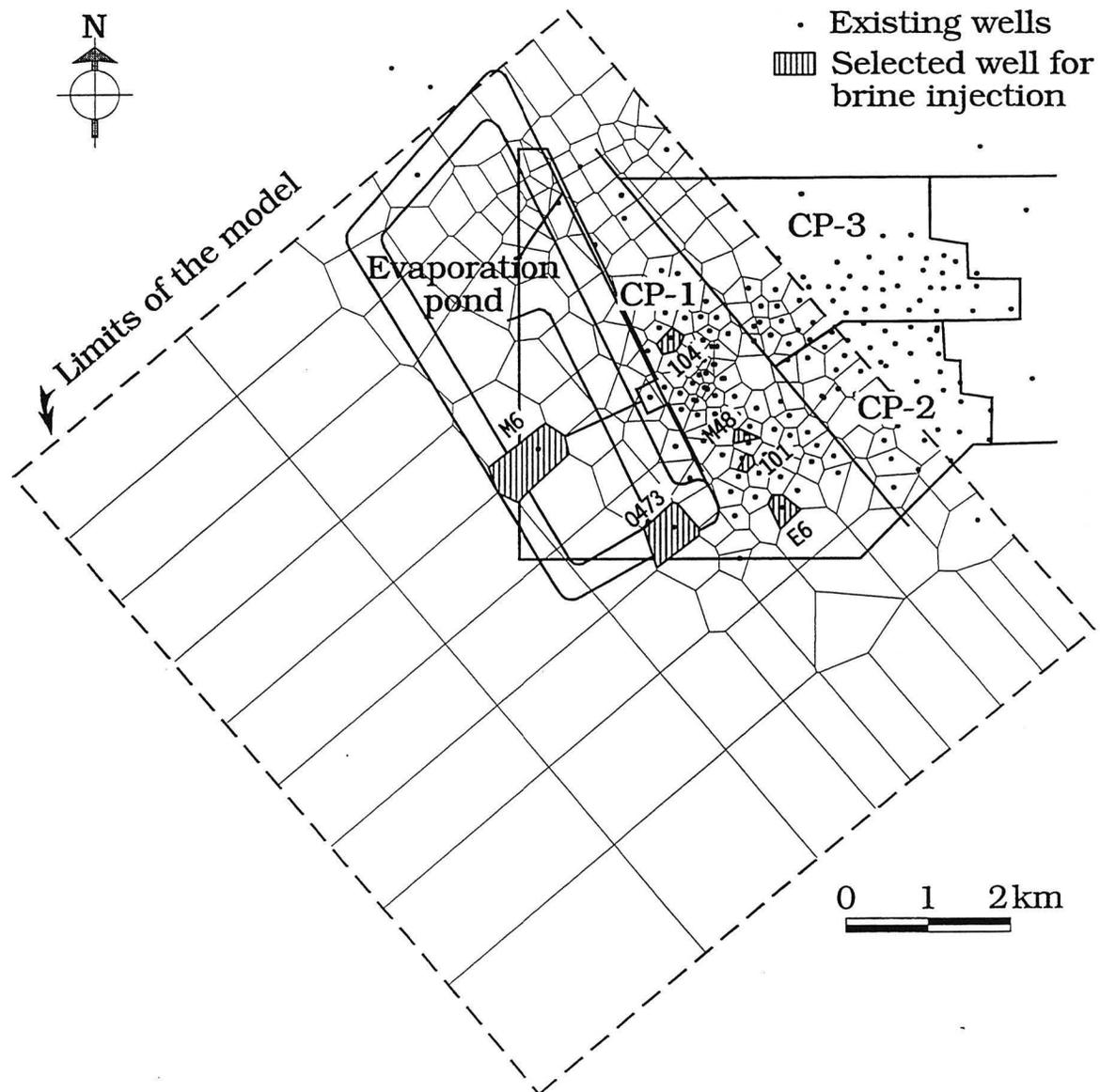


Figure 36. Characteristics of the irregular computational grid of the Cerro Prieto model in Test Problem 11.

4.12. Test Problem 12: Repository Performance Assessment at Yucca Mountain
(Cylindrical 2-D, Two-Phase Flow in Fractured Media, Non-Isothermal with Phase Changes)

4.12.1. *Problem description*

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 Figure 37). 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 liquid saturation distribution after $t = 5.6 \times 10^5$ sec (6.48 days) appears in Figure 38. It shows a narrow region of diminished liquid saturation right at the repository horizon (375 m depth).

4.12.2. *Solver performance*

The problem was run on an Apple Macintosh QUADRA 800 microcomputer. Execution time for 10 time steps, corresponding to a simulated time of 6.48 days, was 924 seconds for the MA28 direct solver [*Duff, 1977*]. All three CG routines in T2CG1 were significantly faster than the MA28. DSLUGM was the fastest, and required 691 CPU seconds. The DSLUCS time requirement was practically the same (698 sec), while DSLUBC was slightly slower. For the specified convergence criterion of 10^{-6} only 3-8 iterations were needed for each equation solution, indicating the efficiency of the CG solutions.

Table 15. Solver Performance in Test Problem 12					
Number of Equations: 1,890			Computing Platform: Apple Macintosh QUADRA 800		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	10	47	-	-	924
DSLUBC	10	47	8	4	716
DSLUCS	10	47	5	3	698
DSLUGM	10	47	6	4	691

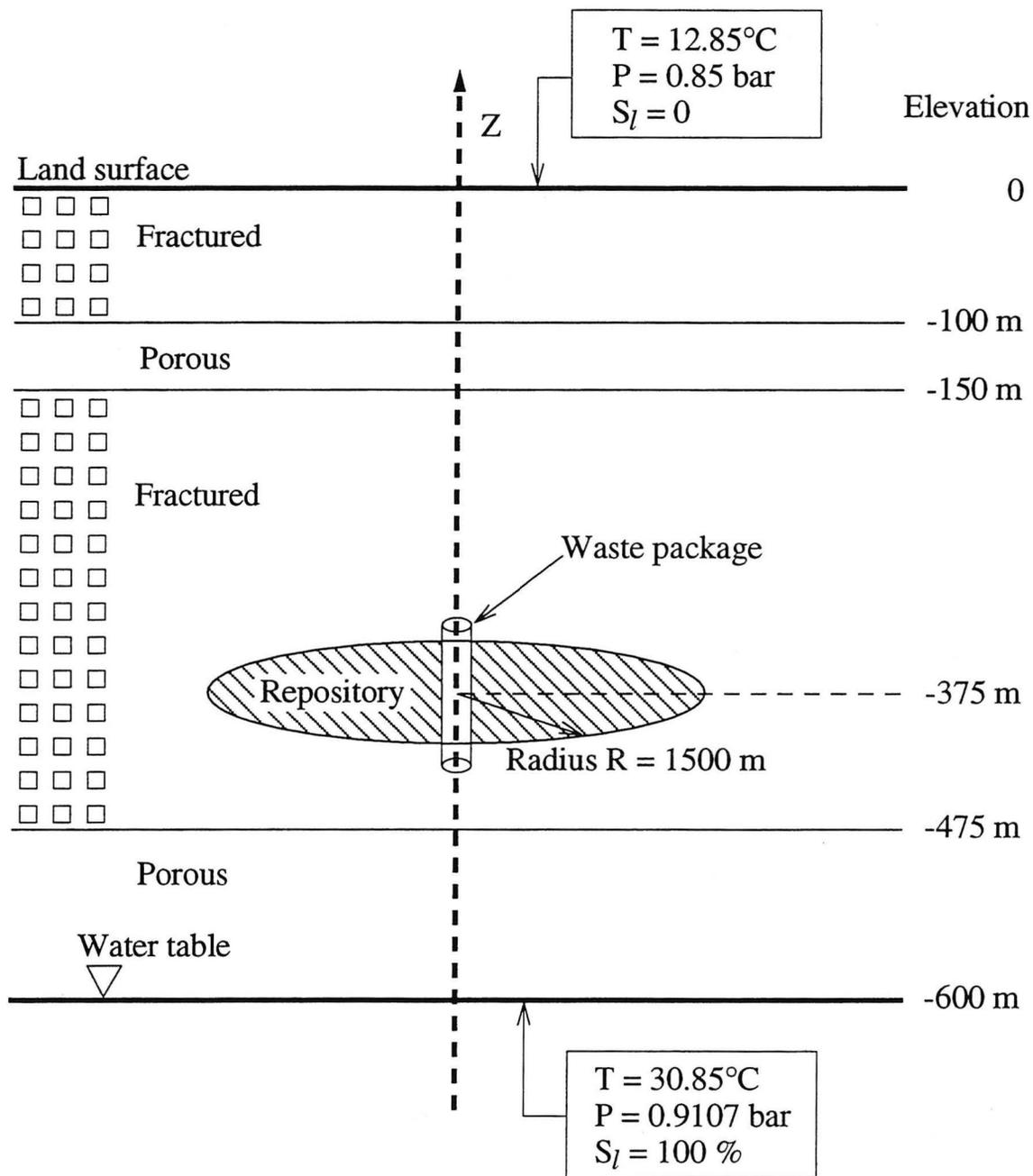


Figure 37: Two-dimensional radially symmetric (r, z) model of a high-level nuclear waste repository at Yucca Mountain.

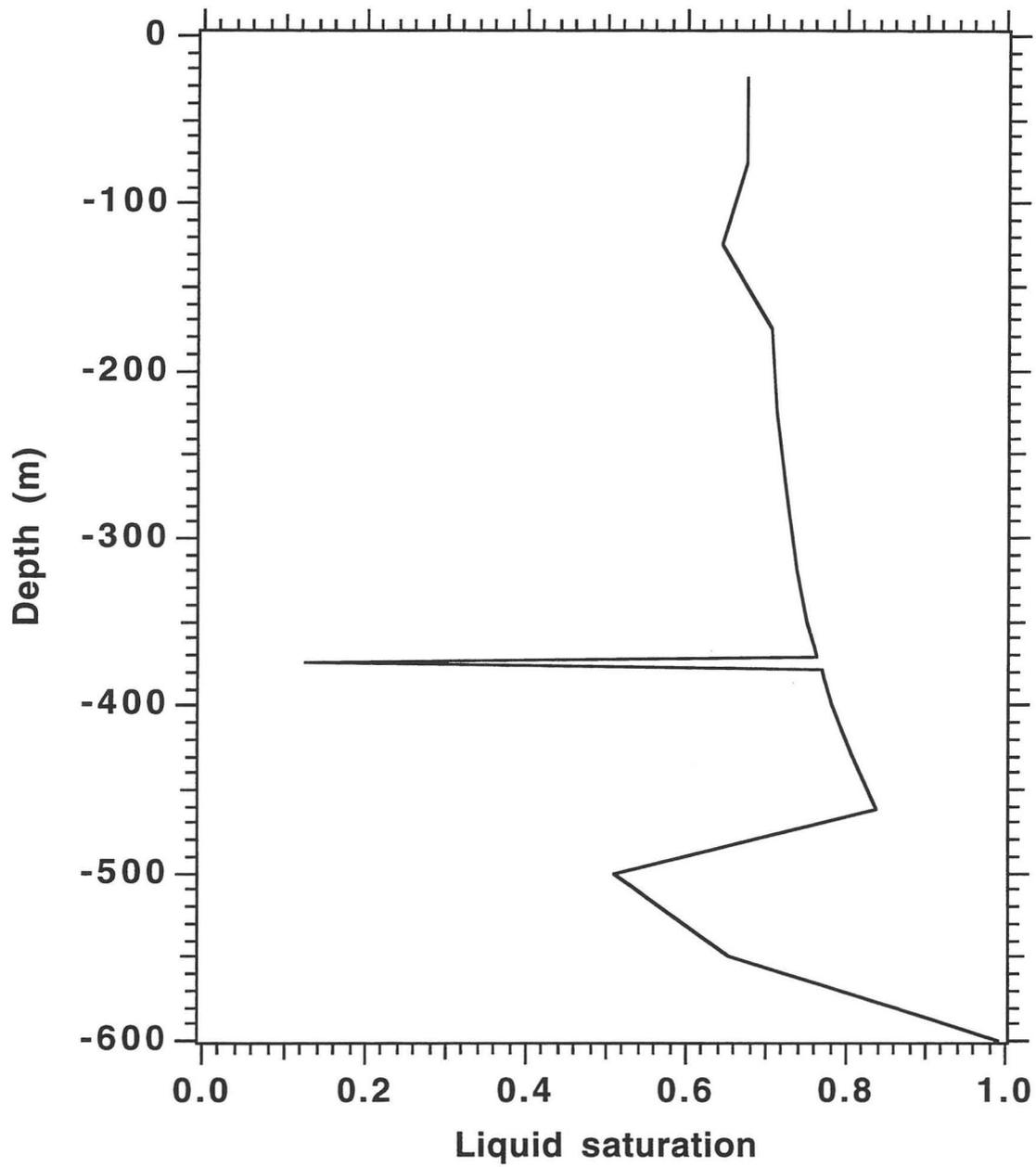


Figure 38. Liquid saturation distribution at $r = 0.1515 \text{ m}$ and $t = 5.6 \times 10^5 \text{ sec}$ in Test Problem 12.

4.13. Test Problem 13: Channelized Water Flow at Yucca Mountain (3-D, Two-Phase Flow in Fractured Media, Non-Isothermal with Phase Changes)

4.13.1. Problem description

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 model in (x,y,z) with $6 \times 8 \times 21 = 1008$ 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 Figure 39. The gridding in the vertical (z) direction is identical to the previous (r,z) model. The first Δy has a width of 1 mm and represents a fracture with a high permeability of $9 \times 10^{-12} m^2$.

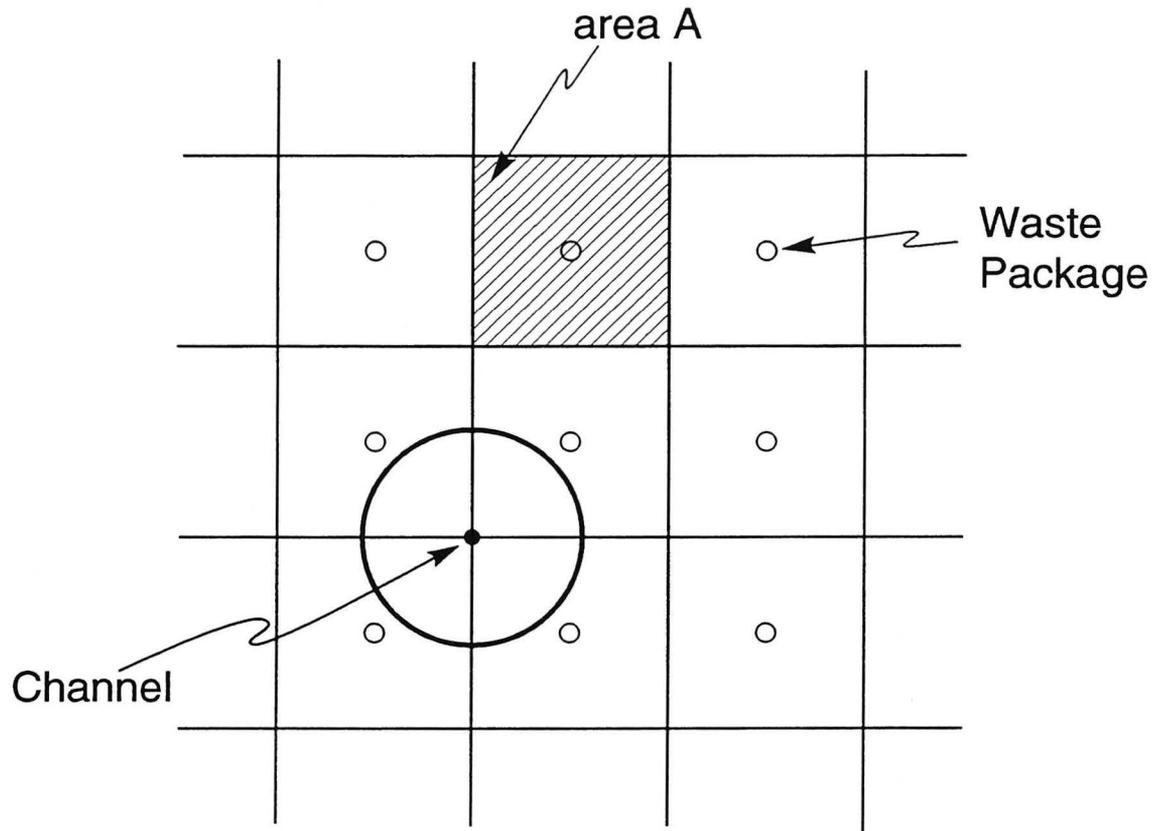
The issue to be addressed by the model is whether vapor generated near the waste packages can be discharged into fractures and then condense at some distance from the waste packages in a sufficiently focused manner to cause rapid and persistent downflow of water past the repository horizon. With the Topopah Spring matrix rock assigned a permeability of $1.9 \times 10^{-18} m^2$, heterogeneity is rather strong with a maximum permeability contrast of approximately 5×10^6 . The liquid saturation distribution at the end of the simulation period ($t = 365.25$ years) is shown in Figure 40. More details can be found in Pruess and Tsang [1994].

4.13.2. Solver performance

Using the EOS4 fluid property module, $3 \times 1008 = 3024$ simultaneous equations are solved. The simulations were again carried out in 64-bit arithmetic on an IBM RS/6000 workstation. 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.

DSLUCS is again the fastest solver, requiring 36 time steps and 582 CPU-seconds to complete the simulation. The linear equation solution with a specified convergence tolerance of 10^{-6} required between 9 and 56 iterations (i.e. a maximum $N_{CG}/N = 1.85\%$), testifying to the efficiency of the method. DSLUGM was a close second, while DSLUBC was measurably slower. Figure 41 shows the convergence of the CG solutions at the first Newtonian iteration of the 10th timestep. The smooth monotonic decline of the DSLUGM solutions and the oscillatory behavior of DSLUBC and DSLUCS are evident.

Table 16. Solver Performance in Test Problem 13					
Number of Equations: 3,024			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28(*)	Could not solve the problem				
DSLUBC	34	165	65	11	687
DSLUCS	36	173	56	9	582
DSLUGM	34	166	170	10	598



ESD-9304-0001

Figure 39. Areal view of an idealized repository with vertical waste package emplacement. Each waste package occupies an area A.

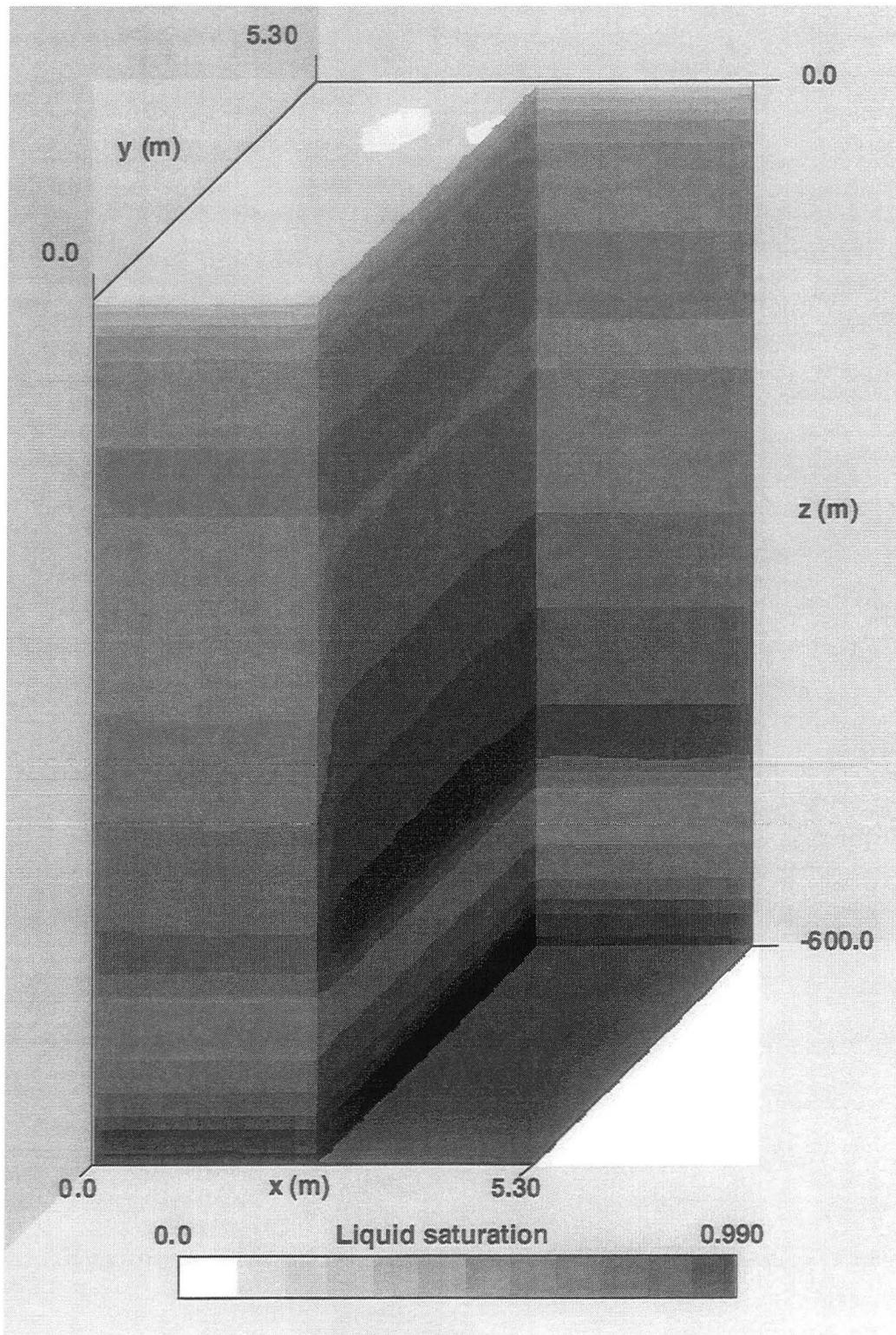


Figure 40. Distribution of liquid saturation in the three-dimensional simulation domain of test Problem 13 at $t = 365.25$ years.

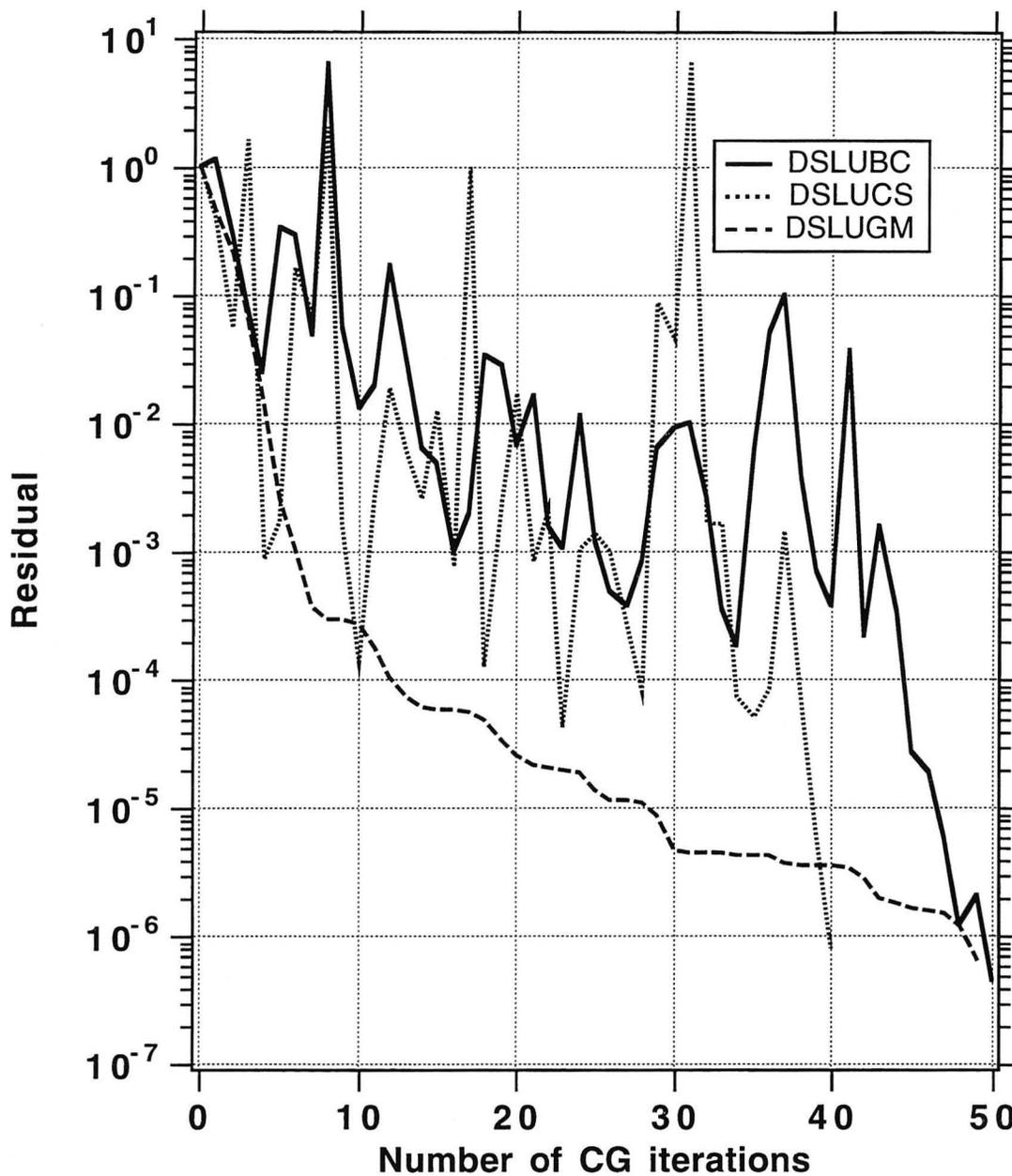


Figure 41. Performance of the CG solvers in the first Newtonian iteration of the 10th timestep in Test Problem 13 (1008 gridblocks, $N = 3,024$ equations, two-phase conditions).

4.14. Test Problem 14: Solutal Convection in Porous Media (2-D, Coupled Groundwater Flow and Solute Transport, Isothermal)

4.13.1. Problem description

In this problem, we use the dispersion module T2DM [Oldenburg and Pruess, 1993] to simulate the Elder free convection problem. Originally conceived as being pure thermal convection with heating from below, the Elder problem is equivalent to pure solutal convection with a salt source at the top [Voss and Souza, 1987]. The large maximum density change (20%) makes this a strongly coupled flow and transport problem. The TOUGH2/T2DM simulation discussed here was conducted by Curt Oldenburg (see Oldenburg and Pruess [1994]). Parameters for the problem are presented in Table 17. The domain, boundary conditions, and discretization for the problem are presented in Figure 42. Because the T2DM module has not yet been released to the public, the dataset for this Test Problem is not provided.

The calculation domain consists of one-half of the symmetric physical domain. Results for the flow field and brine isopleths are shown in Figure 43. The full flow field has four cells with a central upwelling. These high-resolution results agree with the photograph of the experiment presented in Elder [1967]. In particular, Elder's experiment showed upflow near the center just as our results do. Calculations with coarser discretizations produce flow fields with downflow in the center. A complete discussion of these effects is presented in Oldenburg and Pruess [1994].

In the standard TOUGH2 methodology, the flow rate across the interface between two grid blocks depends only on the primary variables and properties of the two connected grid blocks. The flow term thus gives rise to two $NEQ \times NEQ$ submatrices of derivatives in the Jacobian matrix located in the off-diagonal locations corresponding to the two grid blocks involved. Dispersive fluxes depend on the vector quantities of Darcy velocity and species concentration gradient at each interface. In two-dimensional flow there are two vector components: the component perpendicular to the interface can be calculated directly from thermodynamic parameters of the two neighboring grid blocks; the component parallel to the interface must be interpolated from four direct components that involve thermodynamic parameters of six neighboring grid blocks.

T2DM generates for each interface six fully populated submatrices of order NEQ in the Jacobian matrix. The increased number of non-zero terms in the Jacobian matrix increases bandwidth, storage requirements and execution times of the linear equation solvers. Note that for the present problem with dispersivities of zero, many of the potentially non-zero terms in the Jacobian are in fact zero.

4.14.2. Solver performance

The performance of the various conjugate gradient solvers in T2CG1 on the RS/6000 370 and 320H workstations is presented in Table 18. On either machine DSLUCS is the fastest routine, followed by DSLUBC. DSLUGM requires substantially longer CPU times, and does not appear to be a good choice for this type of problem.

Symbol	Quantity	Value	Units
ϕ	Porosity	0.1	-
k	Permeability	4.845×10^{-13}	m^2
μ	Viscosity	1.0×10^{-3}	$Pa\ s$
g	Gravity	9.81	$m\ s^{-2}$
α_T	Transverse dispersivity	0.0	m
α_L	Longitudinal dispersivity	0.0	m
d	Molecular diffusivity	3.565×10^{-6}	$m^2\ s^{-1}$
τ	Tortuosity	1.0	-
ρ_0	Density of pure water	1000.	$kg\ m^{-3}$
ρ_b	Density of pure brine	1200.	$kg\ m^{-3}$

Number of Equations: 5,760			Computing Platforms: IBM RS/6000, Models 370 and 320H		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (*) (sec)
MA28	Could not solve the problem due to insufficient memory				
DSLUBC	30	198	317	63	1,993/4,880
DSLUCS	34	227	385	44	1,806/4,347
DSLUGM	34	235	390	304	2,780/6,768

(*): The first number indicates the 370 Model, the second the 320H model.

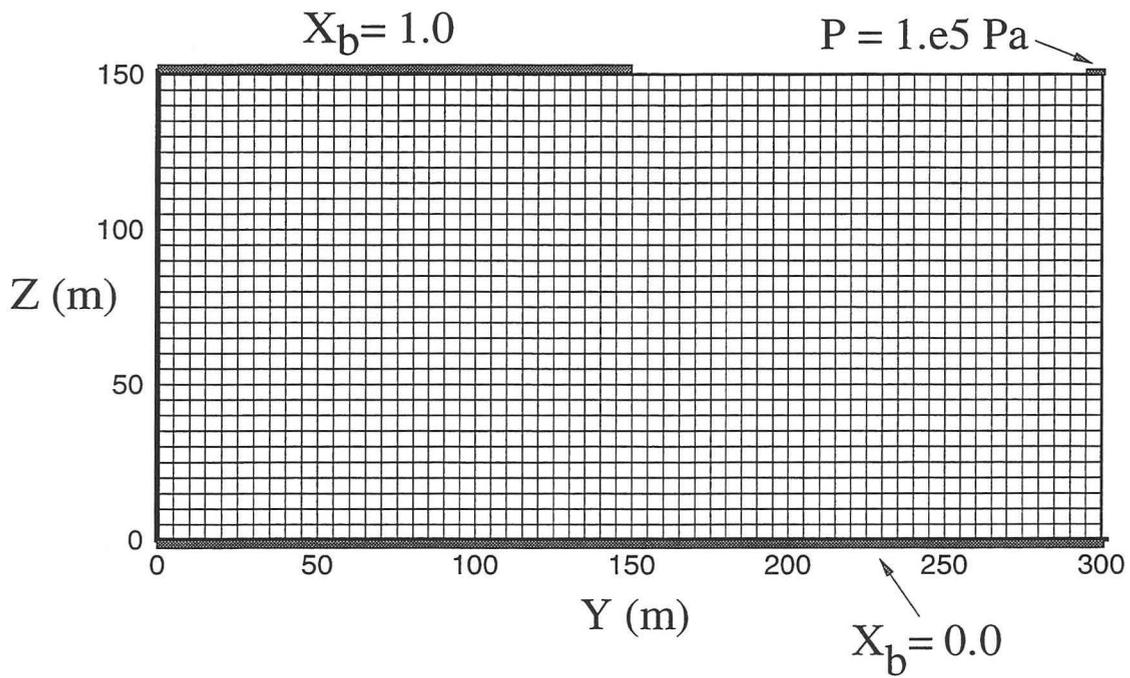


Figure 42. Discretization (uniform $N_Y \times N_Z = 60 \times 32$ grid blocks) for the Elder problem which takes advantage of the symmetry of the problem. The top and bottom rows of grid blocks are too narrow to appear in the figure.

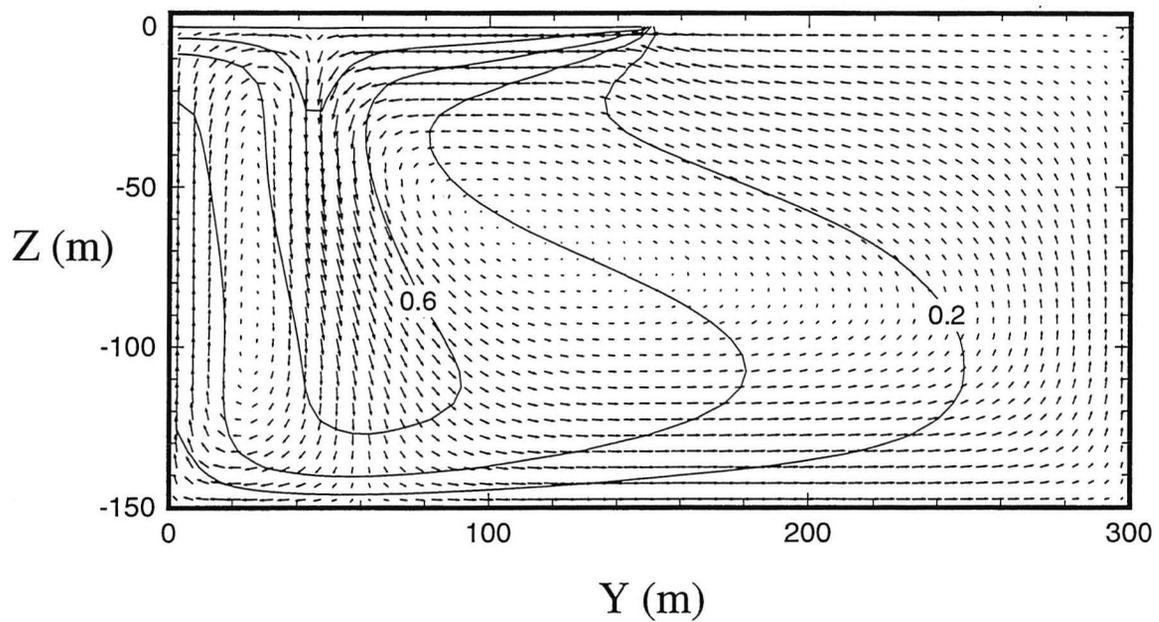


Figure 43. Results for the Elder free-convection problem at $t = 20$ years. Contours are labeled with mass fraction of maximum brine concentration ($X_{b,max} = 1.0$). Flow field and concentration isopleths show absence of central down-welling and multiple convection cells in agreement with the experimental results of *Elder* [1967].

4.15. Test Problem 15: The Buckley-Leverett Problem (1-D, Three-Phase with Oil and Water, Isothermal and Non-Isothermal)

4.15.1 Problem description

This problem consists of water injection into a porous medium filled with NAPL and water. The problem solved here is similar to the one discussed by *Falta et al.* [1995] in the user's manual of the T2VOC code, but has a finer spatial discretization. The initial water saturation S_w in the matrix is equal to the irreducible $S_{wi} = 0.159$. As water is injected into the one-dimensional system at its left boundary, NAPL is displaced and a shock front is developed at the leading edge of the invading water. The Buckley-Leverett problem assumes zero capillary pressures, no inter-phase mass transfer (i.e. zero NAPL solubility), and an isothermal process.

Details of the problem specifications are given by *Faust* [1985], and the T2VOC User's Manual [*Falta et al.*, 1995]. Two sets of simulations were run. The first was conducted on an Apple Macintosh QUADRA 800 microcomputer, and the second on the IBM RS/6000 370 workstation. The reservoir is 304.8 m long, and in the first data set it is divided in 254 equally-sized gridblocks with $\Delta x = 1.2$ m each. An additional inactive gridblock (i.e. with a zero volume) serves as the right-hand boundary. Water is injected at a rate of 1.506×10^{-4} kg/sec into the first gridblock (i.e. the leftmost in the domain), and the same amount leaves the domain through the right-hand boundary. The problem is solved as a non-isothermal system with $NEQ = 4$ and $NK = 3$, resulting in a total of $N = 1016$ equations. Of course, because injection is made isothermally temperatures will not change, but the Jacobian matrix is very different in non-isothermal mode, being less sparse and containing large cross-derivatives between mass- and energy-balance equations. We discuss this problem to demonstrate the dramatically different performance of the various solvers.

The second set of simulations assumed isothermal conditions and the domain was subdivided into a coarser grid, i.e. into 40 equally-sized gridblocks with $\Delta x = 7.6$ m each. Boundary and initial conditions were the same as in the QUADRA run. The problem was solved isothermally (i.e. $NEQ = 3$ and $NK = 3$), and resulted in a total of 126 equations. Both data files can be found on the accompanying diskette under the names "PROB15a" and "PROB15b". The solutions to the Buckley-Leverett problem appear in Figure 44.

4.15.2 Solver performance

Neither DSLUBC nor DSLUCS managed to solve the deceptively simple Buckley-Leverett problem when the simulation was run non-isothermally. This is traced to the fact that the resulting matrices exhibit characteristics far removed from those of symmetric positive definite matrices, under which conditions the CG solutions diverge. DSLUGM was the only one among the CG solvers that was successful in the solution of this problem, requiring only 3 iterations for the solution of the matrix (with the exception of the first timestep when 26 iterations were needed). DSLUGM was slower than the direct solver, but this is attributed to the small size of the problems and the related set-up overheads rather than to a systemic property of DSLUGM.

When the same problem is run isothermally, then no such difficulty is encountered. All three CG solvers in the T2CG1 package are faster than the MA28 direct solver, require the same time to solve the problem, and need only one iteration for all timesteps except the first.

Table 19. Solver Performance in Test Problem 15					
Number of Equations: 1016 (NEQ = 4)			Computing Platform: Apple Macintosh QUADRA 800		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	150	455	-	-	1,208
DSLUBC	Did not converge				
DSLUCS	Did not converge				
DSLUGM	150	455	26	3	1,364
Number of Equations: 126 (NEQ = 3)			Computing Platform: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	159	475	-	-	15
DSLUBC	159	475	21	1	12
DSLUCS	159	475	21	1	12
DSLUGM	159	475	19	1	12

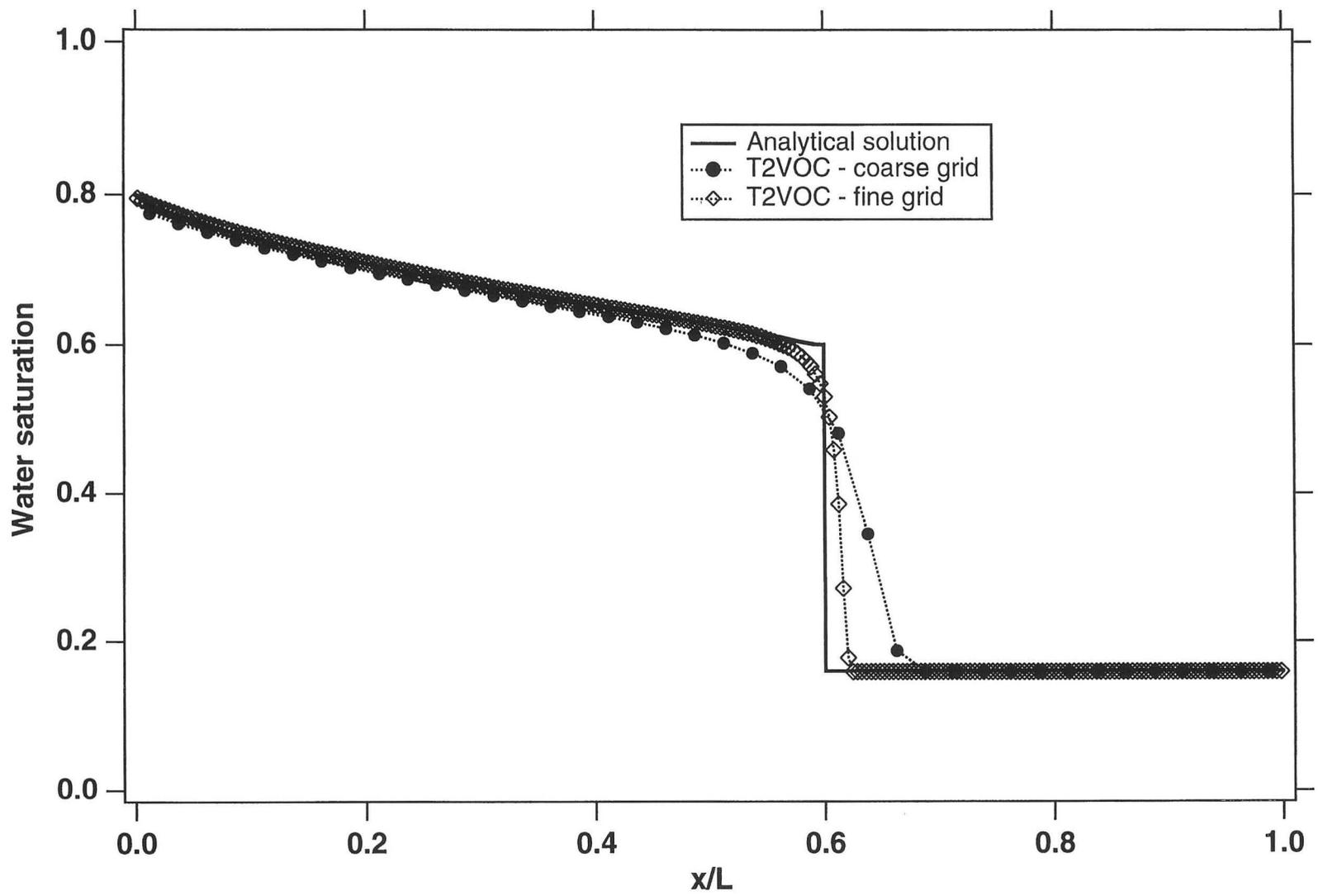


Figure 44. Analytical solution and T2VOC results for the Buckley-Leverett problem in Test Problem 15.

4.16. Test Problem 16: Air Barrier System for Waste Containment in the Vadose Zone (2-D, Three-Phase, Non-Isothermal with Phase Changes)

4.16.1 Problem description

In this problem we evaluate the performance of air barriers in the subsurface. The increase of air saturation in a soil alters significantly its hydraulic characteristics and reduces the relative permeability of liquids. This realization led to the concept that air injection could be used in the context of remedial strategies to create low permeability barriers to contaminated water and NAPL migration. Air offers a number of significant advantages as a barrier fluid: it is not a contaminant, already exists in the vadose zone, is abundant, easily available, free of charge, and has well-known thermodynamic properties.

The design discussed here involves a single layer of horizontal wells below the contamination in the vadose zone. If the wells are numbered, the odd-numbered wells are used for air injection, and the even-numbered ones for gas removal. This system has a predominantly horizontal orientation of flow. This horizontal barrier is characterized by (1) a drier zone near the injection wells, due to water displacement and evaporation, (2) a wetter zone near the removal wells due to recondensation, (3) a high gas pressure zone in the vicinity of injection, and (4) low gas pressure near the removal wells. In the vicinity of the injection wells, contaminated water and NAPLs cannot move downward because of a reduced NAPL relative permeability (caused by high gas saturation) and a pressure barrier to downward flow. In the vicinity of the removal wells, NAPLs and NAPL vapors cannot move downward due to a hydraulic gradient towards the wellbore.

In the vadose zone air injection not only displaces water, but also causes phase changes due to its drying effect. In the vicinity of the injection point, water and volatile NAPLs vaporize. The combination of displacement and drying drastically reduces the liquid relative permeability and creates a dry zone of high capillary suction from which liquid contaminants cannot escape. These effects may be significantly enhanced by increasing the temperature of the injected air. The vaporized water and NAPLs recondense away from the injection point, but their migration is controlled by the presence of the dry zone and the hydraulic impedance to flow between the injection and removal wells. In addition to containment, air barriers may also have a significant element of remediation because the vaporized NAPLs are removed from the subsurface through the gas removal well. Unlike other techniques, air barriers do not introduce a new liquid which can mobilize contaminants, and are easy-to-maintain.

The problem under study involves numerous and extremely non-linear complex processes: multi-phase (aqueous, gas, NAPL) multi-component (water, air, organic compound) flow, dissolution and advective transport, binary diffusion, phase changes (evaporation and condensation) and heat transport. T2VOC [Falta *et al.*, 1995], a member of the TOUGH2 family of codes, was used for the simulation.

We used a typical Hanford soil and TCE as the contaminant. The soil was considered homogeneous and isotropic with a permeability $k = 1.6 \times 10^{-11} \text{ m}^2$ and porosity $\phi = 0.385$. Relative permeability and capillary pressures were given by the Parker *et al.* [1987] 3-phase relationships, with $S_m = 0.13$, $n = 1.53$, $\alpha_{gn} = 13.15$, and $\alpha_{nw} = 15.47$ [Falta *et al.*, 1995]. The watertable was located 45 m from the surface. We modeled a section of the vadoze zone 40 m deep (from the surface) and 2 m wide.

The two-dimensional domain was discretized in 1720 gridblocks (20×86 in x,z). The simulation (NEQ = 4, NK = 3, with 6880 equations) included a continuous leakage of

TCE at a rate $q_0 = 1.3 \text{ kg/day}$ at $z = 18.3 \text{ m}$ from the top and $x = 1 \text{ m}$, with two wells located at $(x = 0 \text{ m}, z = 20 \text{ m})$ and $(x = 2 \text{ m}, z = 20 \text{ m})$. Air was injected into the first well at a rate of $q_a = 1000 \text{ kg/day}$ and an enthalpy of $H_a = 101 \text{ kJ/kg}$, and gas was withdrawn from the second well at a rate of $q_g = 1000 \text{ kg/day}$. The simulation time for both runs was $t_{max} = 720$ days. The complete dataset can be found on the accompanying diskette under the name "PROB16". Figure 45 shows the development of the air barrier, indicated by the gas saturation distribution in the domain.

4.16.2 Solver performance

The solver performance is summarized in Table 20. The large memory requirements essentially preclude the use of the MA28 direct solver in this problem, making the CG routines the only option for a problem of this size. The DSLUBC routine performs very well, and solves the problem in a maximum of $N_{CG} = 264$ iterations. This corresponds to a maximum $N_{CG}/N = 2.84\%$, indicating a very efficient solution.

The DSLUCS routine works well, but requires a longer execution time than DSLUBC despite fewer time steps and Newtonian iterations. This is attributed to the larger number of iterations needed to convergence in DSLUCS (which on a few occasions reach the maximum allowable number of 689), which also yield slightly more accurate solutions. The DSLUGM routine generally fails to converge within the prescribed maximum number of iterations, and manages to converge only for small timesteps (generally smaller than 3,600 sec). This would tremendously increase the execution time for the desired simulation period. Therefore, the DSLUGM simulation of Test Problem 16 was stopped after 50 timesteps.

Figure 46 shows the evolution of the residuals of the CG routines with the number of CG iterations in the first Newtonian iteration of the fifth timestep. The superiority of the DSLUBC solver and the unsuitability of the DSLUGM routine for the solution of this problem is evident from these figures. DSLUGM is converging at such a low rate that is in essence stagnating. The less effective performance of DSLUCS is demonstrated by the significantly larger magnitude of oscillations.

Table 20. Solver Performance in Test Problem 16					
Number of Equations: 6,880			Computing Platforms: IBM RS/6000 370		
SOLVER	Number of Δt 's	Newtonian Iterations	Maximum N_{CG}	Minimum N_{CG}	CPU Time (sec)
MA28	Could not solve the problem				
DSLUBC	515	3637	264	76	64,252
DSLUCS	510	3580	689	80	75,889
DSLUGM	The simulation was discontinued due to very small Δt .				

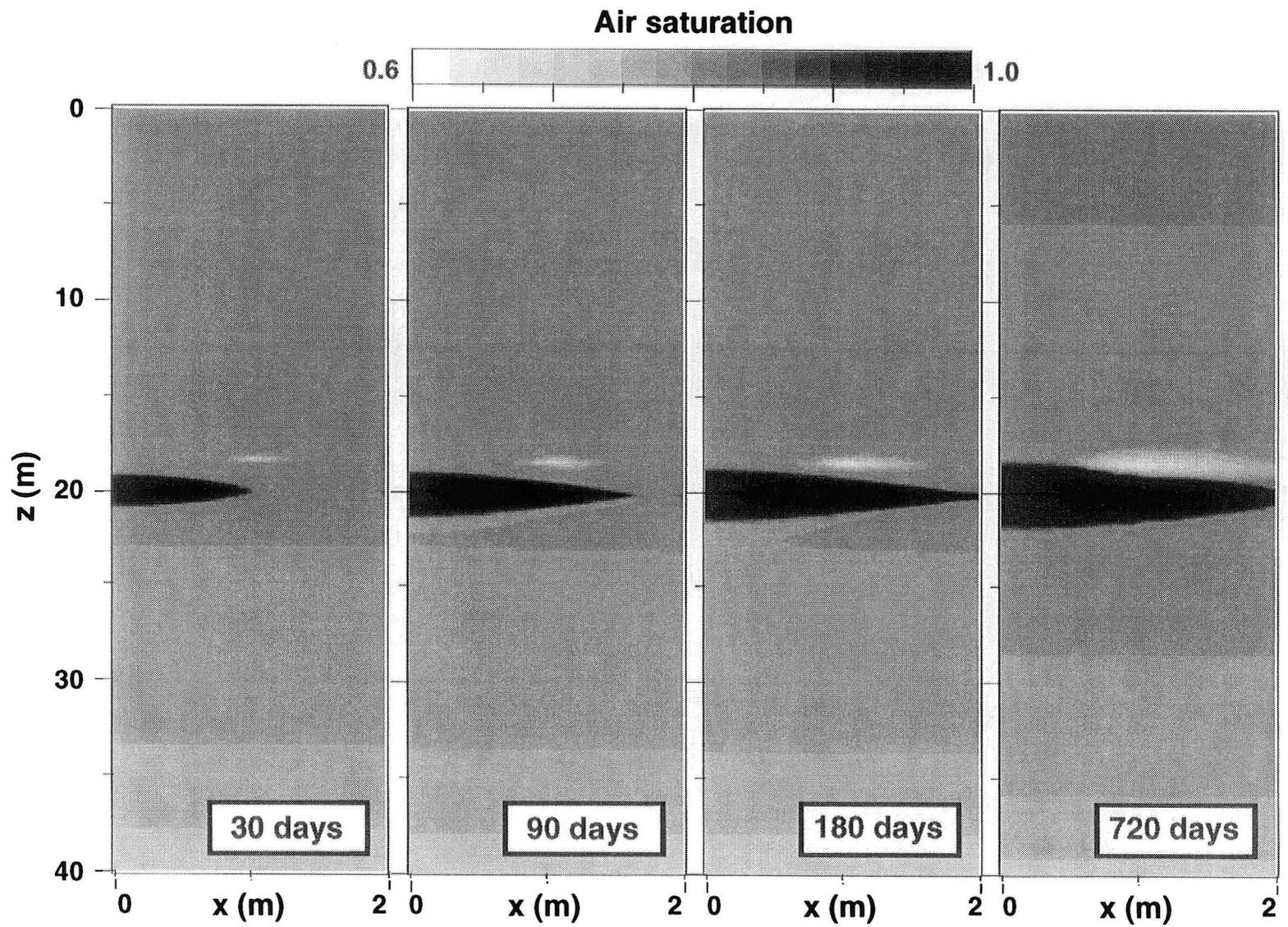


Figure 45. The evolution of the dry air barrier in Test Problem 16.

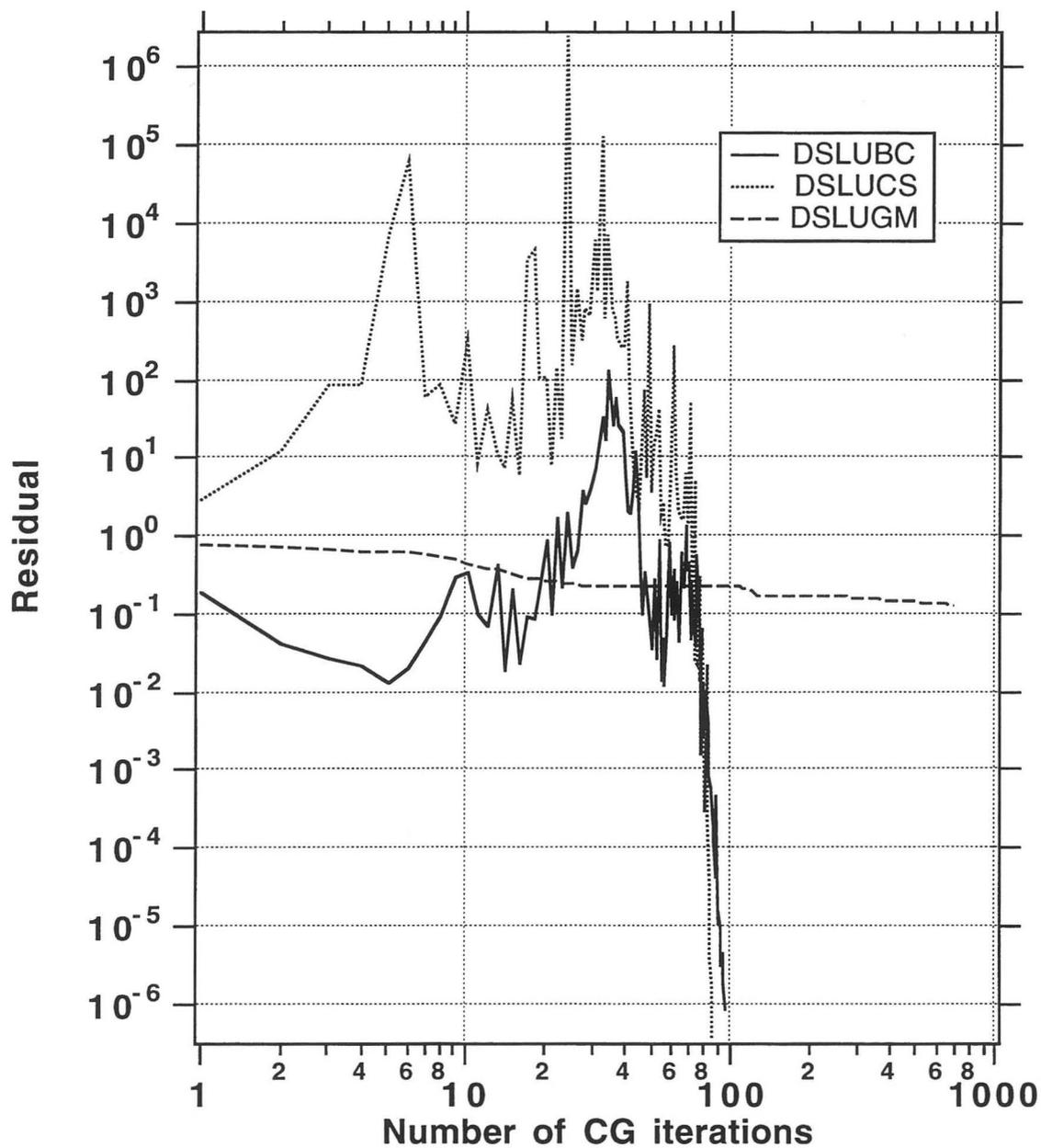


Figure 46. Performance of the CG solvers in the first Newtonian iteration of the 5th timestep in Test Problem 16 (1720 gridblocks, $N = 6,880$ equations).

5. CONCLUSIONS AND DISCUSSION

5.1. Summary and Conclusions

We developed T2CG1, a package of preconditioned conjugate gradient solvers for implementation in the TOUGH2 family of codes. T2CG1 considerably enhances the capabilities of the TOUGH2 family by significantly increasing the execution speed and the size of tractable problems. With the addition of T2CG1 two and three-dimensional flow problems with of the order of ten thousand gridblocks can be run on PCs, microcomputers and workstations. This compares with problem size limits of less than 2000 grid blocks (for 2-D) when using the MA28 direct solver, and less than 400 grid blocks for 3-D problems.

There are three preconditioned CG routines in T2CG1. The first is DSLUBC, and is based on the Bi-Conjugate Gradient (BCG) method. The second routine is DSLUCS, and is based on the Conjugate Gradient Squared (CGS) method. DSLUGM, the third solver, is based on the Generalized Minimum Residual (GMR) method.

In this report we tested the CG solvers in T2CG1 in 16 test problems. These problems covered a wide range of problems of fluid and heat flow and transport through porous media. They ranged from simple one-dimensional single-phase flow to large non-isothermal three-dimensional, three-phase, multi-component systems with phase changes and interphase mass transfer. The size of the problems (indicated by the order of the matrix, N) varied from $N = 126$ to $N = 20,000$. The simulations were performed on a variety of computing platforms: two IBM RS/6000 workstations (models 370 and 320H) using double-precision (64-bit) arithmetic, a Macintosh Quadra 800 microcomputer equipped with a 68040 processor, and a IBM-type PC powered by a 486 DX2-66 processor. We compared the CPU times for the various routines against each other, and against the direct solver MA28 when possible. Based on these results, the following conclusion are reached:

- (a) The CG solvers were invariably faster (and often several times so) than the MA28 direct solver in all 16 problems. Even in very small problems of less than 300 equations one or more of the CG routines were faster than the direct solver.
- (b) MA28 could not solve most of the medium and large problems. This was mostly due to its very large and not well-defined memory requirements (especially in three-dimensional grids), but also to a slow underlying algorithm which could not provide a solution even when the allocated memory was adequate.
- (c) The performance of the CG solvers was problem-, compiler-, and machine-specific.
- (d) Conversely, the CG solvers had well-defined memory requirements, which were significantly lower than the ones for MA28. These low memory requirements and the speed of the routines made possible the solution of 20,000-equation problems on desktop personal computers.
- (e) Although the performance of the three CG solvers varied in the solution of the 16 test problems and occasionally a specific solver failed, we did not encounter any cases where all three CG routines failed. In all test problems one or more CG solvers produced a fast, efficient, and accurate solution.

- (f) In terms of speed, DSLUCS seems to have a clear advantage in most cases. However, this superiority is not unconditional, and is only observed in well-behaved matrices which have structures close to that of symmetric positive definite matrices. In less well-behaved systems it may be less successful, and in the suite of problems we tested we encountered two cases where the DSLUCS solver did not converge.
- (g) DSLUBC seems to be the most robust, as it managed to solve all the test problems. However, it could be considerably slower than DSLUCS.
- (h) DSLUGM was at times faster than DSLUCS, and the only solver which managed to solve the non-isothermal Buckley-Leverett problem (Test Problem 15). On the average DSLUGM was slower than DSLUCS.

In order to evaluate the CG solvers in T2CG1, we investigated the relationship between the CPU time per Newtonian iteration (i.e. per matrix solution) and the order of the matrix, N . In this analysis we used data from the IBM RS/6000 workstation models 370, and sought to determine a relationship of the form

$$t = C \cdot N^m \quad (25)$$

where t is the CPU time per matrix solution, C a constant, N is the order of the matrix, and m an exponent. The exponent m is of importance, as it controls the increase in execution times as the size of the problem grows. In direct solvers, $m = 3$. The smaller the m value, the smaller the execution times and the more competitive the solver. It must be clearly stated that this is a rather crude measure since it is averaging and does not account for the case-specificity in the performance of the CG solvers, which are affected by the dimensionality of the problem, the matrix fill-up, the relative and absolute magnitude of the matrix elements, and the ability of the solver to converge. However, despite its conceptual weaknesses, Equation (25) can provide a basic measure for the evaluation of the relative speed of the solvers.

Figures 47 through 49 show the CPU time (per matrix solution) vs. N for the three CG solvers in T2CG1. The DSLUCS solver has the lowest m ($= 1.36$), confirming our earlier statement about the generally superior speed of this solver. DSLUCS appears to be significantly faster than DSLUBC ($m = 1.50$), which seems to be the second best in terms of speed. However, this statement cannot be made with certainty because the m value of DSLUGM ($=1.57$) is very close to that of DSLUBC.

5.2. Choice of Solver

Selecting the best method for a given class of problems is more an art than a science, and is largely a matter of trial and error. It is also a matter of available storage, as DSLUGM requires four times the memory of DSLUCS and DSLUBC. In this subsection we will attempt to offer some general rules which may prove helpful in the selection of the most efficient solver.

In attempting to solve a large problem, the user is advised to first study the relative performance of the solvers on a smaller problem belonging to the same class. The most obvious way to accomplish this is to use the same properties, but a coarser grid, and run the simulation for a limited number of timesteps. In light of its relative speed, DSLUCS is

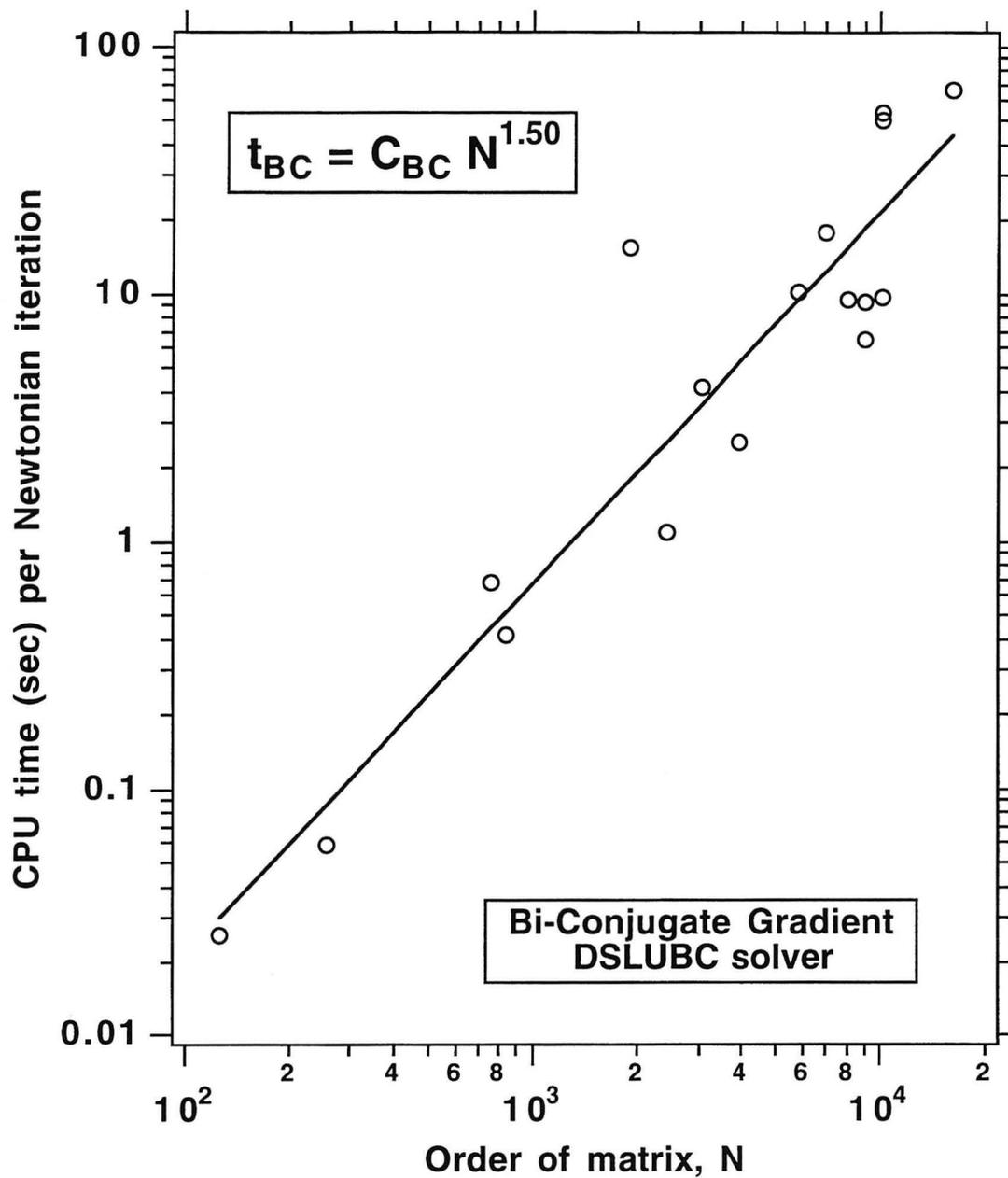


Figure 47. Relationship between the CPU time per Newtonian iteration and the order of the matrix, N , in the DSLUBC solver ($C_{BC} = 2.09 \times 10^{-5}$).

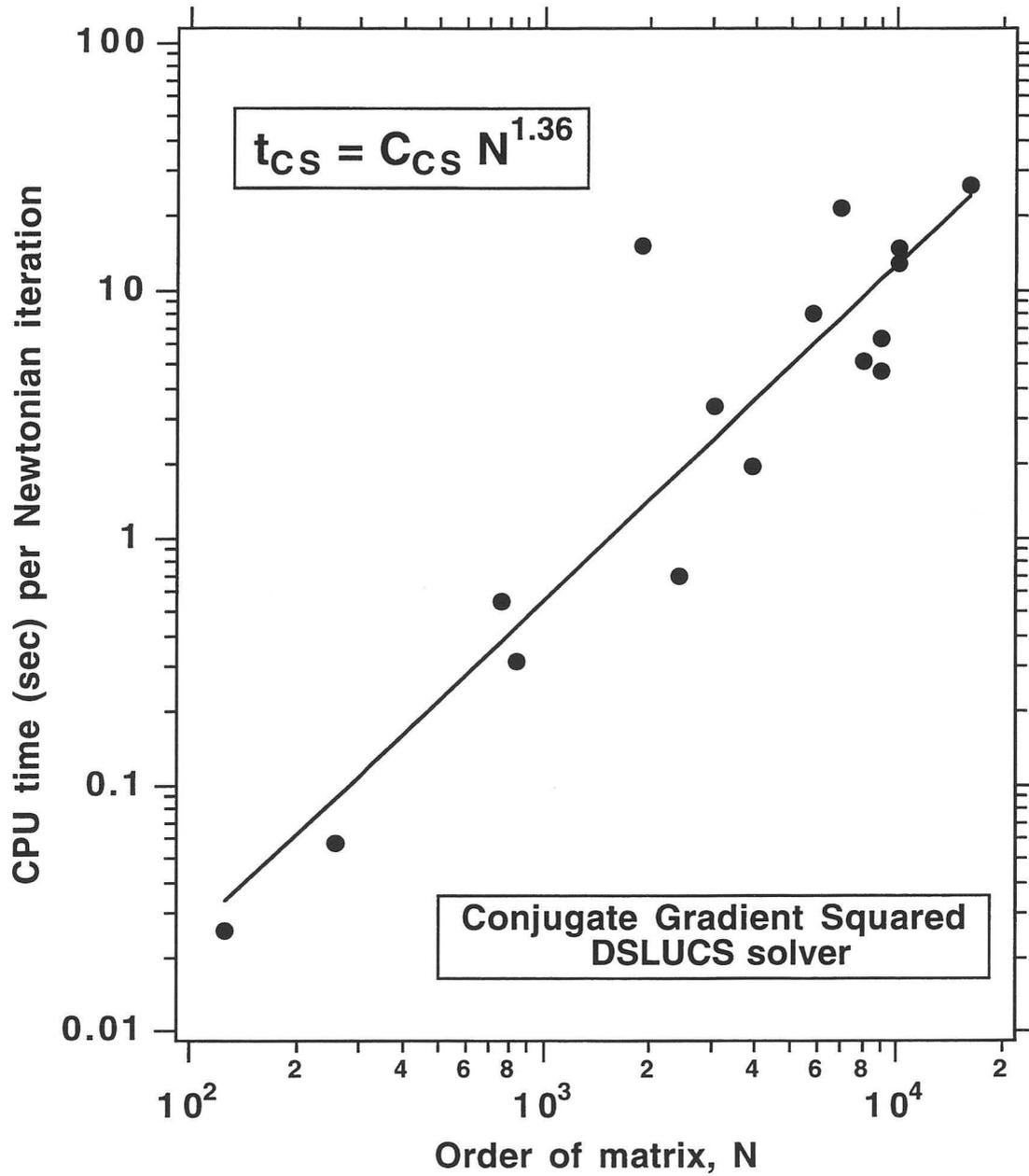


Figure 48. Relationship between the CPU time per Newtonian iteration and the order of the matrix, N , in the DSLUCS solver ($C_{CS} = 4.75 \times 10^{-5}$).

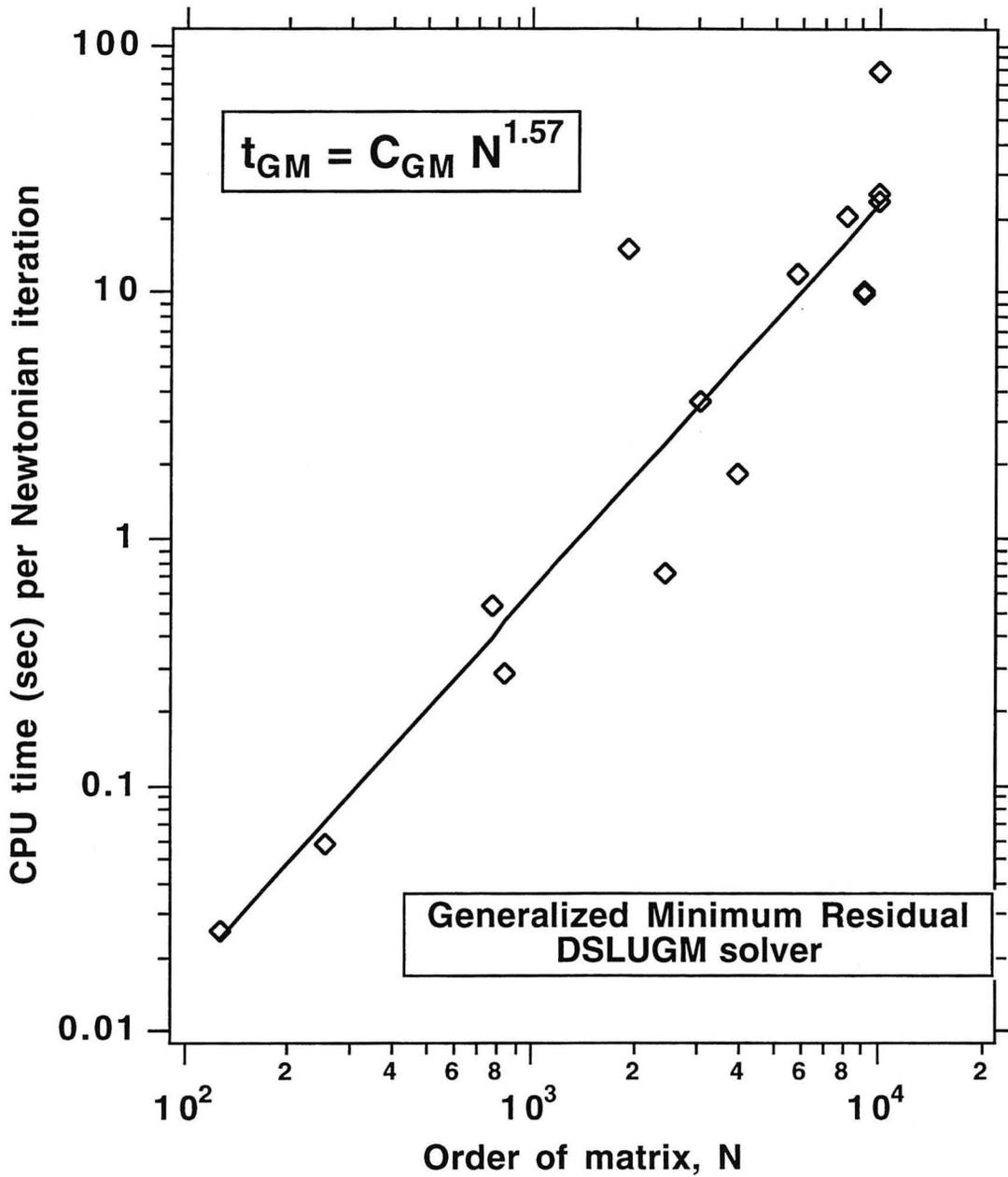


Figure 49. Relationship between the CPU time per Newtonian iteration and the order of the matrix, N , in the DSLUGM solver ($C_{GM} = 1.19 \times 10^{-5}$).

a sound first choice. If DSLUCS fails to deliver a satisfactory performance, DSLUBC should be the next choice. If DSLUBC fails, DSLUGM should be tried.

The user is strongly encouraged to use all three solvers in the small problem before selecting the most successful for application to the larger problem. It must be clearly pointed out that this process is designed to determine which method or methods fail rather than which one is the most appropriate. Although the performance of the solvers may change significantly as the problem size increases, it is highly unlikely (if not impossible) that a solver will be able to solve the larger problem if it has failed to solve the smaller one.

The CG solvers which manage to solve the smaller problem should then be used in the solution of the larger problems for a limited number of timesteps. The user is advised to ensure that substantial changes occur within these timesteps, i.e. the timestep should be sizeable. The fastest routine is then selected, and applied to the large problem for the desired simulation period.

5.3. Important Issues and Precautions

In applying the CG solvers, the user must be aware of a few fundamental facts concerning these (as well as any other) iterative solvers. In using such solvers, the user trades the robustness and slow performance of a direct solver (which will yield the correct solution) for the speed and reduced memory requirements of an iterative solver. Iterative solutions are approximate, but may be made to be so close to the correct solution that they are practically identical. They are case-, compiler-, and machine-specific, and may be less reliable if the solvers are not carefully applied and the results not carefully interpreted.

Because of the case-specificity and the unpredictability of the CG solvers, the user must watch carefully both the convergence and the evolution of residuals over the time increments of the simulation. Close examination of the CG performance statistics in the file LINEQ generated by the TOUGH2 runs becomes a necessity rather than an option. LINEQ records vital CG information at the end of each Newtonian iteration (i.e. each matrix solution).

The user must pay close attention to the value of the parameters IERR and ERR. An IERR value of 2 indicates that the CG routine did not converge after the prescribed maximum number of allowable iterations (see Subsection 3.2). Occasional occurrences of IERR = 2 should not be cause for alarm because the deviations from the actual solutions are lumped and carried into the next Newtonian iteration where it may be resolved. However, the occurrence of IERR = 2 in successive Newtonian iterations is a clear indication of a potential problem. Inaccurate solution of the linear equations may slow down the convergence of the Newtonian (outer) iterations. This may lead to an increased number of Newtonian iterations, or convergence failure within the maximum prescribed number (NOITE, default = 8), and subsequent time step reduction. Inaccurate linear algebra cannot, however, cause Newtonian iterations to converge to a wrong solution.

The simplest and easiest way to address convergence problems is to switch to a different CG solver if IERR = 2 is observed frequently. In extreme cases (usually involving extreme heterogeneity, phase changes, and/or radial geometry) it might be necessary to tighten the CG convergence criterion by changing the source code (subroutine LINEQ, see APPENDIX A). It must be emphasized, however, that this must be an action of last resort and not a routine change.

There are circumstances under which an accurate solution may be obtained despite frequent occurrences of $IERR = 2$. This is possible in cases where the CG routines generally fail to converge, but give results that are "good enough" to permit the Newton-Raphson process to proceed to convergence. In our experience, if the norm ratios (equations 19 and 20) are up to 2 orders of magnitude larger than CLOSUR (i.e. 10^{-4} when the standard value of $CLOSUR = 10^{-6}$ is used), convergence to the true solution is possible. This process (not uncommon in the application of CG solvers) in essence transfers error from the CG (inner) iterations to the Newtonian (outer) iteration component of the solution, and does not pose any problems unless the error is significant. This may result in more CG iterations, more Newtonian iterations (and consequently shorter timesteps), and longer execution times, but does not affect the accuracy of the solution.

The issue of the CG solution behavior as the steady state is approached must also be discussed. A quite common feature of several CG algorithms is the irregular convergence behavior in situations where the iterations are started close to the solution (e.g. in initializations and when approaching steady state). This is a weakness which may lead to severe residual cancellation and errors in the solution. The underlying algorithms of the CG solvers in T2CG1 may exhibit this weakness, which stems from the fact that under the aforementioned conditions the signal (excitation) of the system is very weak. In this case the system response as steady state approaches is extremely small (practically zero) as conditions do not change over time, and it is entirely possible to be overwhelmed by the very small roundoff error (at the machine accuracy level) arising from the matrix-vector multiplications in the CG operations.

Steady state is reached in TOUGH2 simulations when timesteps are large and increasing, and in 10 successive timesteps convergence is attained in a single Newtonian (outer) iteration. When such a behavior is observed, the TOUGH2 internal logic concludes that steady state has been reached and terminates the simulation. Most, but not all, T2CG1 simulations exhibit such normal behavior. The erratic behavior of the CG solvers is characterized by an abnormal jump in the change of the primary variables from the extremely small and often zero values in the previous timesteps, which results in more Newtonian iterations and often timestep reductions. After this first occurrence of trouble, solutions may diverge and may give the appearance of instability.

The key in addressing this potential problem is stopping the essentially converged simulation of the steady state before the first appearance of trouble. The simplest solution is to relax the criterion which requires 10 successive timesteps of convergence in a single Newtonian (outer) iteration before a simulation is stopped. A reduction from 10 to a lower number (e.g. 5) could accomplish this. Such an approach, however, is entirely mechanistic and needs careful monitoring of the changes in the primary variables and the timestep size during implementation. A more elegant solution combines these two considerations by stopping the simulations when the ratio of the maximum change in the primary variables to the timestep size (in seconds) is equal or less than an appropriately low number. In our experience a number of 10^{-14} - 10^{-12} gives satisfactory results. Such a change has not been introduced into TOUGH2, and its implementation is in the user's discretion.

6. ACKNOWLEDGMENTS

Thanks are due to Steve Webb of Sandia National Laboratories, Albuquerque, NM, and to Emilio Antúnez and Curt Oldenburg of LBL for making available to us TOUGH2 input files 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. C. Oldenburg and S. Finsterle are thanked for their helpful review comments.

7. REFERENCES

- Adenekan, A.E., Numerical modeling of multiphase transport of multicomponent organic contaminants and heat in the subsurface, Ph.D. Dissertation, Univ. of Calif., Berkeley, 1992.
- Adenekan, A.E., T.W. Patzek, and K. Pruess, Modeling of multiphase transport of multicomponent organic contaminants and heat in the subsurface: Numerical Model Formulation, *Water Resour. Res.*, 29(11), 3727-3740, 1993.
- Antúnez, E. and M. Lippmann, Numerical study of the effects of brine injection on the CP1 production area of Cerro Prieto, Earth Sciences Division Annual Report 1992, *Report LBL-33000*, Lawrence Berkeley Laboratory, Berkeley, California, 1993.
- Antúnez, E. and K. Pruess, The use of the turning bands method (TBM) to generate stochastic aperture distributions in heterogeneous structures, *Report LBL-35581*, Lawrence Berkeley Laboratory, Berkeley, California, 1994.
- Antúnez, A., G. Moridis and K. Pruess, Large-scale geothermal reservoir simulation on PCs", *Report LBL-35192*, Lawrence Berkeley Lab., Berkeley, Calif., 1994.
- Avdonin, N.A., Some formulas for calculating the temperature field of a stratum subject to thermal injection, *Neft'i Gaz*, 3, 37-41, 1964.
- Axelsson, O., *Solution of Linear Systems of Equations: Iterative Methods*, Lecture Notes in Mathematics 572, Springer-Verlag, Berlin, Heidelberg, New York, 1977.
- Brooks, R.H. and A.T. Corey, Properties of porous media affecting fluid flow, *J. Irrig. Drain. Div. of the ASCE*, 92(IR2), 61-88, 1966.
- Doughty, C. and K. Pruess, A similarity solution for two-phase fluid and heat flow near high-level nuclear waste packages emplaced in porous media, *International Journal of Heat and Mass Transfer*, 33(6), 1205-1222, 1990.
- Doughty, C. and K. Pruess, A similarity solution for two-phase water, air and heat flow near a linear heat source in a porous medium, *Journal of Geophysical Research*, 97(B2), 1821-1838, 1992.
- Duff, I. S., MA28 - A set of FORTRAN subroutines for sparse unsymmetric linear equations, *AERE Harwell Report R 8730*, 1977.
- Elder, J.W., Transient Convection in a Porous Medium, *J. Fluid Mech.*, 27(3), 609-623, 1967.
- Falta, R.W., K. Pruess, I. Javandel and P.A. Witherspoon, Numerical modeling of steam injection for the removal of nonaqueous phase liquids from the subsurface. 1. Numerical formulation, *Water Resour. Res.*, 28(2), 433-449, 1992.
- Falta, R.W., K. Pruess, I. Javandel and P.A. Witherspoon, Numerical modeling of steam injection for the removal of nonaqueous phase liquids from the subsurface. 2. Code validation and application, *Water Resour. Res.*, 28(2), 451-465, 1992.
- Falta, R.W., K. Pruess, S. Finsterle, and A. Battistelli, T2VOC User's Guide, *Report LBL-36400*, Lawrence Berkeley Laboratory, Berkeley, California, March 1995.

- Faust, C.R., Transport of immiscible fluids within and below the unsaturated zone: A numerical study, *Water Resour. Res.*, 21(4), 587-596, 1985.
- Fletcher, R., *Conjugate Gradient Methods for Indefinite Systems, Numerical Analysis*, Lecture Notes in Mathematics 506, Springer-Verlag, New York, 1976.
- Golub, G.H. and C.F. vanLoan, *Matrix Computations*, 2nd edition, Johns Hopkins University Press, Baltimore, MD, 1989.
- Hestenes, M.R. and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards*, 49(6), 409 - 436, 1952.
- Hiriart, G.L. and H.P. Gutiérrez, An update of Cerro Prieto geothermal field, Twenty years of commercial power, *Geothermal Resources Council Bull.*, 289-294, Sept/Oct 1992.
- Leverett, M.C., Capillary behavior in porous solids, *Trans. Soc. Pet. Eng. AIME*, 142,152-169, 1941.
- Mantoglou, A. and J.L. Wilson, The turning bands method for simulation of random fields using line generation by a spectral method, *Water Resour. Res.*, 18(5), 1379-1394, 1982.
- Meyerink, J.A., 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, 1983.
- Moridis, G.J. and K. Pruess, TOUGH simulations of Updegraff's set of fluid and heat flow problem, *Report LBL-32611*, Lawrence Berkeley Laboratory, Berkeley, California, 1992.
- Oldenburg, C.M. and K. Pruess, A two-dimensional dispersion module for the TOUGH2 simulator, *Report LBL-32505*, Lawrence Berkeley Laboratory, Berkeley, California, 1993.
- Oldenburg, C.M. and K. Pruess, Numerical simulation of coupled flow and transport with TOUGH2: A verification study, *Report LBL-35273*, Lawrence Berkeley Laboratory, Berkeley, California, 1994.
- Oldenburg, C.M. and K. Pruess, Dispersive transport dynamics in a strongly coupled groundwater brine flow problem, *Water Resour. Res.*, 31(2), 289-302, 1995.
- O'Sullivan, M. J., A similarity method for geothermal well test analysis, *Water Resour. Res.*, 17(2), 390-398, 1981.
- Parker, J.C., R.J. Lenhard, and T. Kuppusamy, A parametric model for constitutive properties governing multiphase flow in porous media, *Water Resour. Res.*, 23(4), 618-624, 1987.
- Peaceman, D.W., and H.H. Rachford, The numerical solution of parabolic and elliptic differential equations, *J. SIAM*, 3(1), 28-41, 1955.

- 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, 1989.
- Pruess, K., Development of the general purpose simulator MULKOM, Annual Report 1982, *Report LBL-15500*, Lawrence Berkeley Laboratory, Berkeley, California, 1983.
- Pruess, K., TOUGH User's Guide, *Report LBL-20700*, Lawrence Berkeley Laboratory, Berkeley, California, 1987 (also reports *NUREG/CR-4645* and *SAND86-7104*).
- Pruess, K., SHAFT, MULKOM, TOUGH: A set of numerical simulators for multiphase fluid and heat flow, *Geothermia, Rev. Mex. Geoenergia*, 4(1) 185-202, 1988 (also Lawrence Berkeley Laboratory *Report LBL-24430*).
- Pruess, K. (editor), Proceedings of the TOUGH Workshop, *Report LBL-29710*, Lawrence Berkeley Laboratory, Berkeley, California, 1990.
- Pruess, K., TOUGH2 - A general-purpose numerical simulator for multiphase fluid and heat flow, *Report LBL-29400*, Lawrence Berkeley Laboratory, Berkeley, California, 1991.
- Pruess, K., On the validity of a Fickian diffusion model for the spreading of liquid infiltration plumes in partially saturated heterogeneous media, *Report LBL-35134*, Lawrence Berkeley Laboratory, Berkeley, California, 1994.
- Pruess, K. and Y. Tsang, Modeling of strongly heat-driven flow processes at a potential high-level nuclear waste repository at Yucca Mountain, Nevada, *Report LBL-33597*, Lawrence Berkeley Laboratory, Berkeley, California, 1993.
- Pruess, K. and Y. Tsang, Thermal modeling for a potential high-level nuclear waste repository at Yucca Mountain, Nevada, *Report LBL-35381*, Lawrence Berkeley Laboratory, Berkeley, California, 1994.
- Reda, D. C., Natural convection experiments about a finite-length cylindrical heat source in a liquid-saturated porous medium, *Report SAND83-2209C*, Sandia National Laboratories, Albuquerque, NM, 1984.
- Ross, B., J. W. Mercer, S. D. Thomas, and B. H. Lester, Benchmark problems for repository siting models, *NUREG/CR-3097*, Geotrans, Inc., Reston, VA, December 1982.
- Saad, Y. and M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 7(3), 856-869, 1986.
- Scheidegger, A.E., *The Physics of Flow Through Porous Media*, University of Toronto Press, Toronto, 1974.
- Seager, M.K., A SLAP for the Masses, *Report UCRL-100195*, Lawrence Livermore National Laboratory, Livermore, California, 1988.
- Sonneveld, P., CGS, A fast Lanczos-type solver for nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 10(1), 36-52, 1989.

- Stone, H.L., Iterative solution of implicit approximations of multi-dimensional partial differential equations, *SIAM J. Numer. Anal.*, 5(3), 530-558, 1968.
- Tompson, A.F.B., Implementation of the three-dimensional turning bands random field generator, *Water Resour. Res.*, 25(10), 2227-2243, 1989.
- Tsang, Y.W. and K. Pruess, Further modeling studies of gas movement and moisture migration at Yucca Mountain, Nevada, *Report LBL-29127*, Lawrence Berkeley Laboratory, Berkeley, California, 1990.
- van Genuchten, M.T., A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, *Soil Sci. Soc. Am. J.*, 44, 892-898, 1980.
- Varga, R.S., *Matrix Iterative Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1962.
- Vauclin, M., D. Khanji, and G. Vachaud, Experimental and numerical study of a transient, two-dimensional unsaturated-saturated water table recharge problem, *Water Resour. Res.*, 15(5), 1089-1101, 1979.
- Voss, C.I., and W.R. Souza, Variable density flow and solute transport simulations of regional aquifers containing a narrow freshwater-saltwater transition zone, *Water Res. Res.*, 23(10), 1851-1866, 1987.
- Webb, S.W. and J.M. Phelan, Prediction of single component NAPL behavior for the TEVES project using T2VOC, Proceedings of TOUGH Workshop '95, Berkeley, California, March 20-22, 1995.
- Webb, S.W., K.W. Larson, G.A. Freeze, and T.L. Christian-Frear, Summary of applications of TOUGH2 to the evaluation of the multiphase flow processes at the WIPP, Proceedings of TOUGH Workshop '95, Berkeley, California, March 20-22, 1995.
- Young, D.M., *Iterative Solution of Large Linear Systems*. Academic Press, New York, 1971.

APPENDIX A - SOURCE CODE CHANGES

Here we discuss how a user can change some of the *hardwired* parameters which control the convergence of the CG solvers. These parameters are the maximum number of permissible CG iterations, *NMAXIT*, and the CG convergence criterion, *CLOSUR*. It is important to reiterate the author's warning that caution should be exercised when introducing these changes. The part of the code where the changes are to be made is in subroutine LINEQ, and is located at the beginning of the executable portion of the routine. In its default form, this LINEQ part reads as follows:

```

SUBROUTINE LINEQ
.
.
.
      ICALL=ICALL+1
      IF(ICALL.EQ.1) then
        WRITE(11,899)
899      FORMAT(6X,'LINEQ      0.91 CG  31 January   1994',6X,
x        'Interface for linear equation solvers'/
x        47X,'can call MA28 or a package of conjugate gradient',
x        ' solvers')
c
      matslv=mop(21)
      if(matslv.eq.0.or.matslv.gt.4) matslv=3
      nmaxit=max(20,nel*neq/10)
      iclosr=2
      if(matslv.eq.4) iclosr=0
      closur=1.e-6
      isym=0
      iunit=0
      nvectr=30
      seed=1.e-25
      endif

```

If the user intends to change *NMAXIT*, then the first underlined line of code in the above code section is to be modified by changing the arguments in `max()`. The value of the first argument is not very important, and its maximum suggested value is 100. In the second argument (by far the most important) we increase the maximum number of allowable iterations by decreasing the denominator from its default value of 10 (i.e. $NMAXIT = 0.1N$). The user is strongly discouraged from using $NMAXIT > 0.25N$ (corresponding to a denominator equal to 4) for reasons discussed in Subsection 3.2. We suggest that *NMAXIT* never exceed the value given by

```
nmaxit=max(100,nel*neq/4)
```

The CG convergence criterion is changed by modifying the second underlined statement. If changed, we suggest that this line be replaced by

```
closur=max(closmn,1.0e-12)
```

where `closmn` is the new (other than $1.e-6$) CG convergence criterion. This formulation ensures that `closur` is never set at levels close to the machine accuracy where roundoff errors could create convergence problems.

The third underlined statement corresponds to a parameter which does not control convergence, but allows the creation of a detailed record of the evolution of the residuals at each CG iteration, Newtonian iteration, and timestep. To do this the statement is changed to

```
iunit=15
```

and the record of the residuals is printed in the output file `LINEQ`. While monitoring the evolution of the residuals is interesting and not too cumbersome in small problems and with a limited number of timesteps, it may lead to extremely large (several tens of thousand lines long) output files in larger problems covering long simulation periods. Unless there is a strong need for the user to track the performance of a specific solver and only for a limited number of timesteps, this change is strongly discouraged.

**APPENDIX B - MAILING LIST FOR YUCCA MOUNTAIN SITE
CHARACTERIZATION PROJECT***

D. A. Dreyfus (RW-1) [1]
Director
Office of Civilian Radioactive Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, DC 20585

C. A. Kouts (RW-36) [1]
Regulatory Integration Division
OCRWM
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, DC 20585

L. H. Barrett (RW-2) [1]
Acting Deputy Director
Office of Civilian Radioactive Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, DC 20585

S. Rousso, (RW-40) [1]
Office of Waste Acceptance, Storage and
Transportation
OCRWM
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, DC 20585

J. D. Saltzman (RW-10) [1]
Office of Human Resources and
Administration
OCRWM
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, DC 20585

Victoria F. Reich, Librarian [4]
Nuclear Waste Technical Review Board
1100 Wilson Blvd., Suite 910
Arlington, VA 22209

J. C. Bresee (RW-10) [1]
Office of Human Resources and
Administration
OCRWM
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, DC 20585

R. M. Nelson, Jr., Acting Project Manager [5]
Yucca Mountain Site Characterization Office
U.S. Department of Energy
P.O. Box 98608--MS 523
Las Vegas, NV 89193-8608

T. Wood (RW-14) [1]
Contract Management Division
OCRWM
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, DC 20585

Office of the Director [1]
Office of External Affairs
DOE Nevada Operations Office, Nevada
U.S. Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-8518

R. A. Milner (RW-30) [1]
Office of Program Management and
Integration
OCRWM
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, DC 20585

Technical Information Officer [8]
DOE Nevada Operations Office
U.S. Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-8518

* This mailing list is included in compliance with the requirements of the Yucca Mountain Project (YMP).

P. K. Fitzsimmons, Technical Advisor [1]
 Office of Assistant Manager for
 Environmental Safety and Health
 U.S. DOE/Nevada Operations Office
 P.O. Box 98518
 Las Vegas, NV 89193-8518

E. P. Binnall [1]
 Field Systems Group Leader
 Building 50B/4235
 Lawrence Berkeley Laboratory
 Berkeley, CA 94720

D. R. Elle, Director [1]
 Environmental Protection Division
 DOE Nevada Operations Office
 U.S. Department of Energy
 P.O. Box 98518
 Las Vegas, NV 89193-8518

Center for Nuclear Waste Regulatory Analyses [1]
 6220 Culebra Road
 Drawer 28510
 San Antonio, TX 78284

Repository Licensing & Quality Assurance [1]
 Project Directorate
 Division of Waste Management
 U.S. Nuclear Regulatory Commission
 Washington, DC 20555

W. L. Clarke [3]
 Technical Project Officer for YMP
 Attn: YMP/LRC
 Lawrence Livermore National Laboratory
 P.O. Box 5514
 Livermore, CA 94551

Senior Project Manager for Yucca Mountain [1]
 Repository Project Branch
 Division of Waste Management
 U.S. Nuclear Regulatory Commission
 Washington, DC 20555

J. A. Blink [1]
 Deputy Project Leader
 Lawrence Livermore National Laboratory
 101 Convention Center Drive, Suite 820, MS 527
 Las Vegas, NV 89109

NRC Document Control Desk [1]
 Division of Waste Management
 U.S. Nuclear Regulatory Commission
 Washington, DC 20555

J. A. Canepa [4]
 Technical Project Officer for YMP
 N-5, Mail Stop J521
 Los Alamos National Laboratory
 P.O. Box 1663
 Los Alamos, NM 87545

Philip S. Justus [1]
 NRC Site Representative
 301 E. Stewart Ave., Room 203
 Las Vegas, NV 89101

H. N. Kalia [1]
 Project Leader for ESF Test Coordination
 Los Alamos National Laboratory
 Mail Stop 527
 101 Convention Center Drive, Suite 820
 Las Vegas, NV 89101

N. Z. Elkins [1]
 Deputy Technical Project Officer
 Los Alamos National Laboratory
 Mail Stop 527
 101 Convention Center Drive, Suite 820
 Las Vegas, NV 89101

J. S. Stuckless [1]
 Chief, Geologic Studies Program--MS 425
 Yucca Mountain Project Branch
 U.S. Geological Survey
 P.O. Box 25046
 Denver, CO 80225

L. E. Shephard [5]
Technical Project Officer for YMP
Sandia National Laboratories
Organization 6310
P.O. Box 5800
Albuquerque, NM 87185

D. H. Appel, Chief [1]
Hydrologic Investigations Program--MS 421
U.S. Geological Survey
P.O. Box 25046
Denver, CO 80225

J. F. Devine [1]
Assistant Director for Engineering Geology
U.S. Geological Survey
106 National Center
12201 Sunrise Valley Drive
Reston, VA 22092

R. W. Craig, Chief [1]
Nevada Operations Program
U.S. Geological Survey
101 Convention Center Drive
Suite 860, MS 509
Las Vegas, NV 89109

L. R. Hayes [1]
Technical Project Officer
Yucca Mountain Project Branch--MS 425
U.S. Geological Survey
P.O. Box 25046
Denver, CO 80225

A. L. Flint [1]
U.S. Geological Survey
MS 721
P.O. Box 327
Mercury, NV 89023

R. E. Lewis [1]
Yucca Mountain Project Branch--MS 425
U.S. Geological Survey
P.O. Box 25046
Denver, CO 80225

D. A. Beck [1]
Water Resources Division
U.S. Geological Survey
6770 So. Paradise Road
Las Vegas, NV 89119

V. R. Schneider [1]
Asst. Chief Hydrologist--MS 414
Office of Program Coordination &
Technical Support
U.S. Geological Survey
12201 Sunrise Valley Drive
Reston, VA 22092

B. T. Brady [1]
Records Specialist
U.S. Geological Survey--MS 421
P.O. Box 25046
Denver, CO 80225

M. D. Voegele [1]
Technical Project Officer for YMP
Science Applications International Corp.
101 Convention Center Dr., Suite 407
Las Vegas, NV 89109

D. L. Fraser, General Manager [1]
Reynolds Electrical & Engineering Co., Inc.
MS 555
P.O. Box 98521
Las Vegas, NV 89193-8521

L. D. Foust [2]
Nevada Site Manager
TRW Environmental Safety Systems
101 Convention Center Drive
Suite P-110, MS 423
Las Vegas, NV 89109

D. L. Koss [1]
Technical Project Officer for YMP
Reynolds Electrical & Engineering Co., Inc.
MS 408
P.O. Box 98521
Las Vegas, NV 89193-8521

C. E. Ezra [1]
YMP Support Office Manager
EG&G Energy Measurements, Inc.
MS V-02
P.O. Box 1912
Las Vegas, NV 89125

B. W. Colston, President/General Manager [1]
Las Vegas Branch
Raytheon Services Nevada
MS 416
P.O. Box 95487
Las Vegas, NV 89193-5487

E. L. Snow, Program Manager [1]
Roy F. Weston, Inc.
955 L'Enfant Plaza, S.W.
Washington, DC 20024

W. P. Kopatich [1]
Technical Project Officer for YMP
Raytheon Services Nevada
Suite P-230, MS 403
101 Convention Center Dr.
Las Vegas, NV 89109

Technical Information Center [1]
Roy F. Weston, Inc.
955 L'Enfant Plaza, S.W.
Washington, DC 20024

Paul Eslinger, Manager [1]
PASS Program
Pacific Northwest Laboratories
P.O. Box 999
Richland, WA 99352

D. Hedges, Vice President, Quality Assurance [1]
Roy F. Weston, Inc.
4425 Spring Mountain Road, Suite 300
Las Vegas, Nevada 89102

A. T. Tamura [1]
Science and Technology Division
Office of Scientific and Technical Information
U.S. Department of Energy
P.O. Box 62
Oak Ridge, TN 37831

Carlos G. Bell, Jr. [1]
Professor of Civil Engineering
Civil and Mechanical Engineering Dept.
University of Nevada, Las Vegas
4505 So. Maryland Parkway
Las Vegas, NV 89154

John Fordham [1]
Water Resources Center
Desert Research Institute
P.O. Box 60220
Reno, NV 89506

P. J. Weeden, Acting Director [1]
Nuclear Radiation Assessment Division
U.S. Environmental Protection Agency
Environmental Monitoring Systems Laboratory
P.O. Box 93478
Las Vegas, NV 89193-3478

David Rhode [1]
Desert Research Institute
P.O. Box 60220
Reno, NV 89506

ONWI Library [1]
Battelle Columbus Laboratory
Office of Nuclear Waste Isolation
505 King Ave.
Columbus, OH 43201

John Fordham [1]
Water Resources Center
Desert Research Institute
P.O. Box 60220
Reno, NV 89506

T. Hay, Executive Assistant [1]
Office of the Governor
State of Nevada
Capitol Complex
Carson City, NV 89710

Eric Anderson [1]
Mountain West Research-Southwest, Inc.
2901 N. Central Ave., #1000
Phoenix, AZ 85012-2730

R. R. Loux [3]
Executive Director
Agency for Nuclear Projects
State of Nevada
Evergreen Center, Suite 252
1802 N. Carson St.
Carson City, NV 89710

The Honorable Cyril Schank [1]
Chairman
Churchill County Board of Commissioners
190 W. First St.
Fallon, NV 89406

C. H. Johnson [1]
Technical Program Manager
Agency for Nuclear Projects
State of Nevada
Evergreen Center, Suite 252
1802 N. Carson St.
Carson City, NV 89710

Dennis A. Bechtel, Coordinator [1]
Nuclear Waste Division
Clark County Department of Comprehensive
Planning
301 E. Clark Ave., Suite 570
Las Vegas, NV 89101

Juanita D. Hoffman [1]
Nuclear Waste Repository Oversight Program
Esmeralda County
P.O. Box 490
Goldfield, NV 89013

Vernon E. Poe [1]
Office of Nuclear Projects
Mineral County
P.O. Box 1600
Hawthorne, NV 89415

Eureka County Board of Commissioners [1]
Yucca Mountain Information Office
P.O. Box 714
Eureka, NV 89316

Les W. Bradshaw [1]
Program Manager
Nye County Nuclear Waste
Repository Project Office
P.O. Box 1767
Tonopah, NV 89049

Brad R. Mettam [1]
Inyo County Yucca Mountain
Repository Assessment Office
P.O. Drawer L
Independence, CA 93526

William Offutt [1]
Nye County Manager
P.O. Box 153
Tonopah, NV 89049

Lander County Board of
Commissioners [1]
315 South Humboldt St.
Battle Mountain, NV 89820

Phillip. A. Niedzielski-Eichner [1]
Nye County Nuclear Waste
Repository Project Office
P.O. Box 221274
Chantilly, VA 22022-1274

Jason Pitts [1]
Lincoln County Nuclear Waste Project Office
P.O. Box 90
Pioche, NV 89043

Florindo Mariani [1]
White Pine County Coordinator
P.O. Box 135
Ely, NV 89301

Judy Foremaster [1]
City of Caliente Nuclear Waste Project Office
P.O. Box 158
Caliente, NV 89008

Glenn Van Roekel [1]
Director of Community Development
City of Caliente
P.O. Box 158
Caliente, NV 89008

Ray Williams, Jr. [1]
P.O. Box 10
Austin, NV 89310

Commission of the European Communities [1]
200 Rue de la Loi
B-1049 Brussels
BELGIUM

Charles Thistlethwaite, AICP [1]
Associate Planner
Inyo County Planning Department
Drawer L
Independence, CA 93526

M. J. Dorsey, Librarian [2]
YMP Research and Study Center
Reynolds Electrical & Engineering Co., Inc.
MS 407
P.O. Box 98521
Las Vegas, NV 89193-8521

Nye County District Attorney [1]
P.O. Box 593
Tonopah, NV 89049

Amy Anderson [1]
Argonne National Laboratory
Building 362, 9700 So. Cass Ave.
Argonne, IL 60439

Economic Development Department [1]
City of Las Vegas
400 E. Stewart Ave.
Las Vegas, NV 89101

Steve Bradhurst [1]
P.O. Box 1510
Reno, NV 89505

Community Planning and Development [1]
City of North Las Vegas
P.O. Box 4086
North Las Vegas, NV 89030

Michael L. Baughman [1]
Intertech Services Corp.
P.O. Box 93537
Las Vegas, NV 89193

Community Development and Planning [1]
City of Boulder City
P.O. Box 61350
Boulder City, NV 89006

Dr. Moses Karakouzian [1]
1751 E. Reno, #125
Las Vegas, NV 89119

Lawrence Berkeley National Laboratory
University of California
Berkeley, CA 94720