United States
Environmental Protection
Agency

National Risk Management
Research Laboratory
Ada, OK 74820

EPA/600/R-97/007
February 1997

EPA

Ground-Water Model Testing: Systematic Evaluation and Testing of Code Functionality and Performance

GROUND-WATER MODEL TESTING: SYSTEMATIC EVALUATION AND TESTING OF CODE FUNCTIONALITY AND PERFORMANCE

by

Paul K.M. van der Heijde
and
David A. Kanzer

Colorado School of Mines
International Ground Water Modeling Center
Golden, Colorado 80401

CR-818719

Project Officer

Joseph R. Williams
Subsurface Protection and Remediation Division
National Risk Management Research Laboratory
Ada, Oklahoma 74820

NATIONAL RISK MANAGEMENT RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
CINCINNATI, OHIO 45268

DISCLAIMER NOTICE

The U.S. Environmental Protection Agency through its Office of Research and Development funded and managed the research described here under Cooperative Agreement Number CR-818719 to the Colorado School of Mines in Golden, Colorado. It has been subjected to Agency's peer and administrative review, and has been approved for publication as an EPA document.

All research projects making conclusions or recommendations based on environmentally related measurements and funded by the Environmental Protection Agency are required to participate in the Agency Quality Assurance Program. This project did not involve environmentally related measurements and did not involve a Quality Assurance Project Plan.

Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

FOREWORD

The U.S. Environmental Protection Agency is charged by Congress with protecting the Nation's land, air, and water resources. Under a mandate of national environmental laws, the Agency strives to formulate and implement actions leading to a compatible balance between human activities and the ability of natural systems to support and nurture life. To meet these mandates, EPA's research program is providing data and technical support for solving environmental problems today and building a science knowledge base necessary to manage our ecological resources wisely, understand how pollutants affect our health, and prevent or reduce environmental risks in the future.

The National Risk Management Research Laboratory is the Agency's center for investigation of technological and management approaches for reducing risks from threats to human health and the environment. The focus of the Laboratory's research program is on methods for the prevention and control of pollution to air, land, water, and subsurface resources; protection of water quality in public water systems; remediation of contaminated sites and ground water; and prevention and control of indoor air pollution. The goal of this research effort is to catalyze development and implementation of innovative, cost-effective environmental technologies; develop scientific and engineering information needed by EPA to support regulatory and policy decisions; and provide technical support and information transfer to ensure effective implementation of environmental regulations and strategies.

The use of computer-based models for ground-water model predictions continues to proliferate, and has become an integral part of most site investigation, management and remedial decision-making activities. The reliability of these assessments and decisions must be demonstrated through evaluation of the correctness of the conceptual model, the availability and quality of model data, and the adequateness of the predictive tools, or computer-based models. This report presents issues and approaches related to the testing of computer codes utilized in predicting ground-water responses. It is the intent of this report to provide the ground-water modeling community with a useful tool for the evaluation of computer codes during both the development and acceptance stages of model application. The report also includes three MathCAD worksheets containing analytical solutions discussed in the testing procedures.

Clinton W. Hall
Subsurface Protection and Remediation Division
National Risk Management Research Laboratory

ABSTRACT

Effective use of ground-water simulation codes as management decision tools requires the establishment of their functionality, performance characteristics, and applicability to the problem at hand. This is accomplished through application of a systematic code-testing protocol and code selection strategy. This report describes a code testing protocol, containing two main elements: functionality analysis and performance evaluation. Functionality analysis is the description and measurement of the capabilities of a simulation code. Performance evaluation concerns the appraisal of a code's operational characteristics (*e.g.,* computational accuracy and efficiency, sensitivity for problem design and model parameters, and reproducibility). Furthermore, this report discusses applicability assessment, *i.e.*, providing information on a code's capabilities in simulating complex, real-world ground-water problems.

The protocol for testing and evaluation of a code's functionality and performance consists of a series of steps and procedures. First, the code is analyzed with respect to its simulation functions and operational characteristics. This is followed by the design or selection of relevant test problems, the so-called test strategy. The set of test problems is chosen such that all code functions and features are addressed. Results of the testing are documented in tables and matrices, which provide a quick overview of the completeness of the testing, in various types of informative graphs, and with a set of statistical measures indicative of the test results. The actual testing may take the form of: (1) benchmarking using known, independently derived solutions; (2) intracomparison using different code functions inciting the same system responses; (3) intercomparison with comparable simulation codes; or (4) comparison with field or laboratory experiments. The results of the various tests are analyzed to identify performance strengths and weaknesses of code and testing procedures. The final step consists of documenting the results in report form, archiving the baselined code and test files, and communicating the results to the different audiences in an appropriate format. The results of code testing are analyzed using standardized statistical and graphical techniques, and presented using informative tables, tabular matrices, and graphs.

The protocol is demonstrated and evaluated using the three-dimensional finite difference flow and solute transport simulation code, FTWORK.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The mission of the International Ground Water Modeling Center (IGWMC) is to advance and improve the use of modeling methodologies in the development and implementation of effective ground-water management procedures by regulatory and planning agencies, and by industry. The study, reported on in this document, was conceived as a major initiative in support of the Center's mission by developing a set of systematic ground-water code testing procedures and code testing tools, brought together in a comprehensive protocol. Major funding for the study and related efforts in disseminating the study results and gaining acceptance of the protocol has been provided by the Robert S. Kerr Environmental Research Center, U.S. Environmental Protection Agency, Ada, Oklahoma. Among others, the code testing and evaluation protocol described in this report is being adapted by the American Society for Testing and Materials (ASTM) as a Standard Guide; the functionality description part of the protocol forms the basis for an ASTM Standard Guide, under development, for the description of the functionality and the classification of ground-water modeling codes.

Many elements of the protocol development and code testing exercises presented in this report are based on the unpublished M.Sc. thesis "The Design and Evaluation of Testing Protocols for Rectangularly Discretized Ground-Water Simulation Models" by David Kanzer, defended at the Colorado School of Mines (CSM) in January 1995. Additional valuable contributions to the research were made by Suzanne Paschke and Don Meyer, graduate students at CSM. The authors are grateful for the extensive administrative support provided by IGWMC Staff Assistants Mary Pigman and Mary Vigil, and for the constructive advice from Dr. Keith Turner, Professor of Geology and Geological Engineering at the Colorado School of Mines.

Paul K.M. van der Heijde
Golden, Colorado

EXECUTIVE SUMMARY

INTRODUCTION

Reliability of ground-water model predictions typically depends on the correctness of the conceptual model, the availability and quality of model data, and the adequateness of the predictive tools. In ground-water modeling, the predictive tools consist of one or more computer codes for data analysis, system simulation, and presentation of results. This report focuses on the testing of the computer codes used in predicting ground-water responses. The importance of this aspect of ground-water modeling is illustrated by the efforts currently under way within the American Society for Testing and Materials (ASTM) to codify the systematic description and the testing of the capabilities of ground-water modeling codes, and within the American Society of Civil Engineers (ASCE) to provide guidance on this issue.

The development of a ground-water modeling code typically consists of: 1) definition of design criteria and determination of applicable software standards and practices; 2) the development of algorithms and program structure; 3) computer programming; 4) preparation of documentation; 5) code testing; and 6) independent review of scientific principles, mathematical framework, software and documentation. Proper Quality Assurance (QA) requires that when the development of a ground-water modeling code is initiated, procedures are formulated to ensure that the final product conforms with the design objectives and specifications, and that it correctly performs the incorporated functions. These procedures cover the formulation and evaluation of the code's theoretical foundation and code design criteria, the application of coding standards and practices, and the establishment of the code's credentials through review, systematic testing of its functional design, and evaluation of its performance characteristics. The two major approaches to achieve acceptance of a ground-water modeling code are: 1) the evaluation or (peer) review process covering all phases of the code development; and 2) quantitative comparison with independently obtained data for the reference ground-water system.

CODE TESTING

A systematic approach to code testing combines elements of error-detection, evaluation of the operational characteristics of the code, and assessment of its suitability to solve certain types of management problems, with dedicated test problems, relevant test data sets, and informative performance measures. The results of code testing are expressed in terms of *correctness* (*e.g.,* in comparison with a benchmark), *reliability* (*e.g.,* reproducibility of results, convergence and stability of solution algorithms, and absence of terminal failures), *efficiency* of coded algorithms (in terms of numerical accuracy versus code execution time, and memory and mass storage requirements), and *resources required* for model setup and analysis (*e.g.,* input preparation time, effort needed to make output ready for graphic analysis).

The code-testing protocol described in this report is applied in a step-wise fashion. First, the code is analyzed with respect to its simulation functions and operational characteristics. Potential code performance issues are identified, based on analysis of simulated processes, mathematical solution methods, computer limitations and execution environment. This is followed by the formulation of a test strategy, consisting of design or selection of relevant test problems. The set of test problems is chosen such that all code functions and features of concern are addressed. Results of the testing are documented in tables and matrices providing an overview of the completeness of the testing, in various types of informative graphs, and with a set of statistical measures. The actual testing may take the form of benchmarking using known, independently derived solutions, intra-comparison using different code functions inciting the same system responses, inter-comparison with comparable simulation codes, or comparison with field or laboratory experiments. It is important that each test is documented with respect to test objectives, model setup for both the tested code and the benchmark, if applicable (structure, discretization, parameters), and results for each test (for both the tested code and the benchmark).

Functionality of a ground-water modeling code is defined as the set of functions and features the code offers the user in terms of model framework geometry, simulated processes, boundary conditions, and analytical and operational capabilities. The code's functionality needs to be defined in sufficient detail for potential users to assess the code's utility, as well as to enable the code developer to design a meaningful code testing strategy. *Functionality analysis* involves the identification and description of the code's functions, and the subsequent qualitative evaluation of each code function or group of functions for conceptual correctness and error-free operation. The information generated by functionality analysis is organized into a summary structure, or matrix, that brings together the description of code functionality, code-evaluation status, and appropriate test problems. This *functionality matrix* is formulated by combining a complete description of the code functions and features with the objectives of the test cases. The functionality matrix illustrates the extent of the performed functionality analysis.

CODE TESTING AND EVALUATION PROTOCOL

step 1	analyze the code documentation with respect to simulation functions, operational features, mathematical framework, and software implementation;
step 2	identify code performance issues based on understanding of simulated processes, mathematical methods, computer limitations, and software environment;
step 3	develop testing strategy that addresses relevant code functionality and performance issues, including selection and/or design of test problems and determination of appropriate evaluation measures;
step 4	execute test problems and analyze results using selected graphic and statistical evaluation techniques;
step 5	collect code performance issues and code test problems in overview tables and display matrices reflecting correctness, accuracy, efficiency, and field applicability;
step 6	identify performance strengths and weaknesses of code and testing procedure;
step 7	document each test setup and results in report form and as electronic files (text, data, results, graphics); and
step 8	communicate results (*e.g.*, executive summary, overview report. etc.).

Performance evaluation is aimed at quantitatively characterizing the operational characteristics of the code in terms of:

- computational accuracy and efficiency;
- operational reliability;
- sensitivity for problem design and model parameters; and
- level of effort and resources required for model setup and simulation analysis.

Results of the performance evaluation are expressed both in checklists and in tabular form. Reporting on performance evaluation should provide potential users information on the performance as a function of problem complexity and setup, selection of simulation control parameters, and spatial and temporal discretization. The functionality matrix and performance tables, together with the supporting test results and comments, should provide the information needed to select a code for a site-specific application and to evaluate the appropriateness of a code used at a particular site.

TESTING STRATEGY

Comprehensive testing of a code's functionality and performance is accomplished through a variety of test methods. Determining the importance of the tested functions and the ratio of tested versus non-tested functions provides an indication of the completeness of the testing. Based on the analysis of functionality and performance issues, a code testing strategy is developed. An effective code testing strategy consists of:

- formulation of test objectives (as related to code functionality and performance issues), and of test priorities;
- selection and/or design of test problems and determination of type and extent of testing for selected code functions;
- determination of level of effort to be spent on sensitivity analysis for each test problem;
- selection of the qualitative and quantitative measures to be used in the evaluation of the code's performance; and
- determination of the level of detail to be included in the test report and the format of reporting.

In developing the code testing strategy, code applicability issues should be considered in terms of the types of ground-water management problems the code is particularly suitable to handle. Specifically, attention is given in the design of test problems to representative hydrogeology, engineering designs, and management scenarios.

The test procedure includes three levels of testing. At Level I, a code is tested for correctness of coded algorithms, code logic and programming errors by: 1) conducting step-by-step numerical walk-throughs of the complete code or through selected parts of the code; 2) performing simple, conceptual or intuitive tests aimed at specific code functions; and 3) comparing with independent, accurate benchmarks (*e.g.*, analytical solutions). If the benchmark computations themselves have been made using a computer code, this computer code should in turn be subjected to rigorous testing by comparing computed results with independently derived and published data.

At Level II, a code is tested to: 1) evaluate functions not addressed at Level I; and 2) evaluate potentially problematic combinations of functions. At this level, code testing is performed by intracomparison (*i.e.*, comparison between runs with the same code using different functions to represent a particular feature), and intercomparison (*i.e.*, comparison between different codes simulating the same problem). Typically, synthetic data sets are used representing hypothetical, often simplified ground-water systems.

At Level III, a code (and its underlying theoretical framework) is tested to determine how well a model's theoretical foundation and computer implementation describe actual system behavior, and to demonstrate a code's applicability to representative field problems. At this level, testing is performed by simulating a field or laboratory experiment and comparing the calculated and independently observed cause-and-effect responses. Because measured values of model input, system parameters and system responses are samples of the real system, they inherently incorporate measurement errors, are subject to uncertainty, and may suffer from interpretive bias. Therefore, this type of testing will always retain an element of incompleteness and subjectivity.

The test strategy requires that first Level I testing is conducted (often during code development), and, if successfully completed, this is followed by Level 2 testing. The code may gain further credibility and user confidence by subjecting it to Level 3 testing (*i.e.*, field or laboratory testing). Although, ideally, code testing should be performed for the full range of parameters and stresses the code is designed to simulate, in practice this is often not feasible due to budget and time constraints. Therefore, prospective code users need to assess whether the documented tests adequately address the conditions expected in the target application(s). If previous testing has not been sufficient in this respect, additional code testing may be necessary.

EVALUATION MEASURES

Evaluation of code testing results should be based on: 1) visual inspection of the graphical representation of variables computed with the numerical model and its benchmark; and 2) quantitative measures of the goodness-of-fit. Such quantitative measures, or *evaluation* or *performance criteria* characterize the differences between the results derived with the simulation code and the benchmark, or between the results obtained with two comparable simulation codes.

Graphical measures are especially significant to obtain a first, qualitative impression of test results, and to evaluate test results that do not lend themselves to statistical analysis. For example, graphical representation of solution convergence characteristics may indicate numerical oscillations and instabilities in the iteration process. Practical considerations may prevent the use of all data-pairs in the generation of graphical measures. The conclusions from visual inspection of graphic representations of testing results are described qualitatively (and subjectively).

Useful quantitative evaluation measures for code testing: 1) *Mean Error* (ME), defined as the mean difference (i.e., deviation) between the dependent variable calculated by the numerical model and the benchmark value of the dependent variable; 2) *Mean Absolute Error* (MAE), defined as the average of the absolute values of the deviations; 3) *Positive Mean Error* (PME) and *Negative Mean Error* (NME), defined as the ME for the positive deviations and negative deviations, respectively; 4) *Mean Error Ratio* (MER), a composite measure indicating systematic overpredicting or underpredicting by the code; 5) *Maximum Positive Error* (MPR) and *Maximum Negative Error* (MNE), defined as the maximum positive and negative deviation, respectively, indicating potential inconsistencies or sensitive model behavior; and 6) *Root Mean Squared Error* (RMSE), defined as the square root of the average of the squared differences between the dependent variable calculated by the numerical model and its benchmark equivalent.

Various computed variables may be the focus of graphic or statistical comparison, including hydraulic heads (in space and time), head gradients, global water balance, internal and boundary fluxes, velocities (direction and magnitude), flow path lines, capture zones, travel times, and location of free surfaces and seepage surfaces, concentrations, mass fluxes, and breakthrough curves at observation points and sinks (wells, streams).

DISCUSSION

The functionality analysis and performance evaluation protocol presented in this report provides a comprehensive framework for systematic and in-depth evaluation of a variety of ground-water simulation codes. While allowing flexibility in implementation, it secures, if properly applied, addressing all potential coding problems. It should be noted that the protocol does not replace scientific review nor the use of sound programming principles. Most effectively, the code testing under the protocol should be performed as part of the code development process. Additional testing in accordance with the protocol may be performed under direction of regulatory agencies, or by end-users. If properly documented, code testing in accordance with the protocol supports effective independent review and assessment for application suitability. As such, the protocol contributes significantly to improved quality assurance in code development and use in ground-water modeling.

1. INTRODUCTION

1.1. BACKGROUND

Ground-water modeling has become an important methodology in support of the planning and decision-making processes involved in ground-water management. Ground-water models provide an analytical framework for obtaining an understanding of the mechanisms and controls of ground-water systems and the processes that influence their quality, especially those caused by human intervention in such systems. For managers of water resources, models may provide essential support for planning and screening of alternative policies, regulations, and engineering designs affecting ground-water. This is particularly evident with respect to ground-water resources development, ground-water protection, and aquifer restoration.

Assessment of the validity of modeling-based projections is difficult and often controversial (*e.g.,* van der Heijde and Park, 1986; Tsang, 1987; Tsang, 1991; Konikow and Bredehoeft, 1992; Bredehoeft and Konikow, 1993). The four major components contributing to the success or failure of a modeling exercise are:

- the availability of field information (*i.e.*, quality and completeness of data);
- the correctness of the conceptual site model and the level of detail in the model schematization;
- the type and quality of the analytical tools (*e.g.*, geostatistical and hydrogeological software), and
- the competence of the team of experts involved in the preparation of the modeling-based advice.

As computer codes are essential building blocks of modeling-supported management, it is crucial that before such codes are used as planning and decision-making tools, their credentials are established and their suitability determined through systematic evaluation of their correctness, performance, sensitivity to input uncertainty, and applicability to typical field problems. Such a systematic approach, in this report referred to as *code testing and evaluation protocol*, should consist of evaluation or review of the underlying physical concepts and mathematical model formulations, a rather qualitative process, and extensive code evaluation and testing, a more quantitative process. Without subjecting a ground-water simulation code to systematic testing and evaluation, results obtained with the code may suffer from low levels of confidence (van der Heijde and Elnawawy, 1992).

Code testing (or model testing if the underlying principles are explicitly evaluated) is significantly more than determining that "the code works" (*i.e.*, the modeler's aim to minimize errors that cause the model not to work), or that "the code does not work" (*i.e.*, the user's aim to minimize accepting an incorrect model) (Burns, 1983). It might prove very difficult to come up with objective criteria to make such judgment, specifically as ground-water modeling codes are always based on approximative and simplified concepts. Therefore, acceptance of a modeling code depends not only on a series of successful tests, but also on a history of successful applications to a variety of site conditions and management problems, especially if one or more of such successful applications reflect the conditions present at the project site.

1.2. CODE TESTING ISSUES

To date, most ground-water model evaluations have been limited to rather qualitative peer review of model theory, while code testing has been restricted to partial and ad-hoc testing (van der Heijde and Elnawawy, 1992). Often, published test results do not provide insight in the completeness of the testing procedure, are difficult to reproduce, and only partially analyzed. In most cases, objectives of test problems are absent, poorly formulated, or when present, not evaluated. Furthermore, specification of code functions and operational characteristics, needed by a user to make educated decisions regarding code selection and implementation, is often incomplete, inaccurate, or dispersed throughout the documentation. In many cases, determining if a simulation code includes a particular, desired function can require significant effort on the side of a reviewer or potential user.

Inconsistent and incomplete code testing by code developers can be attributed to the lack of a standard code testing and evaluation protocol. In the absence of such a framework they may find it difficult to determine when a code (and its underlying mathematical model) has been adequately tested. Consequently, there are wide variations in the level of code testing performed, as well as in the documentation of test results.

Taking a systematic, well-defined and controlled approach to the development of ground-water simulation codes is an essential part of Quality Assurance (QA). Van der Heijde and Elnawawy (1992) describe the QA in code development and application in detail. An important element of such QA is code testing and performance evaluation.

1.3. CODE TESTING AND EVALUATION PROTOCOL

When the development of a ground-water modeling code is initiated, procedures are formulated to ensure that the final product conforms with the design objectives and specifications, and that it correctly performs the incorporated functions. These procedures cover the formulation and evaluation of the code's theoretical foundation and code design criteria, the application of coding standards and practices, and the establishment of the code's credentials through review and testing of its functional design and evaluation of its performance characteristics. To evaluate ground-water modeling software in a systematic and consistent manner, the International Ground Water Modeling Center (IGWMC) has formulated a quality assurance framework for code development that includes scientific and technical reviews, a three-level code testing strategy, and code baseline documentation (van der Heijde and Elnawawy, 1992).

In this report, the code testing part of the quality assurance framework has been expanded with a systematic functionality analysis and performance evaluation protocol. The protocol provides a framework of procedures and test problems to quantitatively and qualitatively characterize various types of ground-water simulation codes. It includes strategies for design of test problems and evaluating test results. The application of the protocol is illustrated using the block-centered finite difference model for simulation of three-dimensional ground-water flow and solute transport in saturated media, FTWORK (Faust *et al.*,1990).

It should be noted that quality assurance in the development of ground-water modeling codes cannot guarantee acceptable quality of the code or a ground-water modeling study in which the code has been used. However, adequate quality assurance can provide safeguards against the use in a modeling study of faulty codes or incorrect theoretical considerations and assumptions. Furthermore, there is no way to guarantee that modeling-based advice is entirely correct, nor that the ground-water model used in the preparation of the advice (or any scientific model or theory, for that matter) can ever be proven to be entirely correct. Rather, a model can only be invalidated by disagreement of its predictions with independently derived observations of the studied system because of incorrect application of the selected code, the selection of an inappropriate code, the use of an inadequately tested code, or invalidity of or errors in the underlying theoretical framework.

Although the protocol has been developed using a numerical simulation code for site-specific saturated zone flow and transport, it has been designed to be applicable to codes for simulation of other systems. Such codes would include those for flow and transport in the vadose zone, and other type codes, such as those representing analytical solutions, or have been designed for programmatic assessments.

Complete adherence to this protocol may not always be feasible. If this protocol is not integrally followed, the elements of non-compliance should be clearly identified and the reasons for the partial compliance should be given. For example, partial compliance might result from lack of benchmark solutions, or is, by design, focused on only those code functions relevant to the user.

1.4. REPORT ORGANIZATION

The report begins with a review of existing code testing literature to evaluate past code testing programs, determine key elements of a comprehensive code testing protocol, formulate efficient qualitative test assessment methods, and compile effective test problems. This is followed by the formulation of a comprehensive code testing protocol and discussion of testing strategies. Methods for the development of code-evaluation problem sets are presented followed by a discussion of various graphical and statistical tools for evaluation of code testing results. This protocol is then applied to the category of codes designed to simulate three-dimensional flow and solute transport in the saturated zone of the subsurface. Finally, an example of the protocol's use is presented featuring the FTWORK code, followed by a discussion of the protocol's utility.

1.5. TERMINOLOGY

The terminology used in ground-water modeling often leads to confusion and heated discussions. For example, the term "ground-water model" may refer to the generalized computer code designed for application to many different sites, or to the use of such code at a particular site as an “operational model.” Therefore, a glossary of terms is provided at the end of the report. Where possible, the description of these terms follows the definitions agreed upon in Subcommittee D18.21 of the American Society for Testing and Materials (ASTM).

There are two terms in describing ground-water model evaluation procedures that have recently become rather controversial: “verification” and “validation.” Konikow and Bredehoeft (1992) suggest that a ground-water model "cannot be proven or validated, only tested and invalidated." They argue that ground-water models are only conceptual approximations of real world systems and due to the random nature of many parameters and uncertainty in their measurement simulation codes render non-unique solutions. They conclude that "the terms verification and validation are misleading and their use in ground-water science should be abandoned in favor of more meaningful model-assessment descriptors." This statement makes sense in the context of a site-specific ground-water model application, but does not agree with common software engineering practices (Adrion *et al.*, 1986). Van der Heijde and Elnawawy (1992) note that in software testing literature the terms program or code “verification” and “validation” are well-defined and widely used. In converting the use of these software engineering terms to ground-water modeling, they suggest that most types of ground-water modeling codes cannot be truly verified or validated in a quantitative sense, rather that such codes can only be analyzed for deviation from some reference or benchmark and characterized with respect to other performance issues. In this report, the latter approach to code testing is referred to as functionality analysis and performance evaluation of the software.

The use of various code development and testing terms is directly related to the code development process as illustrated in Figure 1-1. The object for model research in ground water is a subset of the hydrologic system, called the reference system. It contains selected subsurface and sometimes surface elements of the global hydrologic system. The selection of a particular reference system is influenced by regulatory and management priorities, and by the nature of the hydrologic system. The conceptual model of the selected reference system forms the basis for quantifying the causal relationships among various components of this system, and between this system and its environment. These relationships are defined mathematically, resulting in a mathematical model.

If the solution of the mathematical equations is complex or when many repetitious calculations are required, this solution is implemented on a computer system, resulting in a computer code. The conceptual formulations, mathematical descriptions, and computer coding constitute the (generic) mode. Attributing the parameters and stresses in the generic model (*i.e.*, parameterization) resulting from characterization of the reference system, provides an operational model of the reference system.

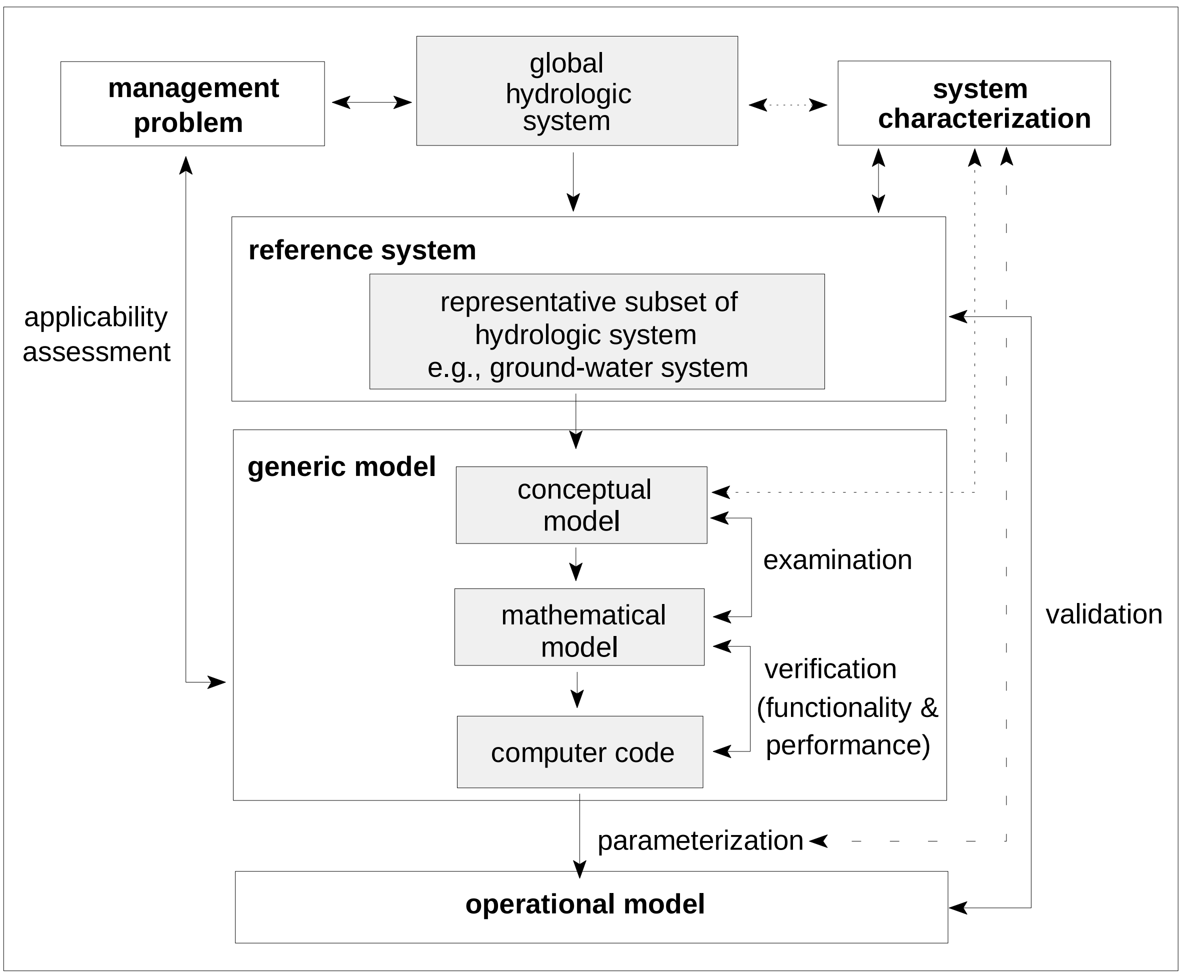


Figure 1-1. Code development and testing concepts.

1.5.1. <u>The Term "Validation"</u>

Historically, validation studies in ground-water modeling are based on the use of well-monitored laboratory experiments and field studies. Sometimes such studies include post-audits. This approach is used to increase the confidence in the ability of simulation codes to represent the real world problems for which they have been designed, as well as the credibility of code-based predictions. Such studies require large amounts of data and can be very expensive. Furthermore, the measured field data, used as input parameters or system response benchmarks, are only small samples of the modeled domain and are subject to measurement error, which reduces the value of this "validation" approach in code testing. Because of the inherent problems to "code validation," this process is considered rather subjective (National Research Council, 1990).

During the HYDROCOIN code testing project (see Section 2) the meaning of the term "model validation" has been extensively discussed. In the context of that project, validation is performed by comparing modeling results with experimental results (HYDROCOIN, 1987). A framework for model validation was formulated, aimed at showing that a model correctly predicts physical phenomena. In the context of the performance assessment of radioactive waste repositories, this involves calibration, comparison between calculations and experimental data, and convincing the scientific community, decision makers and the general public. The framework includes the following elements:

- description of the physical system and model calibration;
- prediction of a performance measure that is independent of the data used for model calibration;
- comparison with the results of alternative models;
- analysis of the discrepancies between different models and between the models and the experimental data; and
- presentation of the results to the scientific community, decision makers and the general public.

The U.S. Nuclear Regulatory Commission considers validation in the context of code application as a process that is concerned with providing assurance that the model reflects reality (Davis *et al.*, 1991). If a model is considered "not invalid," it does not constitute "validity." It only provides a means for the modeler to demonstrate that the model (*i.e.*, code application) is not incorrect. This helps in building confidence in the model's predictions, especially as perfection (*i.e.*, determining if a model is "valid") is not possible.

In this report, *code validation* in ground-water modeling is defined as the process of determining how well a ground-water modeling code's theoretical foundation and computer implementation describe actual system behavior in terms of the degree of correlation between calculated and independently observed cause-and-effect responses of the reference ground-water system for which the code has been developed. Code validation, as defined above, is by nature a subjective and open-ended process; the result of the code validation process is a level of confidence in the code's ability to simulate the reference system, or the determination of the code's inability to simulate such a system. As there is no practical way to determine that a ground-water code correctly simulates all variants of the reference system, the code can never be considered "validated."

1.5.2. The Term "Verification"

In ground-water modeling, the term "verification" has been used in two different ways: 1) evaluating the correctness of a computer program; and 2) evaluating the correctness of a calibrated model of a regional or site-specific ground-water system (Anderson and Woesnner, 1992; National Research Council, 1990). ASTM (1984) lists the purposes of model verification as: 1) establishing the correctness and accuracy of the computational algorithms used to solve the governing equations; and 2) ensuring that the computer code is fully operational and that there are no problems in obtaining a solution. Due to the practical limitations of code validation in ground-water modeling, most of the documented code testing has been limited to what is defined in this report as "code verification," not to be confused with the terms "model verification" or "application verification" (ASTM, 1993).

In this report, *code verification* in ground-water modeling is defined as the process of demonstrating the consistency, completeness, correctness and accuracy of a ground-water modeling code with respect to its design criteria by evaluating the functionality and operational characteristics of the code and testing embedded algorithms and internal data transfers through execution of problems for which independent benchmarks are available. A code can be considered "verified" when all its functions and operational characteristics have been tested and have met specific performance criteria, established at the beginning of the verification procedure. Considering a code verified does not imply that a ground-water model application constructed with the code is verified.

1.5.3. Closure

Although verification and validation are two commonly used terms describing components of code evaluation in ground-water modeling, in this report they are only referred to for cross-referencing purposes. Three new terms, directly related to specific objectives of code evaluation processes, are defined and discussed: *functionality analysis*, *performance evaluation*, and *applicability assessment*.

2. GROUND-WATER CODE TESTING PRACTICES

2.1. HISTORIC DEVELOPMENT

2.1.1. Test Approaches

Since the late 1960s, when ground-water modeling became a focus of research and field application, code developers and users have been concerned with the utility and performance of ground-water modeling codes. The major approach to address these concerns has proven to be *code testing.* Codes representing an analytical model were typically tested by comparison with published results of the analytical solution involved, or by comparison with manual calculations. Initially, testing of codes based on a numerical solution to the governing equations took place in three forms:

1) benchmarking using independently derived solutions to the simulation test problem, often in the form of analytical solutions (*e.g.*, Pinder and Bredehoeft, 1968; Witherspoon, *et al.*, 1968; Prickett and Lonnquist, 1971; Cooley, 1972; Ward *et al.*, 1984) (see Fig. 2-1);

2) simulation of and comparison with well-characterized laboratory experiments or analogs (*e.g.*, Prickett and Lonnquist, 1971; Sá Da Costa and Wilson, 1979; Aral and Tang, 1988) (see Fig. 2-2); and

3) field demonstration (sometimes called “field comparison”) and example application (*e.g.*, Pinder and Bredehoeft, 1968; Frind and Pinder, 1973; Konikow and Bredehoeft, 1978; Voss, 1984; Ward *et al.*, 1984).

When more complex numerical modeling codes became available, which could not be fully tested using these three approaches, attention focused on two additional test methods:

4) simulation of well-characterized and monitored field experiments (also called “field validation”) (*e.g.*, Frind and Hokkanen, 1987; Molson *et al.*, 1992; Hills et al., 1994); and

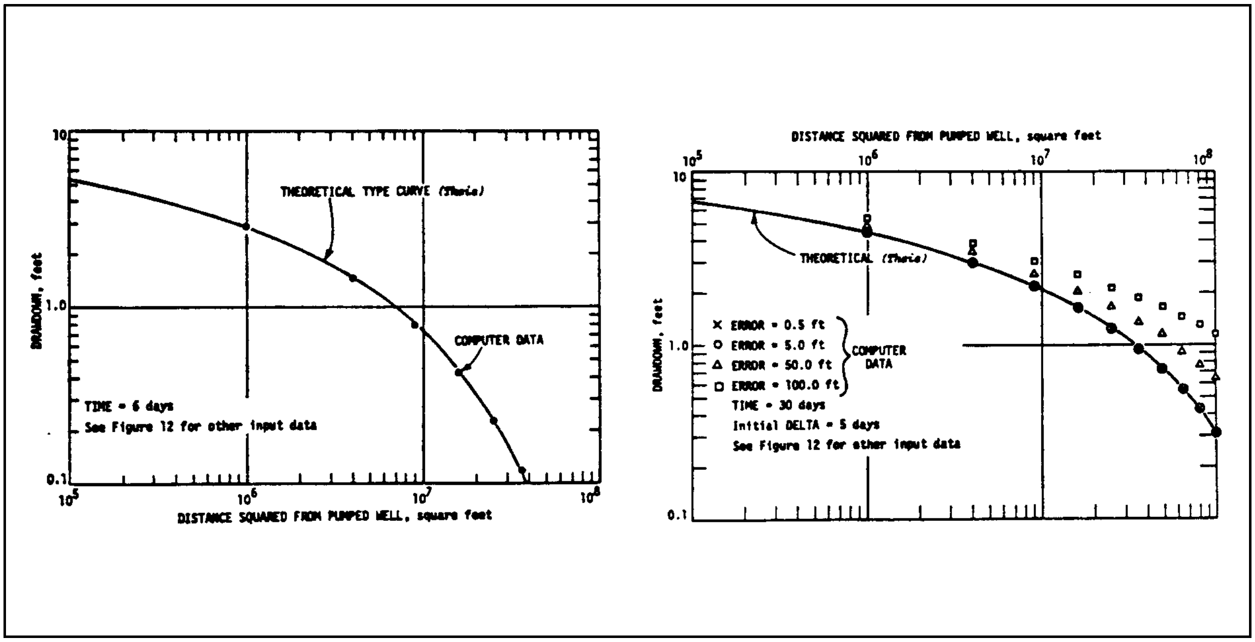


Figure 2-1. Examples of comparison of numerical and analytical solutions (from Prickett and Lonnquist, 1971)

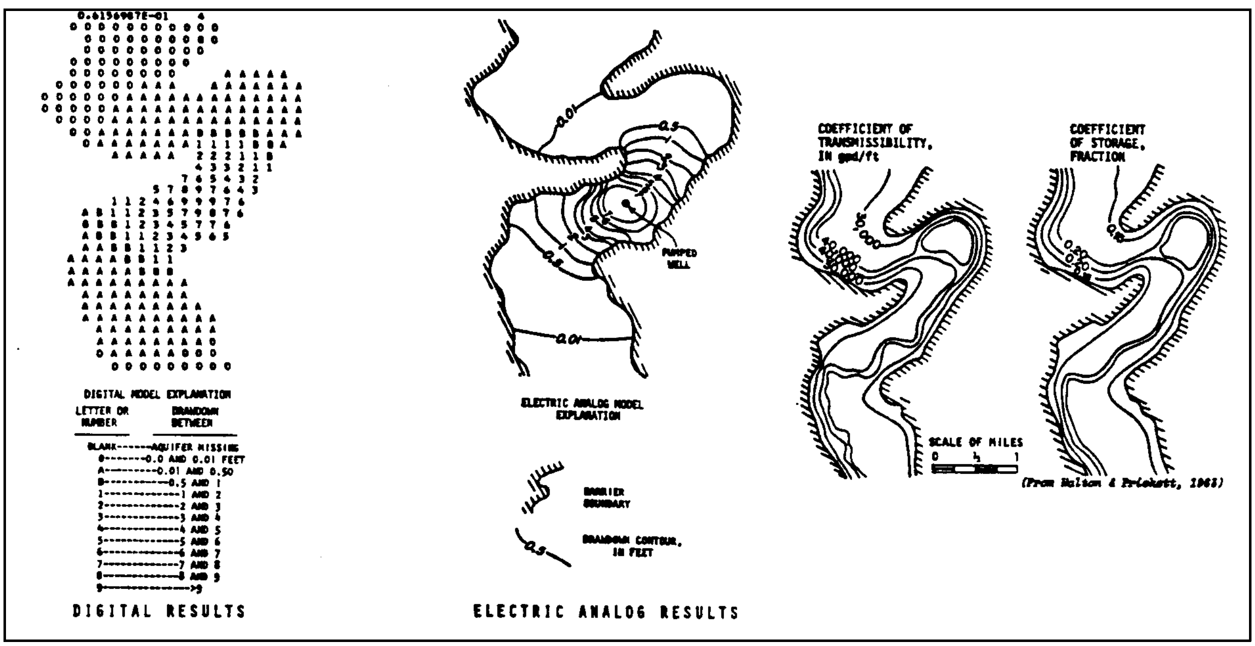


Figure 2-2. Example of comparison of numerical and laboratory analog solutions (from Prickett and Lonnquist, 1971)

5) code-intercomparison using hypothetical problems with synthetic data sets (e.g., Burnett and Frind, 1987; Park and Liggett, 1991).

Many of the test problems developed in the 1970s and early 1980s have become "classical" problems, used by other researchers to demonstrate the correctness of their modeling codes. Segol (1994) describes many of such test problems, as well as sample applications of these problems in the testing of computer codes (and their underlying mathematical models). It should be noted that analytical models in turn have been compared with laboratory experiments and field studies to demonstrate their correctness and to understand their limitations (*e.g.*, Hoopes and Harleman, 1967; Chandler and McWhorter, 1975; Simmons and McKeon, 1984).

Early numerical modeling efforts focused on two-and three-dimensional saturated zone flow systems. There is a relative abundance of analytical solutions available for saturated flow problems, specifically with respect to well and drain hydraulics (*e.g.*, Bear, 1979; DeWiest, 1966; Edelman, 1972; Huisman, 1972; Marino and Luthin, 1982). A recent compilation of analytical drain solutions has been prepared by Beljin and Murdoch (1994). Many of these analytical solutions pertain to one-dimensional or radial-symmetric flow problems with different flow conditions, including steady-state and transient flow, single and multiple aquifers, confined, leaky-confined, and unconfined aquifers, anisotropy, partial penetration of production and observation wells and drains, and time-varying boundary conditions or aquifer stresses. Appropriate use of the principle of superposition enhances the utility of these solutions. As a result, the variety of saturated situations described by the available analytical solutions supports their widespread use for testing two- and three-dimensional numerical flow models. Although analytical solutions are, in general, highly simplified representations of real-world conditions, they are very valuable in code testing as they provide an independent check on the correctness and accuracy of numerical models, and insight in the sensitivity of the results to key parameters (Burns, 1983).

The five-point approach to code evaluation has also been used for more complex problems, such as flow in the unsaturated zone, solute and heat transport in the saturated and unsaturated zone, flow and transport processes in fractured rock, and salt-water intrusion problems. Where available, analytical models are preferred benchmarks for codes designed to simulate such problems (e.g., Ward *et al.*, 1984; Essaid, 1990). However, the use of analytical solutions in testing is severely restricted by limitations resulting from the assumptions made in deriving the solutions to the governing equations, as well by the lesser extent of their availability. For example, the application of analytical solutions in the testing of numerical solute transport codes has been limited by the rather generally used assumption of uniform ground-water flow in these solutions (*e.g.*, Bear, 1979; Beljin, 1992; Cleary and Ungs, 1978; Fried, 1975; Javandel *et al.*, 1984; van Genuchten and Alves, 1982; Walton, 1984). There are relatively few analytical transport solutions dealing with nonuniform flow conditions (*e.g.*, Chen, 1985; Hoopes and Harleman, 1967; Lenau, 1973). Still, initial testing of transport codes is often performed by comparing with one or more analytical solutions to explore a code's ability to simulate transport conditions known for the challenge they provide to numerical techniques.

Due to the lack of analytical solutions for testing of complex simulation codes, testing of these codes has focused on: 1) intercomparison of computational results derived by codes designed to handle similar types of problems (*e.g.*, Beljin, 1988; INTRACOIN 1984, 1986; HYDROCOIN, 1988, 1990; Lobo Ferreira, 1988); 2) field comparison (*e.g.*, Ward *et al.*, 1984); and 3) example application to either real field problems (e.g., Faust *et al.*, 1993) or hypothetical situations (*e.g.*, idealized field problems; Kaluarachchi and Parker, 1989).

The mathematical descriptions of the physical processes represented in the models has frequently been compared with or directly derived from well-controlled laboratory experiments as part of a research project leading to model formulation (Warrick *et al.*, 1971; Haverkamp *et al.*, 1977). Typically, these mathematical formulations are subject to peer review before being accepted as the base for an operational computer code. Comparison with these experimental results allows the researcher to discriminate between alternative mathematical formulations, to determine the level of mechanistic detail needed in a reasonably accurate model representation, and to analyze model sensitivity for physical parameters and numerical formulations (Burns, 1983). An important advantage of laboratory experiments for code testing is that they are performed in a well-controlled environment that minimizes uncertainty in initial and boundary conditions (Davis *et al.*, 1991). However, the fact that the experiments are performed on samples that exhibit relatively little geometric variability often proves to be both an advantage (assessment of processes is not confused by other effects) and limitation in code testing (codes are not tested for natural heterogeneity). This type of testing does not allow evaluation of numerical techniques and coding with respect to geometric features present in the real world.

Some users prefer to rely on comparison with field experiments for establishing code credibility. These experiments may be specifically designed for research purposes (*e.g.*, *Mackay et al.*, 1986), or consist of a well-characterized existing system. Examples of the second approach are found, among others, in the efforts of Huyakorn *et al.* (1984a) and Frind and Hokkanen (1987) to simulate the movement of a well-monitored chloride plume in an aquifer subjected to highly-detailed investigations. Being able to simulate accurately phenomena observed in the field provides a convincing argument for the correctness of the code. However, poor results may not be indicative of code problems. Field experiments are often subject to significant uncertainty in parameter distribution, and initial and boundary conditions (Davis *et al.*, 1991). Furthermore, comparison with field experiments is subject to possible conceptual misunderstanding of field conditions. Also, published, well-controlled and monitored field experiments cover only a limited subset of the variety of conditions typically encountered in the field.

Hypothetical problems are often used to test certain computational features which are not represented in simple, analytical models (van der Heijde and Elnawawy, 1992). Such features may include irregular boundaries, time varying stresses (sources/sinks), heterogeneity and anisotropy in aquifer properties, and grid orientation and geometry. The synthetic or hypothetical system used for such a test is defined by synthetic system parameters, initial and boundary conditions and system stresses. As no independently observed system responses are available, testing takes place either by evaluating individual code behavior with respect to numerical consistency and stability, or by comparing the simulations made with the various codes, so-called “code-intercomparison.” An example of code intercomparison using synthetic data sets is given by Kinzelbach (1987a). He compared four two-dimensional solute transport codes based on different numerical solution techniques to a number of test cases using equal discretization of the space and time. The test cases were selected on basis of expected numerical problems with one or more of the numerical solution techniques.

2.1.2. Test Evaluation Techniques

Almost all code-evaluation studies, performed both by code developers and users, have utilized rather qualitative evaluation techniques that do not include a systematic approach to test problem design, test strategy, or evaluation procedures and measures (Beljin; 1988). It is often not clear from the code documentation if the performed tests stress or (inadvertently) hide unexpected problems or faulty code. In general, test objectives and evaluation criteria are absent. Presentation of test results is often limited to a graph showing a small number of control points in the space domain, often arranged along principle coordinate axes and using non-optimal graphing scales. In some cases, test problems are not presented as such but described as “example problems” (e.g., Contractor, 1981; Voss, 1984). This situation is very confusing to the users of code testing results.

2.1.3. Test Strategies

An important criterion, sometimes used to evaluate code testing efforts, is whether or not the tests address the major aspects, conditions and processes relevant for the intended use of the model (Davis *et al.*, 1991). To address the inadequateness and inconsistency of many code testing efforts, van der Heijde and Elnawawy (1992) recommended a systematic analysis of the code testing process and the development of a code testing and evaluation protocol.

In the mid 1980s the International Ground Water Modeling Center (IGWMC) developed a code testing strategy for ground-water models. Early versions of this strategy have been presented in van der Heijde *et al*. (1985), and applied by Huyakorn *et al*. (1984a) and Beljin (1988) to two-dimensional flow and solute transport codes. The objective of the IGWMC test strategy was to provide a framework for evaluation of a code using analytical solutions, hypothetical test problems, and field experiments. Special attention was given to the formulation of test objectives. However, as is the case with most test programs for ground-water modeling codes, the early version of the IGWMC code testing strategy did not address the completeness and effectiveness of the testing performed. To address this concern, van der Heijde *et al.* (1993) presented an expanded version of the IGWMC testing strategy. In this new version, three different code testing objectives are recognized: 1) functionality analysis; 2) performance evaluation; and 3) applicability assessment.

2.2. CODE TESTING ISSUES

Code testing in case of analytical models is a rather straightforward process. In general, code testing issues for this type of codes focus on the correct coding of the closed form solution and the input and output handling, and in case of a series approximation, also on the accuracy of the included terms and the domain for which the series has been defined. Code testing in case of numerical models is more complicated. Numerical modeling is based on finding approximate solutions for the governing equations. These approximations generally require discretization of the modeled space and time domains. The numerical solutions are given in the form of tables of numbers representing values of the dependent variable in the discretized domains.

2.2.1. Discretization Issues

The accuracy of the numerical solution is influenced by the resolution of spatial discretization (*i.e.*, grid size), the time discretization (*i.e.*, time-stepping), and the geometry of the discretized spatial elements or cells. If stability and convergence issues have been addressed, accurate numerical solutions can be generated using high resolution numerical grids, and small time steps. This approach is based on the principle that the smaller the discretization is in space and time, the better the approximate numerical solution will represent the real (unknown) solution of the governing partial differential equation (Huyakorn and Pinder, 1983).

Using simplified test problems provides modelers the opportunity to design and adjust the spatial and temporal discretization to optimally match a target analytical solution, if available. However, for test problems for which no analytical solution exist, the design of optimal discretization may require performing a sensitivity analysis of discretization resolution, refining grid and time-stepping till the simulation results are independent of cell sizes and time steps (Gupta *et al*., 1987). The importance of discretization considerations in code testing is illustrated by Gupta *et al*. (1987) for various test problems (see Fig. 2-3 and Fig. 2-4).

2.2.2. Numerical Algorithm and Computer Accuracy Problems

There are various possible sources of error in a computer code implementation of a numerical solution method. These errors are related to the approximation method, the solution algorithm, or the computer platform for which the code is compiled. For example, numerical problems are well-known in solving the advective-dispersive solute transport equation (Huyakorn and Pinder, 1983). They may also occur when modeling non-linear flow problems. The following section discusses some of these problems to highlight code performance test issues and to provide background information for the development of an effective code testing strategy.

Round-off error is the difference between the "true" representation and the machine representation of the dependent variable (Jain, 1984). The true representation refers to the complete mathematical description in the approximate formula. Roundoff errors in computer-based calculations occurs when using floating-point (real) numbers to represent parameters and variables (Press *et al*., 1992).

Truncation error is the difference between the true representation and the exact value of the variable (Jain, 1984). Truncation errors are considered algorithm errors and occur frequently because often the distribution of the unknown variable is represented by a truncated polynomial expansion. This error can be controlled by increasing the number of polynomial elements used to represent the distribution of the variable. However, such an approach often results in a significant increase in the complexity of the numerical solution.

The inherited error is a cumulative error promulgated through a sequence of computational steps. These errors not only influence the accuracy of the computations, but might also be the cause of stability or convergence problems. A method is stable if the effect of any single fixed round-off error is bounded, independent of the number of (discretization) mesh points (Jain, 1984). This means that both the truncation error is controlled and that the inherited error is not growing unchecked.

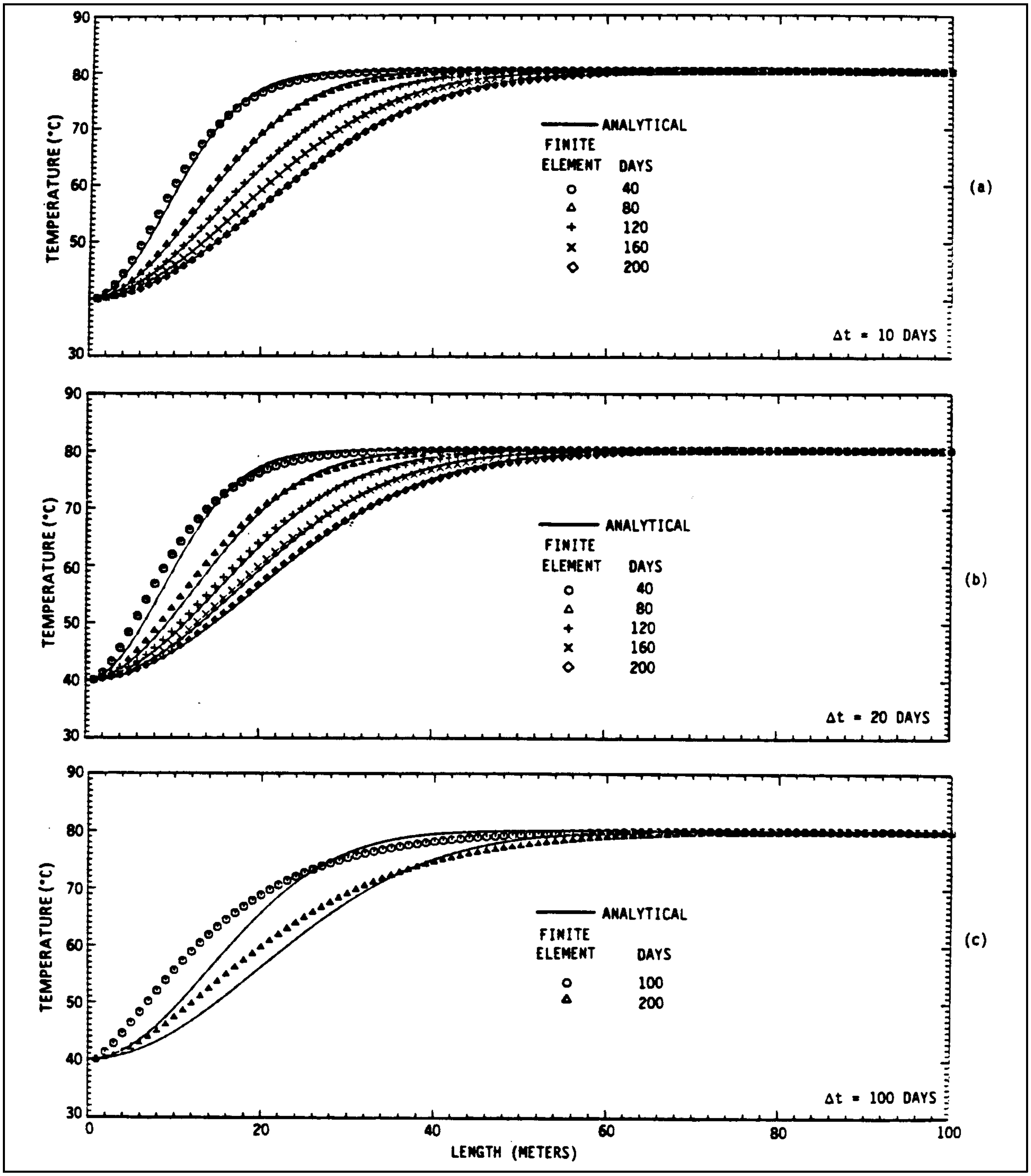


Figure 2-3. Comparison of the analytical and CFEST numerical solution of the radial Avdonin problem for heat transport for various time discretizations (from Gupta *et al.*, 1987).

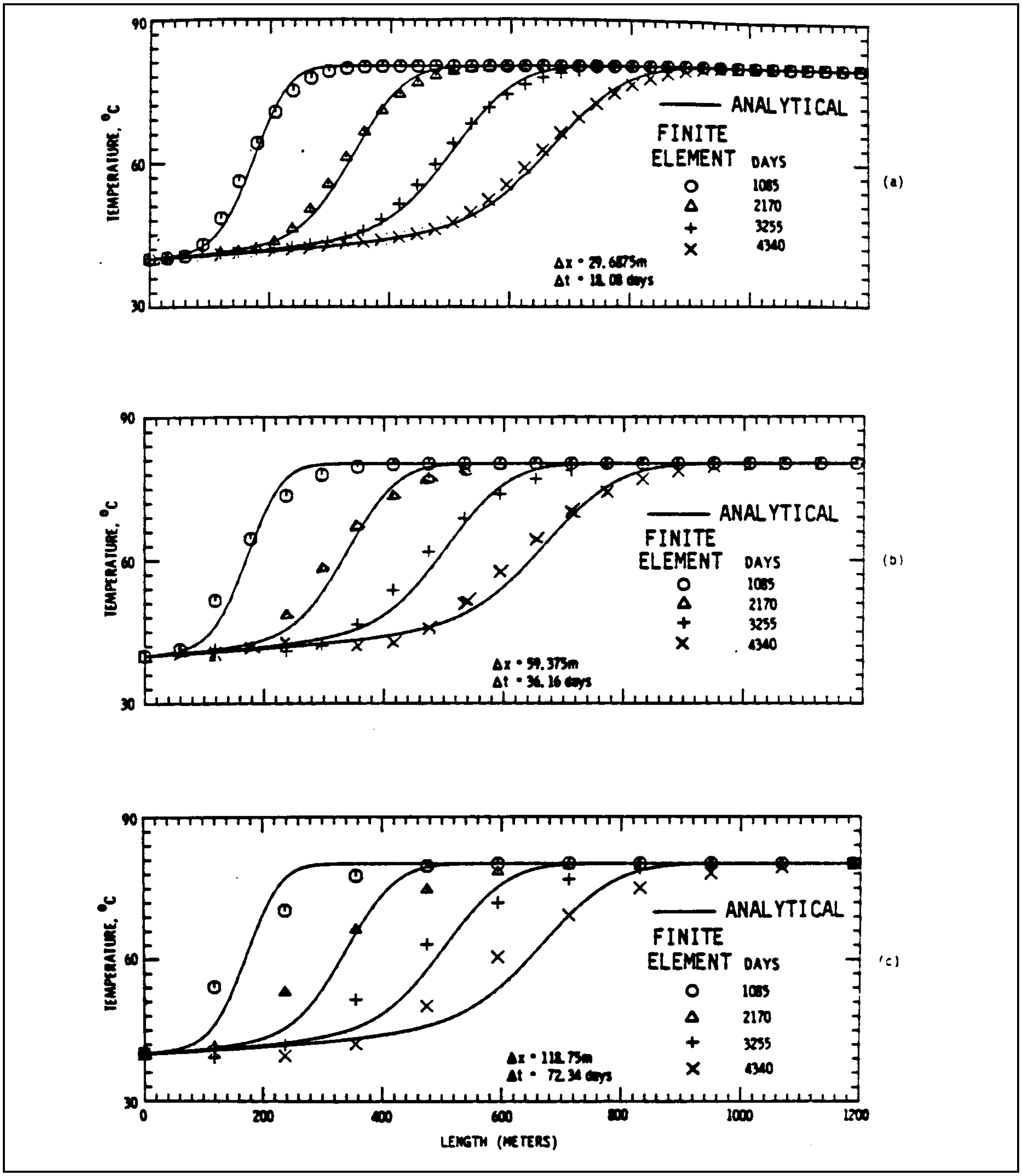


Figure 2-4. Comparison of the analytical and CFEST numerical solution of the linear Avdonin heat transport problem for various spatial discretizations (from Gupta *et al.*, 1987).

A numerical solution is said to converge if the differences between the analytical solution and the numerical solution decrease when the spatial and temporal discretization is refined (Jain, 1984). This assumes that a closed-form (analytical) solution to the governing partial differential equation exists, and the numerical solution approximates the analytical solution for the specific boundary conditions. If an analytical solution is not available, a numerical solution is considered converging if the differences between successive iterations decrease in a continuous manner.

Stability problems specifically occur in solving transient problems due to cumulative effects of the round-off error. An example of an unstable numerical solution is given in Figure 2-5. Stability analysis of the employed time-stepping scheme may provide an analytical representation of the stability constraint or stability condition. However, in many complex problems a simple criterion is not available (Huyakorn and Pinder, 1983). A typical ground-water situation, prone to stability problems, is the computation of a free surface using an explicit solution scheme which often leads to uncontrolled oscillations unless the time step is very small; another example is the simulation of fluid flow in dry soils (Huyakorn and Pinder, 1983). Furthermore, potential numerical instability can be encountered in the simulation of regions characterized by large contrasts in hydraulic conductivity in conjunction with high recharge rates (HYDROCOIN, 1988).

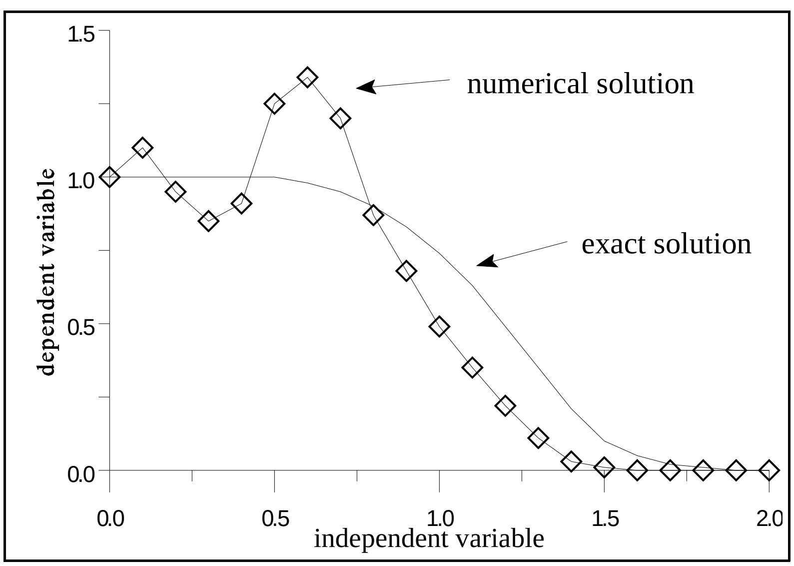


Figure 2-5. Numerical solution of the advective-dispersive solute transport equation exhibiting instability (after Peaceman, 1977).

Convergence is directly related to stability of the solution scheme (Jain, 1984). Often, the issue for ground-water modeling is not convergence versus non-convergence, but the rate of convergence, as this entity determines the computational time required to solve a particular problem (see Fig. 2-6). Some modeled systems are particular convergence-sensitive to parameter and discretization choices, such as the computation of a free surface in a water-table aquifer, leakage in an aquifer-aquitard system, the position of the interface between salt and fresh water in a coastal aquifer, and flow under highly nonlinear unsaturated conditions (Huyakorn and Pinder, 1983). For example, the HYDROCOIN code intercomparison study concluded (HYDROCOIN, 1988) that for some combinations of numerical method, matrix solver and discretization, large permeability contrasts in vadose zone flow modeling can result in a discontinuous moisture content distribution in the model domain, causing instability and non-convergence.

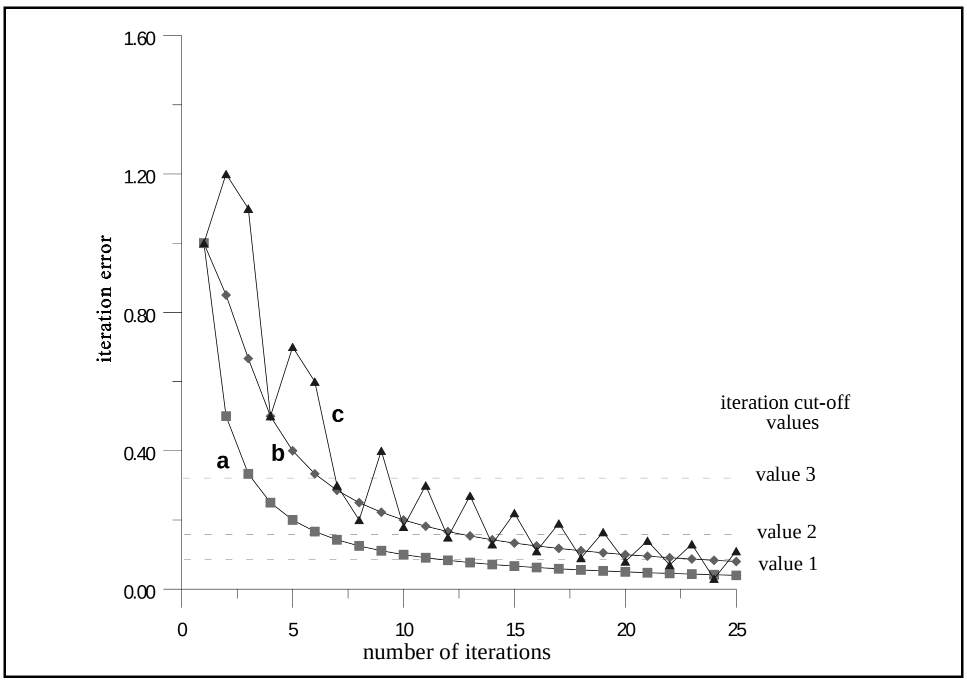


Figure 2-6a. Various types of converging solutions: a) high convergence rate; b) moderate convergence rate; c) oscillatory convergence.

It should be noted that in the case of oscillatory non-convergence behavior the iteration cutoff criterion can still be met (see Fig. 2-6b, both curve a and curve b). However, if the criterion is too strict, convergence might not be reached due to roundoff errors (see Fig. 2-6a, curve b and cutoff value 2), at least not within the specified maximum number of iterations. During code testing the iteration behavior should be analyzed for optimal accuracy. Figure 2-6a shows that curve ‘a’ reaches iteration criterion 3 very rapidly. However, this curve still steeply descends. It is better to choose a cutoff value in the less steep segment of the iteration curve, for example value 2. This might force the code tester to rerun the test problem a few times if the initial cutoff value is too high. Also, quite often the first few iterations result in an increase of iteration error, or in rapid variations between positive and negative errors as is illustrated in Fig. 6a, curve ‘c’ (*e.g.*, Andersen, 1993, p.13-9 &13-10).

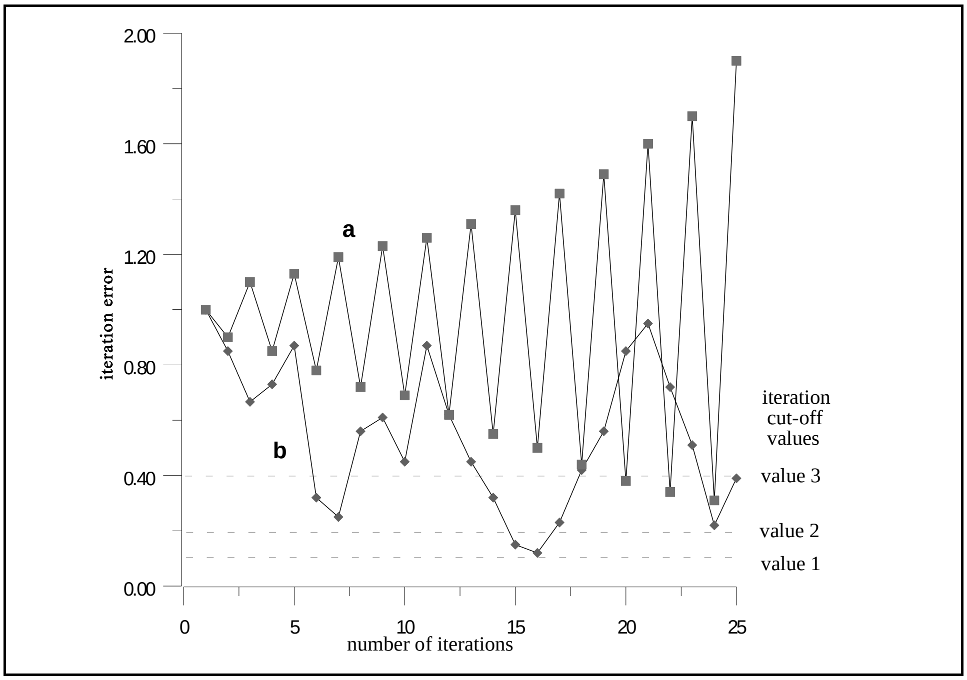


Figure 2-6b. Two types of non-converging solutions:
a) uncontrolled oscillations; b) limited oscillations.

Another problem is that although a stable solution might be achieved by using a specific solution method, that solution might be less accurate. An example is the use of a “consistent” mass matrix versus a “lumped” mass matrix in solving the flow equations in a system of saturated and unsaturated soils (Huyakorn and Pinder, 1983). Although theoretical evaluation of convergence and stability is often possible (Jain, 1984; Milne, 1970), such an analysis might be complex and impractical. Trial runs for a range of parameter combinations provide an effective alternative method of testing for such numerical problems.

As mentioned before, transport simulation models are prone to some specific algorithm problems: numerical dispersion and oscillations, specifically in advection-dominated problems where the transport equation is hyperbolic. Numerical dispersion is an artificial front-smearing effect of the computational scheme which resembles and may be indistinguishable from the effect of the actual physical dispersion mechanism (Huyakorn and Pinder 1983). Spatial oscillations (overshoot and undershoot) occur often near a concentration front, especially under advection-dominated transport conditions. Overshoot occurs when upstream of the moving front erroneous high values of concentrations are computed, while undershoot describes the analog phenomenon downstream of the front, sometimes resulting in negative concentrations. These oscillations are typically controlled by the Peclet number ($Pe=V*\Delta s/D$ with V=velocity, Δs=characteristic length, and D=dispersion coefficient) and the Courant number ($Cr=V*\Delta t/\Delta s$, where Δt is the time step size).

Because of the difficulties encountered in the numerical solution of the advection-dominated transport equation, it is important to use the mass balance as a check on the acceptability of the numerical solution. The mass balance for both the flow and transport solution of the transport problem should be evaluated for each test simulation run. As mass balance errors are a function of discretization and the iteration cut-off value, among others, each numerical simulation code should include the option to calculate such global mass balances.

As many solute transport problems in ground water are convection dominated, numerical methods specifically developed for hyperbolic partial differential equations are popular, such as the method of characteristics and the random walk method (Kinzelbach, 1987b). To use these particle tracking methods, specific application requirements need to be satisfied. For example, it is important to limit the distance traveled by individual particles to a fraction of the cell spacing to fulfill the Courant criterion. The random walk method is based on the theorem that, in the limit of large particle numbers and assuming that the dispersivities are space independent, the random walk analog represents the advective and dispersive components of the transport equation (Kinzelbach, 1987b). A relatively large number of particles are needed in this method to obtain reasonable results. Furthermore, many simulation codes based on random walk method assume a gradually changing flow field. If this is not appropriate, for example at stagnation points, the dispersion derivatives in the convection term of the transport equation should be included. If in layered aquifers the particles are transported close to or across layers, other techniques need to be incorporated such as boundary reflection, or buffering (Kinzelbach, 1987b). Another issue with particle tracking methods is the manner in which particles are considered captured by a pumping well. Often this is done by defining a circular capture zone around the well with a user-specified or hard-wired radius. The combination of capture radius and maximum time-step travel distance has a major influence on the accuracy of the breakthrough curve.

Underlying particle tracking methods is the notion that if the time-varying flow field is known, unique pathlines exist between (almost) any two points in the model domain (except for pathlines starting in a singularity). In practice, exact determination of pathlines is only possible in a limited number of simplified situations. Therefore, pathlines determination requires numerical integration of the velocity-based pathline equations with respect to time, so-called forwards pathline tracking (Kinzelbach, 1987b). Inversion of the pathline equations results in backwards pathline tracking. Travel times can be determined by integration along the pathline. Particle tracking methods use this approximate approach to simulate advective transport or the advective part of advective-dispersive transport.

The numerical integration of the pathline equation provides a source for inaccuracies in modeling. The velocity at the particle location is obtained through interpolation. Various methods exist, among others, dependent on the use of the finite element or finite difference method in determining the head distribution (*e.g.*, Konikow and Bredehoeft, 1978; Prickett et al., *1981*; Shafer, 1987; Pollock, 1989; Zheng, 1989; Franz and Guiguer, 1990; Goode, 1990). As particle tracking is widely used in the study of ground-water protection and contamination problems, testing the coded integration and interpolation algorithms is an essential part of code development quality assurance. In general, testing of particle tracking simulation codes requires finer grids than the grids required by codes which calculate hydraulic heads, velocities or contaminant concentrations directly from the governing equations (HYDROCOIN, 1988). The HYDROCOIN study (1988) concluded that vector quantities (fluxes, velocity field and trajectory pathlines) show larger discrepancies than scalar quantities (pressures, heads) when compared to reference solutions when calculated by integrating velocities with post-head-simulation algorithms (Nicholson *et al.*, 1987)(see Figure 2-7a and 2-7b). Testing pathline algorithms in numerical simulation code is often performed by comparing the results with analytical expressions for the stream function (*e.g.*, HYDROCOIN level 3, test case 7).

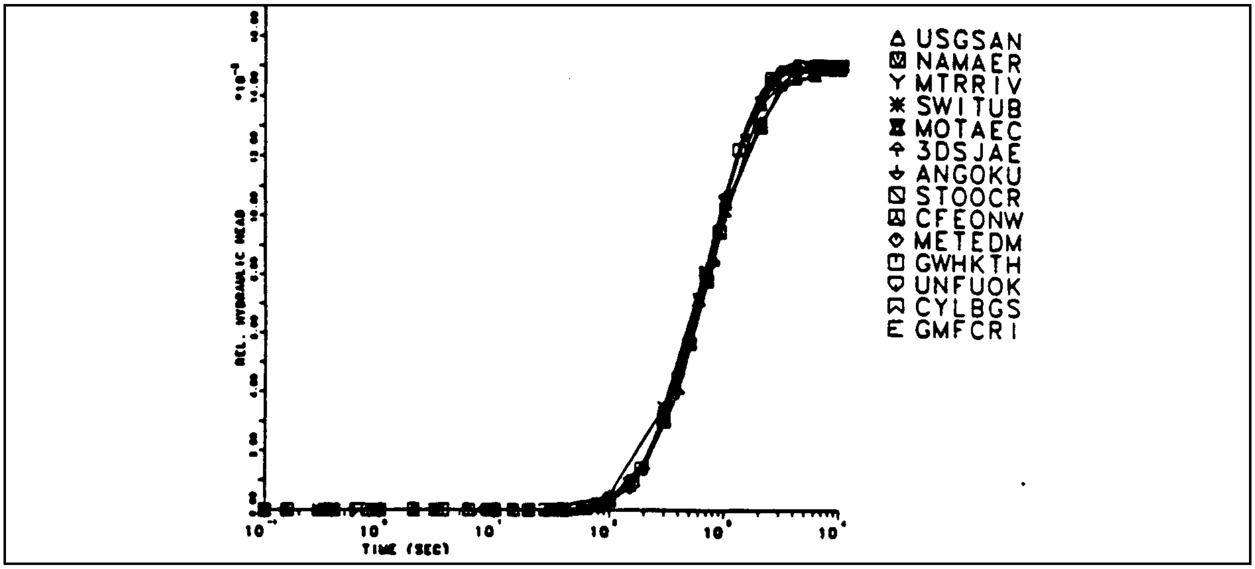


Figure 2-7a. HYDROCOIN, level 1, case 1: relative hydraulic head computed by various simulation codes (from HYDROCOIN, 1986).

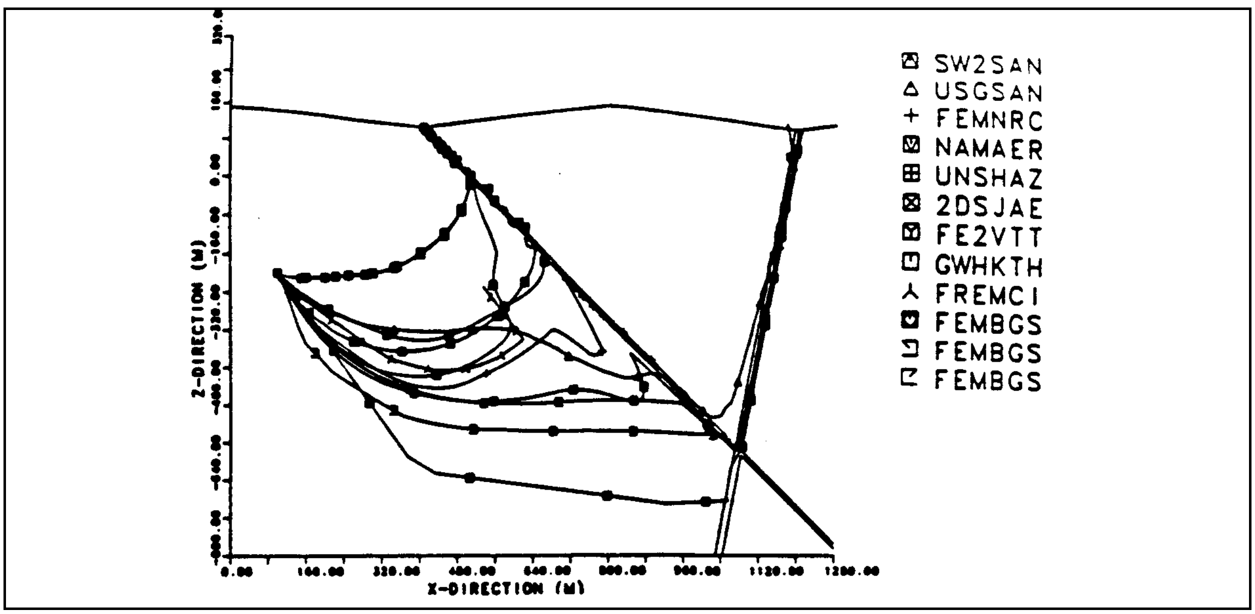


Figure 2-7b. HYDROCOIN, level 1, case 2: pathline trajectories computed by various simulation codes using a coarse mesh (from HYDROCOIN, 1986).

2.3. CODE TEST CASES

Two of the most elaborate ground-water simulation code testing projects were conducted in the 1980s (INTRACOIN, 1984; HYDROCOIN, 1988,1990). The test problems, designed during these studies, can be divided in three groups: 1) physical processes-oriented tests; 2) field characteristics-oriented tests; and 3) code feature(s)-oriented tests (*e.g.*, efficiency and accuracy of particle tracking algorithms). In the INTRACOIN study (1984), participating code-testing teams were asked to provide five computed items to facilitate accuracy and performance intercomparison: 1) constituent concentrations at the end of the migration path with respect to time (*i.e.*, breakthrough curves); 2) the maximum constituent concentration and the time at which this maximum concentration is reached; 3) determination of the value of half of the maximum constituent concentration and the time at which this value is reached; 4) the total CPU time required when executed on a standard computer; and 5) the time required for one-single precision floating point multiplication to be completed when executed on a standard reference computer. It should be noted that there were no requirements with respect to spatial or temporal distribution of results nor calculation of quantitative measures for intercomparison. In other comparison studies, spatial distribution of computed variables, breakthrough curves, and mass balances were the focus of the intercomparison (*e.g.*, Kinzelbach, 1987b; Beljin, 1988). Lobo Ferreira (1988) specifically included problem set up, CPU time, and computer resource use.

In a follow-up project to INTRACOIN, various simulation codes were tested for a variety of geological conditions (HYDROCOIN, 1988). Dependent on the test problem, intercomparison variables included: 1) hydraulic heads and pressures; 2) salt concentrations; 3) temperatures; 4) flow velocities; 5) flow pathlines; 6) travel times; 7) flow rates (fluxes); and 8) location of the water-table. Furthermore, mass balance errors and flux distributions were computed and compared. It was concluded that to be able to perform code intercomparison, test problems should be well-defined and bounded, and provided level of detail of the problem description should restrict the ability of the different code testing teams to provide their own interpretation for model setup. Input parameterization, discretization of space and time, and implementation of boundary conditions should be consistent, specifically in testing against field data sets and complex hypothetical test problems.

Another issue is that often the analytical solution of a test problem requires assumptions with respect to the modeled processes, geometry of the model domain, and boundary conditions which cannot be exactly met in the numerical solution. In some cases, this issue can be addressed by careful selection of the boundary of the model domain and the spatial and temporal discretization for the numerical solution. An example is the requirement of an infinite aquifer extent in the Theis solution (Theis, 1935). This requirement is replaced in the numerical simulation by a boundary at a large distance from the well and ensuring that the time domain does not allow significant drawdown at the boundary (*e.g.*, Gupta *et al.*, 1987). In other cases, differences between the analytical and numerical solution result which cannot be removed completely. An example is the inability of some codes to accept zero values for transverse dispersion when simulating one-dimensional solute transport test cases.

As mentioned earlier, the International Ground Water Modeling Center (IGWMC) has developed a three-level simulation code testing approach (van der Heijde *et al.*, 1985; van der Heijde and Elnawawy, 1992). At Level 1, the code is tested by comparing simulation results against an analytical solution. At Level 2, synthetic data sets are used as the basis for code intercomparison. These data sets are developed using hypothetical problems for which no independent benchmark exists. At Level 3, the code is used to simulate well-characterized and monitored laboratory or field experiments. One of the first applications of the IGWMC code testing approach was performed by Huyakorn *et al.* (1984b). They implemented the IGWMC procedure in testing the two-dimensional finite-element flow and transport code, SEFTRAN (Huyakorn *et al.*, 1984b). Six Level 1 test problems were used to evaluate the transport simulation capabilities of this finite-element code. A realistic range of flow and transport parameters was chosen to analyze the numerical behavior of the code under various potential application conditions. The six Level 1 problems ranged in complexity from simple one-dimensional transport in a uniform flow field to transport in a nonuniform two-dimensional flow field created by a recharging-discharging well pair. Each of the problems was defined by detailed problem statements which included input specifications, spatial and temporal discretization procedures, and simulation results. The problem statement included test objectives, a discussion of field situations for which the simplified analytical solutions may be applicable, and the analytical solution.

The Huyakorn *et al.* (1984a) study also presented two Level 2 test problems, based on hypothetical field situations. They were characterized by irregular geometry, complex boundary conditions and heterogeneous, anisotropic aquifer conditions. The first Level 2 test problem involved a cross-sectional analysis of contaminant transport in an unconfined aquifer with steady-state flow and sharply contrasting physical properties, located underneath a landfill (see Figure 2-8 and table 2-1). Evaluation of the results, in the absence of a second simulation code for intercomparison, was limited to a qualitative discussion of the appropriateness of the head and concentration distributions. The second case involved areal analysis of contaminant transport released from a constant-head disposal pit into a confined aquifer subject to pumping (see Figure 2-9 and table 2-2). In each case, realistic physical conditions and practical values of aquifer parameters were used in order to develop a meaningful interpretation of the behavior of the hypothetical system. This, in turn, provided the basis for qualitatively assessing the behavior of the simulation code.

Huyakorn *et al.* (1984a) also performed a partial Level 3 benchmark test by using the field data set that describes the movement of a chloride plume at a landfill located at the Canadian Forces Base Borden in Ontario, Canada. At the time of the code testing study by Huyakorn *et al.* this site had been studied extensively and detailed information regarding hydrogeological characteristics and plume movement had been published. Furthermore, various modeling studies had been performed previously and their results were available for intercomparison. As this field problem is three-dimensional in nature, Huyakorn *et al.* (1984a) used the two-dimensional SEFTRAN code in both the profile mode and the areal (planar) mode. Due to differences in saturated thickness, effective transmissivity in the horizontal simulations was divided in a number of zones. The report is not clear regarding the results of the flow and transport simulations. Qualitative statements, such as "Predicted chloride concentrations are generally in good agreement with observed concentrations presented in Figure...., although downgradient concentrations are slightly high." are not supported by tables or comparative line graphs. This test problem is often referred to as the "Waterloo field verification problem." Due to the three-dimensional nature of the hydrogeology and the plume movement, it is not well-suited for testing models in a two-dimensional profile simulation mode. However, it provides an excellent test case for three-dimensional saturated flow and transport codes. It should be noted that the chloride plume movement at the Borden landfill has been the prototype for the two- and three-dimensional versions of the "Waterloo Test Problem,"a synthetic test data set with the same geometry and comparable boundary conditions as encountered at the Borden site. This test is often used to evaluate a code's capability to simulate solute transport from a source on the top boundary through an aquifer with various types of discontinuities and parameter distributions (Segol, 1994).

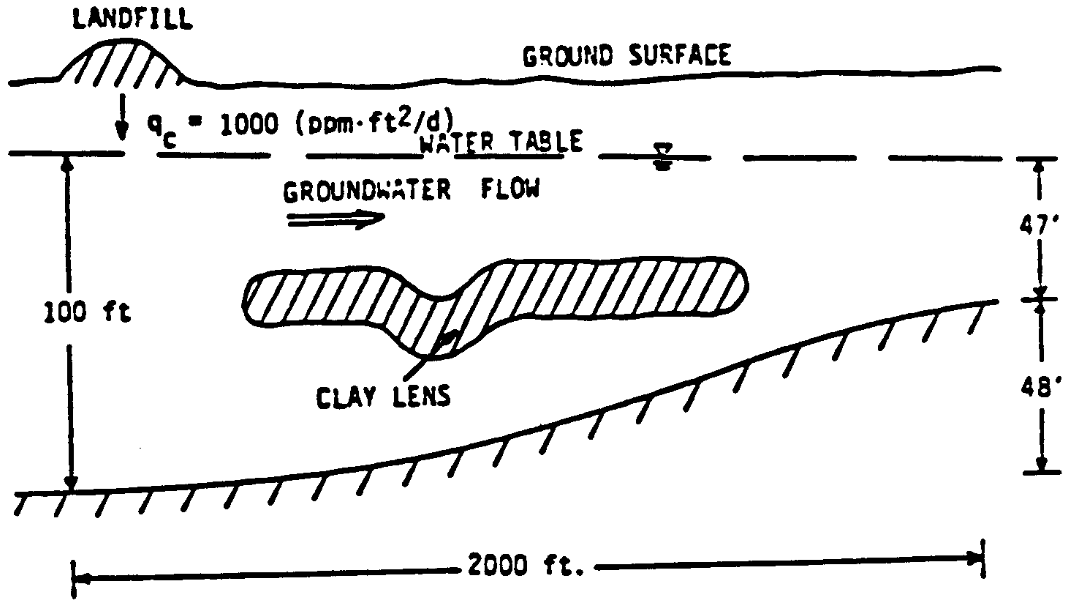


Figure 2-8. Schematic depiction of Level 2, case 1 (after Huyakorn et al., 1984a)

The IGWMC evaluation approach was also applied by Beljin (1988) using three simulation codes: SEFTRAN (Huyakorn *et al.*, 1984b), USGS-2D-MOC (Konikow and Bredehoeft, 1978) and RANDOM WALK (Prickett *et al.*, 1981). The numerical accuracies of the various algorithms were measured by comparing the simulation results to the results of five analytical solute transport solutions (Level 1 benchmark solutions). Code sensitivities to various parameters and to time and space discretization schemes were also evaluated. Five of the six Level 1 test problems formulated by Huyakorn *et al.* (1984a) were used. The agreement between the simulated and benchmark analytical solutions were assessed qualitatively and quantitatively. The report described the results using five qualitative categories: "poor," "reasonable," "acceptable," "good," and "very good." Quantitatively, the results were expressed by the root-mean-squared error between the values of contaminant concentration calculated by the code and by the analytical model.

Table 2-1. Values of physical parameters for Level 2, case 1 (from Huyakorn *et al.*, 1984a)

Parameter	Value
Aquifer properties	
Hydraulic conductivity, K_{xx}	10 ft/d
Hydraulic conductivity, K_{yy}	5 ft/d
Hydraulic conductivity, K_{xy}	2 ft/d
Porosity	0,25
Longitudinal dispersivity	200 ft
Transverse dispersivity	50 ft
Clay lens properties	
Hydraulic conductivity	0.002 ft/d
Porosity	0.45
Longitudinal dispersivity	100 ft
Transverse dispersivity	20 ft

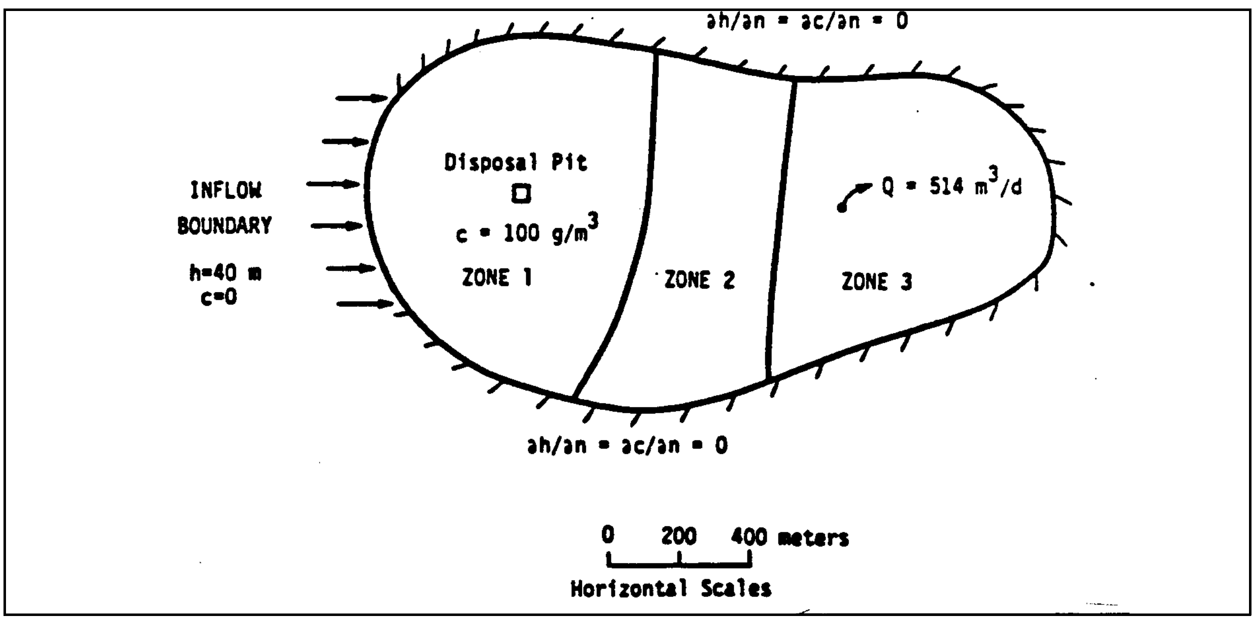


Figure 2-9. Schematic depiction of Level 2, case 2 (after Huyakorn et al., 1994a)

Table 2-2. Values of physical parameters for Level 2, case 2 (from Huyakorn *et al.*, 1984a)

Parameter	Value
Aquifer zone 1 properties	
Transmissivity, T_{xx}	100 m^2/d
Transmissivity, T_{yy}	50 m^2/d
Storage coefficient	0.02
Porosity	0,25
Aquifer thickness	8 m
Longitudinal dispersivity	50 m
Transverse dispersivity	15 m
Aquifer zone 2 properties	
Transmissivity, T_{xx}	200 m^2/d
Transmissivity, T_{yy}	100 m^2/d
Storage coefficient	0.01
Porosity	0,20
Aquifer thickness	78.75 m
Longitudinal dispersivity	75 m
Transverse dispersivity	30 m
Aquifer zone 3 properties	
Transmissivity, T_{xx}	400 m^2/d
Transmissivity, T_{yy}	250 m^2/d
Storage coefficient	0.04
Porosity	0,20
Aquifer thickness	40 m
Longitudinal dispersivity	40 m
Transverse dispersivity	10 m

Another perspective to code testing can be found in following the verification history of the U.S. Geological Survey Modular Three-Dimensional Finite Difference Ground-Water Flow Model (MODFLOW). MODFLOW was first published in 1984 documenting the FORTRAN 66 implementation (McDonald and Harbaugh, 1984), and is based on the theoretical framework of earlier, well-established and verified finite difference flow models (Trescott, 1975; Trescott and Larson, 1976; Trescott *et al.*, 1976). In 1988, a new version of the MODFLOW was published (McDonald and Harbaugh, 1988), documenting the FORTRAN 77 implementation of the code. Although many test problems have been used during the development of the MODFLOW code, none of these problems were discussed in the published documentation, which only provides a single sample application (McDonald and Harbaugh, 1988, Appendix D). Despite the lack in formal verification, MODFLOW has become a widely-used and accepted simulation code, to the extent that verification of other codes often includes intercomparison with MODFLOW results. In part, this might be the result of the acceptance by the ground-water community of the simulation codes from which MODFLOW originated. Also, the many successful applications to practical problems have contributed to its credibility. See for example the many applications cited by Anderson and Woessner (1992) and the USGS Regional Aquifer System Analysis studies (Weeks and Sun, 1987).

MODFLOW is a finite difference code for steady-state and transient simulation of two-dimensional, quasi-three-dimensional, and fully three-dimensional saturated, constant density flow problems in combinations of confined and unconfined aquifer-aquitard systems above an impermeable base (McDonald and Harbaugh, 1988). The simulated flow processes are described by the governing partial differential equation which includes anisotropy, but does not include cross-terms for hydraulic conductivity. Porous media heterogeneity is introduced during the formulation of the finite difference equations, as are the various source and sink terms, and flow and head boundary conditions. Additional capabilities are handled by specifically designed solution algorithms, such as time varying stresses using stress periods, preparation of conductance terms, mass balance calculations, and dewatering/rewetting of cells. Other features are simulated by careful manipulation of boundary conditions and code functions (*e.g.*, water table position and seepage face position; Anderson and Woessner, 1992).

The MODFLOW program consists of a main program (MAIN) and a large number of subroutines, called modules. These modules are grouped into “packages.” The “standard” MODFLOW program (McDonald and Harbaugh, 1988) includes 10 packages (see Table 2-3).

Table 2-3. List of MODFLOW packages (from McDonald and Harbaugh, 1988).

Package abbreviation	Package name	Description
BAS	Basic	Handles those tasks that are part of the model as a whole. Among those tasks are specification of boundaries, determination of time-step length, establishment of initial conditions, and printing of results.
BCF	Block-Centered Flow	Calculates terms of finite difference equations which represents flow within porous medium; specifically flow from cell to cell and flow into storage
WEL	Well	Adds terms representing flow to wells to the finite difference equations
RCH	Recharge	Adds terms representing areally distributed recharge to the finite difference equations
RIV	River	Adds terms representing flow to rivers to the finite difference equations
DRN	Drain	Adds terms representing flow to drains to the finite difference equations
EVT	Evapotranspiration	Adds terms representing evapotranspiration to the finite difference equations
GHB	General-Head Boundary	Adds terms representing general-head boundaries to the finite difference equations
SIP	Strongly Implicit Procedure	Iteratively solves the system of finite difference equations using the Strongly Implicit procedure
SOR	Slice-Successive Overrelaxation	Iteratively solves the system of finite difference equations using the Slice-Successive Overrelaxation technique

Testing of the many features of a code with the complexity of MODFLOW requires a carefully designed testing strategy. Andersen (1993) presented a comprehensive set of problems designed for self-study in ground-water modeling using the MODFLOW code. In addition to its educational objective, the author intended the problems to serve in the verification of the code. To this purpose, where possible, MODFLOW results were compared to analytical solutions (benchmarking), results of other models (intercomparison), or to itself using alternative input functions to represent the same problem feature (intracomparison). The set of twenty instructional and verification problems has been used to verify the operation of other codes (*e.g.*, Larson and Esling, 1993; Dendrou and Dendrou, 1994), as well as the correct installation and compilation of the MODFLOW code on computer platforms different from the platform where these test problems were developed. Andersen (1993) presents two tables summarizing the performed tests. In the first table an overview is given of the test problems and the type of verification that the tests represent (see Table 2-4). The second table shows which MODFLOW packages have been used in each tests (see Table 2-5). From this latter overview it appears that each program package is used at least twice in the series of tests.

The manual by Andersen (1993) does not provide an overview of the code functions which have been used in the testing. However, review of the problem descriptions shows that most code features have been addressed (see Table 2-4). Furthermore, the results of the verification exercises are highly dependent on grid design, time-stepping, representation of boundary conditions, choice of numerical parameters, and selection of appropriate code options. A table, comparable with table 2-5, listing features versus test problem would be highly useful for verification analysis purposes.

2.4. DISCUSSION

In 1992, van der Heijde and Elnawawy (1992) identified the need to complement the three-level code testing approach with a systematic procedure for the design and use of test problems allowing code testers and reviewers to judge the completeness of the performed testing in terms of: 1) code "reliability" (*e.g.*, stability and reproducibility of solution algorithms); 2) the efficiency of coded algorithms and input/output data transfers (*e.g.*, code performance in terms of numerical accuracy versus time of computation, memory use and storage requirements); 3) the amount of required preparation resources (*e.g.*, data preparation and output data reduction and analysis time); and 4) the sensitivity of the simulation code to grid design, simulation processes, boundary conditions, and to a wide variety of input parameter values. Such a testing procedure should alleviate the problem that in most code testing exercises conducted in the past, only a very limited number of code functions and operational conditions have actually been addressed. It should contain elements of earlier studies which were judged to be useful as well as new components addressing code-evaluation deficiencies. Issues which should be addressed in the protocol include:

Table 2-4. MODFLOW test problems and type of testing (after Andersen, 1993).

Problem No.	Description	Analytical or semi-analytical solution	Inter-comparison with another numerical model	Alternate boundary condition or model configuration
1	Transient radial flow to a well (Theis, 1935)	X		
2	Transient radial flow to a well with horizontal anisotropy (Papadopulos, 1965)	X		
3	Transient radial flow to a well with confined-unconfined condition conversion (Moench and Prickett, 1972)	X		
4	Steady-state flow in a square, single layer, model domain with fixed-head and no-flow boundaries and a pumping well; calculation of head			X[1]
5	Steady-state flow in a square, single layer, model domain with fixed-head and no-flow boundaries and a pumping well; calculation of mass balance			X
6	Steady-state flow in a square, single layer, model domain with fixed-head and no-flow boundaries with uniform recharge; similarity solutions in model calibration for transmissivity and recharge			
7	Steady-state flow in a square, single layer, model domain with fixed-head and no-flow boundaries; with or without uniform recharge and/or a pumping well			X
8	Steady-state flow in a square, single layer, model domain with fixed-head and no-flow boundaries and a pumping well; grid and time stepping considerations (Rushton and Tomlinson, 1977)	X	X FE method	
9	Steady-state flow in a square, single layer, model domain with an internal stream (leaky boundary nodes) and no-flow lateral boundaries; stresses include uniform recharge and a pumping well; calibration and prediction exercise			
10	Transient, one-dimensional horizontal flow resulting from variations in areal recharge; model domain bounded by a constant head and a no-flow boundary; transient calibration of recharge			

11	Transient flow in a one-dimensional vertical model of two aquifers separated by an aquitard; representation of aquitards implicitly as a leakage term or implicitly as a separate model layer.			
12	Transient flow in a aquifer-aquitard system with a fully-penetrating well in the aquifer (Hantush, 1960)	X	X FE method	
13	Steady state flow in a three-layer, heterogeneous, single aquifer system with square model domain, partially recharge in top layer, fixed head at one boundary and no-flow at other boundaries; solution technique (SIP/SSOR) and convergence			
14	Steady state flow in a single-layer, unconfined aquifer with square model domain, pumping from a well, internal head-dependent flux nodes and no-flow at all boundaries; internal third-type boundary represented using river package, as a general-head boundary, as a drain, as a line of ET nodes, and as a two-layer system			X (5 different implemen-tations of 3rd type b.c.
15	Steady-state, one-dimensional flow system resulting from two fixed-head boundaries, intersected by a drain			X
16	Steady state flow in a homogeneous aquifer with sloping base and rectangular model domain; uniform areal recharge and spatially varying ET; fixed head at one boundary and no-flow at other boundaries		X FD method	
17	Transient radial flow to partially penetrating and multi-layer screened wells in a stratified aquifer represented by a multi-layer model			
18	Steady-state cross-sectional simulation of steep head gradients in stratified, uniformly recharged, unconfined aquifer with highly variable thickness, layer pinchout, and sloping beddings; model domain is laterally bounded by a no flow boundary and a specified head boundary.			
19	Transient flow in a real world, single aquifer system (Musquodoboit Harbor Aquifer, Nova Scotia) subject to various planned pumping regimes		X FE and FD methods	
20	Transient flow in a real world, single aquifer system (Lipari Landfill, New Jersey) subject to various hydrologic control options in remedial design		X FD method	

1) This problem can be approximated using the Thiem (1906) solution and image theory to represent the boundaries.

Table 2-5. MODFLOW test problems and packages used in tests (after Andersen, 1993).

Problem No.	BAS	BCF	WEL	RCH	RIV	DRN	EVT	GHB	SIP	SOR
1	X	X	X						X	
2	X	X	X						X	
3	X	X	X						X	
4	X	X	X	X					X	
5	X	X		X					X	
6	X	X	X	X					X	
7	X	X	X	X					X	
8	X	X	X					X	X	
9	X	X	X	X	X				X	
10	X	X		X						X
11	X	X							X	
12	X	X	X						X	
13	X	X		X					X	X
14	X	X	X		X	X	X	X	X	
15	X	X				X			X	
16	X	X	X	X			X		X	
17	X	X	X						X	
18	X	X		X					X	X
19	X	X	X		X				X	
20	X	X		X		X			X	

- address complex problem descriptions and modeling issues (*e.g.*, heterogeneity, anisotropy, irregular boundary conditions);
- incorporate successful, previously defined test cases;
- use test cases which are not subject to ambiguous or unstable boundary conditions and/or ambiguous numerical implementations;
- use test cases that are clearly designed to meet specific objectives;

- develop standard, unbiased, accuracy and evaluation measures;
- avoid the use of secondary quantities (*e.g.*, trajectory pathlines that may be calculated by a post-processor) in simulation code-evaluation;
- be able to address problems encountered in previous studies, specifically as related to spatial and temporal discretization issues; and
- establish and incorporate test cases that represent realistic scenarios, rather than hypothetical cases that have no bearing on real-world conditions.

The development of a standardized, unbiased, systematic code-testing and evaluation program that incorporates these measures and approaches should significantly increase the QA of results generated by simulation codes. The availability of standard code-evaluation results should help remove ambiguity regarding their performance and operation and increase their acceptance by project managers and regulators.

Extensive code testing is typically based on one or more of six test approaches: 1) benchmarking; 2) comparison with controlled laboratory experiments or analogs; 3) comparison with controlled field experiments; 4) code intercomparison; 5) code intracomparison; and 6) field comparison or field demonstration. A comprehensive test strategy should include these approaches.

Almost all previous code-evaluation studies have utilized rather qualitative evaluation techniques that lack a systematic approach to formulation of a test strategy, test problem design, and evaluation procedures and measures. Test objectives and evaluation criteria (*i.e.*, performance targets) are often absent. As this situation is very confusing to users of test results, the new protocol should address this problem.

The potential problems in solving the flow and transport equations numerically make it necessary to allow for a flexible, code-type specific test strategy that needs to address such issues as stability, accuracy, convergence, and roundoff errors. Evaluation techniques should be both quantitatively and qualitatively in nature, and should include an assessment of the dependent variable in space and time as well as indirectly derived entities such as the global mass balance for flow and transport, flux distributions, pathlines, and travel times.

3. CODE TESTING AND EVALUATION PROTOCOL

3.1. OVERVIEW

A systematic approach to code testing combines elements of error-detection, evaluation of the operational characteristics of the code, and assessment of its suitability to solve certain types of management problems, with well-designed test problems, carefully selected test data sets, and informative performance measures. Such a systematic approach is represented by the *functionality analysis, performance evaluation and applicability assessment protocol*, developed by the International Ground Water Modeling Center (van der Heijde *et al.*, 1993). In this protocol, systematic development of test objectives is combined with a comprehensive code testing strategy. Test results are expressed in terms of correctness (*e.g.*, in comparison with a benchmark), reliability (*e.g.*, reproducibility of results, convergence and stability of solution algorithms, and absence of terminal failures), efficiency of coded algorithms (in terms of achieved numerical accuracy versus memory requirements and code execution time), and resources required for model setup (*e.g.*, input preparation time). The protocol consists of a number of sequential steps (see Fig. 3-1): 1) analyze the code's functionality; 2) identify potential problem areas; 3) develop a code testing strategy; 4) execute tests and analyze results; 5) prepare overview tables of results; 6) identify performance problems; 7) document findings; and 8) communicate results. In the following sections, each of these steps will be discussed in detail.

The main issue in reviewing previous code testing studies appears to be the lack in systematically addressing code features and providing insight in the completeness and effectiveness of the performed testing. Another major issue is the inconsistency and incompleteness of code documentation in describing the code's functions and features. The new code testing protocol addresses these deficiencies by defining three code testing components, systematically addressing these components in a test strategy, and reporting the test results using test matrices and tables. The three main components of this protocol are: 1) functionality analysis; 2) performance evaluation; and 3) applicability assessment. Functionality analysis is a rather qualitative process in contrast to performance evaluation, which is a quantitative process. While evaluating the functionality and performance of a code, its usefulness in addressing field problems is assessed in a qualitative manner.

Functionality analysis involves the identification and description of the functions of a simulation code in terms of model framework geometry, simulated processes, boundary conditions, and analytical capabilities (see Table 3-1 for an example of code functions), and the subsequent evaluation of each code function or group of functions for conceptual and computational correctness and consistency. The information generated by functionality analysis is organized into a summary structure, or matrix, that brings together the description of code functionality, code-evaluation status, and appropriate test problems. This *functionality matrix* is formulated combining a complete description of the code functions and features with the objectives of carefully selected test problems (see Table 3-2). The functionality matrix provides a quick way to illustrate or check the extent of the performed functionality analysis.

CODE TESTING AND EVALUATION PROTOCOL

step 1	analyze the code documentation with respect to simulation functions, operational features, mathematical framework, and software implementation;
step 2	identify potential code performance issues based on understanding of simulated processes, mathematical methods, computer limitations, and software environment;
step 3	develop testing strategy and test problems which addresses relevant code performance issues as they are viewed by stakeholders (*e.g.*, researchers, code developers, code users, fund managers, regulatory decision makers, project decision makers);
step 4	execute test problems and analyze results using standard graphic and statistical techniques;
step 5	collect code performance issues and code test problems in overview tables and matrix displays reflecting correctness, accuracy, efficiency, and field applicability;
step 6	identify performance strengths and weaknesses of code and testing procedure;
step 7	document each test setup and results in report form and as electronic files (text, data, results, graphics); and
step 8	communicate results (*e.g.*, executive summary, overview report. etc.).

Figure 3.1. Code testing and evaluation protocol

Table 3-1. Functions and features of a typical three-dimensional saturated porous medium finite-difference flow and transport model.

General Model Capabilities
- uncoupled Darcian ground-water flow and non-conservative single-component solute transport in saturated porous medium
- distributed parameter discretization

Spatial Orientation
- 1-D horizontal
- 1-D vertical
- 2-D horizontal
- 2-D vertical
- quasi 3-D (layered)
- fully 3-D

Grid Design
- 1-D, 2-D, or 3-D block-centered finite difference grid with constant or variable cell size

Time Discretization
- steady state flow
- transient flow
- transient transport
- variable time step size
- multiple transport time steps per flow time step
- multiple flow time steps per stress period
- variable stress periods

Matrix Solvers
- SOR
- ADI
- PCG

Aquifer Conditions
- confined
- leaky-confined
- unconfined

Aquifer Systems
- single aquifer
- single aquifer/aquitard
- multiple aquifers/aquitards

Variable Aquifer Conditions in Space
- variable layer thickness
- confined and unconfined conditions in same aquifer
- aquitard pinch out
- aquifer pinch out

Changing Aquifer Conditions in Time
- desaturation of cells at water table
- resaturation of cells at water table
- confined/unconfined conversion

Parameter Representations
- hydraulic conductivity: heterogeneous (variable in space), anisotropic
- storage coefficient: heterogeneous
- longitudinal dispersivity: heterogeneous
- transverse dispersivity: heterogeneous
- sorption coefficient: homogeneous (single value for total model area)
- decay coefficient: homogeneous

Fluid Conditions
- density constant in time and space
- viscosity constant in time and space

continued.....

Table 3-1 - continued.

Boundary Conditions for Flow
- fixed head
- prescribed time-varying head
- zero flow
- fixed boundary flux
- prescribed time-varying boundary flux
- areal recharge - variable in space and time
- induced recharge from or discharge to stream; stream may not be directly connected to ground water
- drains
- evapotranspiration dependent on distance surface to water table
- free surface, seepage surface

Solute Transport Processes
- advection
- hydrodynamic dispersion
- molecular diffusion
- linear equilibrium sorption
- first-order radioactive decay
- first-order chemical/microbial decay

Boundary Conditions for Solute Transport
- fixed concentration
- prescribed time-varying concentration
- zero solute flux
- specified constant or time-varying solute flux
- areal recharge of given (constant or time-varying) concentration
- induced infiltration of given (constant or time-varying) concentration
- concentration dependent solute flux

Sources/sinks
- injection/production well with constant or time-varying flow rate
- injection well with constant or time-varying concentration
- injection well with constant or time-varying solute flux
- production well with aquifer concentration-dependent solute outflux
- springs with head-dependent flow rate and aquifer concentration-dependent solute flux

Performance evaluation is aimed at characterizing the operational characteristics of the code in terms of: 1) computational accuracy; 2) limitations with respect to numerical convergence and stability; 3) sensitivity for grid orientation and resolution, and for time discretization; 4) sensitivity for model parameters; 5) efficiency of coded algorithms (including bandwidth, rate of convergence, memory usage, disk I/O intensity, etc.); and 6) resources required for model setup and simulation analysis. Tests are analyzed using various quantitative, often statistical evaluation techniques, as well as qualitatively using ranking and graphical techniques. Results of the performance evaluation are reported in checklists and in tabular form (see for example Tables 3-3a and 3-3b). Reporting on performance evaluation should provide potential users information on the performance as a function of problem complexity and setup, selection of simulation control parameters, and spatial and temporal discretization.

Table 3-2. Generic model functionality matrix; checked cells indicate that objective of test problem corresponds with a code function.

	functions				
test problem objective	function 1	function 2	function 3	function 4	function 5
test 1		X			X
test 2	X				
test 3		X			
test 4				X	X
test 5		X		X	
test 6				X	

Applicability assessment focuses on determining for which types of management problems the code is particularly suitable. In addressing this component of the protocol when the test strategy is formulated, attention is given to representative hydrogeology, engineering designs, and management strategies. Results of this assessment are primarily expressed qualitatively. An *applicability matrix* is used to document the extent of the applicability assessment, comparable to the functionality matrix. Reporting on applicability assessment includes information on how the test problems were implemented in terms of model setup and parameter allocation, providing users insight in the optimal use of the code for the particular type of applications.

The code testing protocol is implemented using a three-level *code testing strategy*, incorporating six types of test problems: 1) conceptual or intuitive tests; 2) analytical solutions and hand calculations; 3) hypothetical test problems with code intercomparison and intracomparison; 4) laboratory experiments; 5) field experiments; and 6) field applications. Reporting of test activities and results takes three forms: 1) documentation of individual tests; 2) analysis of completeness of test strategy and implications of test results; and 3) communication of test results to stakeholders.

Table 3-3a. Example performance evaluation table -- part 1

test case	number of nodes	number of time steps	time step (days)	convergence (number of iterations)	CPU use (sec)	RAM allocation (Kbytes)
1	500	1	10	5	11	550
2	500	1	10	50 (maximum)	205	550
3	500	1	10	11	34	550
4	500	1	10	22	55	550
5a	500	1	10	7	21	550
5b	5000	1	10	9	309	3880
5c	500	10	1	21	80	550

Table 3-3b. Example performance evaluation table -- part 2

test case	sensitivity to grid size[a)]	sensitivity to grid orientation[b)]	sensitivity to time discretization[c)]	stability[d)]	reproducibility[e)]
1	.1	.01	.1	satisfactory	0
2	.02	.007	.2	unsatisfactory	15
3	.03	.02	.1	satisfactory	0
4	.001	.008	.3	satisfactory	0
5a	.3	.04	.3	satisfactory	0
5b	.25	.05	.25	satisfactory	0
5c	.21	.045	.1	satisfactory	0

a) Sensitivity to grid size is determined by comparing the sum of absolute values of the differences in computed nodal values with the sum of computed nodal values divided by 2, employing two grid designs differing a factor 10 in number of active nodes.

b) Sensitivity to grid orientation is determined by comparing the sum of absolute values of the differences in computed nodal values with the sum of computed values divided by 2, using two identical grid designs rotated 45° with respect to each other.

c) Sensitivity to time discretization is determined by comparing the sum of absolute values of the differences in computed nodal values with the sum of computed values divided by 2, using for a constant period two time discretizations differing a factor 10.

d) Stability is rated "unsatisfactory" if in one or more runs stability problems are encountered; otherwise stability is rated "satisfactory."

e) Reproducibility is given in terms of a standard deviation for 10 runs using the same input data set.

3.2. PROTOCOL AUDIENCE

One of the problems with earlier code testing approaches was that they implicitly adopted a single often knowledgeable code testing audience. This target audience was assumed to understand typical functionality or performance problems related to specific types of codes. Communication of the test results to other than the target audience often led to misinterpretation of the test results, for instance, due to different interpretation of qualitative result descriptors. For the development of the protocol presented in this report, six audiences (or stakeholders) are identified: 1) researchers and peer reviewers focussed on the theoretical framework underlying the simulation code; 2) code developers and their programmers focused on the coding of algorithms, data structures and user interfaces; 3) code users focussed on addressing real world problems; 4) independent code testers providing expert advice on the functionality, correctness and efficiency of the code; 5) program managers, clients and other fund managers; and 6) regulatory decision makers. The needs and concerns of these stakeholders often vary substantially.

The primary interest of *researchers* is to improve the understanding of the physical world qualitatively by formulating governing principles and concepts and quantitatively by describing mathematical relationships. Simulation codes are often a tool in the scientific process to better understand complex natural phenomena. In general, code development is not the primary goal of such research. For this audience, model testing equates with establishing in the eyes of their peers that process descriptions, formulation of boundary conditions, and mathematical equations and solution methods are correct.

Code developers are primarily interested in determining and demonstrating that their code operates according to its intended objectives and yields accurate results. Code testing involves such issues as correct and efficient implementation of algorithms and code structures, numerical precision, and correct input and output handling, and efficient code operation.

Code users are, in general, interested in obtaining information concerning the code's functionality and applicability to the problem at hand, ease-of-use, efficiency in the use of resources, and sensitivity for parameter uncertainty. The relative benefits of one simulation code versus another are often determined by objective evaluation of what the code can do and how fast it can do it.

Project Managers/Fund Providers are primarily interested in knowing if a code is applicable and suitable for the specific project, the use of a particular code is the most cost-effective approach to solve the problem, and the results obtained with the code acceptable for involved regulatory agencies.

Finally, *regulatory decision makers* are focused on the credibility and accuracy of the simulation results. This credibility is based on the use of an adequate and reliable code (*i.e.*, the tool), a well-conceptualized site, good data, and well-executed problem analysis.

The proposed functionality analysis, performance evaluation and applicability assessment protocol aims to provide key information elements for regulators, fund providers and managers and code users, and instruction for systematic testing and documentation for model researchers and code developers.

3.3. SOME PROTOCOL DESIGN ISSUES

The protocol is meant to be used in testing a wide range of subsurface fluid flow and transport codes. These simulation codes employ a variety of mathematical process descriptions and solution techniques and various operational features to accommodate the complexity of real world problems and the management strategies and engineering approaches in addressing these problems. Therefore, the protocol is designed to handle:

- a large variety of process descriptions, boundary conditions and system stresses;
- a wide range of code applications (*e.g.*, situations, parameter ranges);
- different spatial (*i.e.*, grid discretization / nodal distributions) and temporal discretization schemes;
- different mathematical solution techniques; and
- different computer languages, hardware platforms and software environments.

The code testing evaluation criteria consist of a series of statistical and graphical measures which describe the test results in either absolute or relative terms. The measures included in the protocol were selected based on the following considerations:

- well-defined, meaningful, and objective;
- easy to use, either manual or by using standard computer software;
- quantitative descriptor of accuracy and performance; and
- qualitative illustrator differences between code results and the benchmark solution.

3.4. THE TEST METHOD

3.4.1. Functionality Analysis

Ground-water simulation codes typically include a variety of simulation functions and operational features. Furthermore, such codes are characterized by their mathematical framework and computer implementation issues. Thus, before systematic testing can take place, these code characteristics need to be identified, defining the code's *functionality*. Functionality description defines, in qualitative terms, the available functions of the simulation code. It should be noted that using the functionality description element of the protocol in a consistent, comprehensive manner while developing a code's documentation will provide necessary, easy-accessible information for code selection.

Based on the resulting *functionality description*, a *functionality test strategy* is developed consisting of : 1) designing or selecting test problems, targeted at all of the identified characteristics; 2) test running the code for meaningful and challenging parameter selections; 3) standardized qualitative and quantitative analysis of the test results; and 4) documenting the results in a comprehensive and informative manner. The execution of such a test strategy is called *functionality testing*. Functionality testing is a part of the protocol's test strategy discussed later in this chapter; in the test strategy functionality testing is combined with performance testing and applicability assessment aspects. In the protocol, the combined procedures of functionality description and functionality testing is defined as *functionality analysis*.

The objectives of functionality analysis are:

- to identify and describe functions and features of a simulation code;

- to test and evaluate the code for conceptual and numerical correctness, and efficient and error-free operation; and
- to document code description and test results in a consistent, intercomparable, comprehensive and informative manner.

The functionality analysis procedure is illustrated in Figure 3-2 and Table 3-4. They show the order in which the various steps are taken.

Table 3-4. Functionality Analysis as a four-step procedure.

Step 1:	Analysis and description of the code's functionality;
Step 2:	Determination of test issues and design of functionality aspects of test strategy;
Step 3:	Execution of test problems, producing standardized test evaluation data; and
Step 4:	Evaluation of produced test information using established graphical and statistical measures, and production of functionality matrix.

3.4.1.1. *Functionality Description*

Functionality description is the qualitative analysis of the capabilities of a simulation code. The available functions are grouped and systematically described using a set of standard descriptors. These descriptors have been developed as part of an earlier ground-water model information management project (van der Heijde, 1994) and are presented in tabular form in Appendix A. If necessary, the list of descriptors may be adapted or expanded to cover features resulting from new research or software development progress. Based on these tables of descriptors, a checklist has been developed to present a quick overview of functions and features of the code (see Appendix B).

The standard format is designed to be applicable to any ground-water simulation model code. It includes a brief overview description of the simulation code (*i.e.*,code authors, contact address,

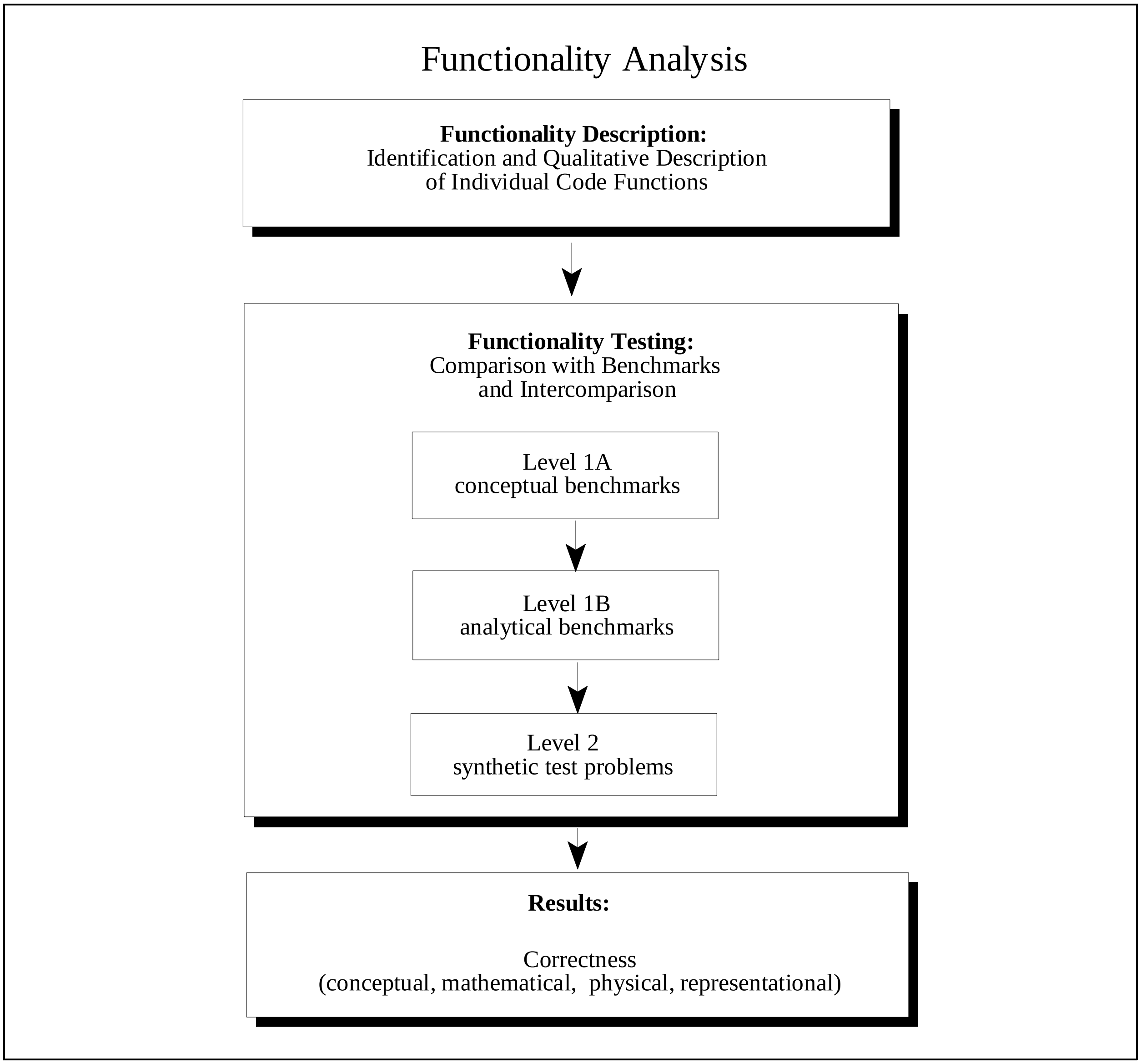


Figure 3-2. Overview of functionality analysis procedure.

required computer platforms, etc.). This is followed by a section that is divided into functionality categories corresponding to sets of specific code functions. This approach facilitates the selection, by potential users, of the most suitable code for a given application, based on review of the standard code description. It also defines, for code developers, testers and reviewers, the simulation code functions that must be documented and tested.

3.4.1.2. *Identification of Potential Performance Issues*

The code functionality description forms the basis for the identification of issues and concerns related to code correctness and performance. The issues can be grouped in five broad categories: 1) conceptual problems in theoretical framework; 2) mathematical (non-coding) issues related to formulation of equations, solution techniques, etc.; 3) implementation of algorithms in code logic and code structures; 4) I/O handling (*e.g.*, file interaction, keyboard/screen interaction); and 5) internal data handling (*e.g.,* argument handling in subroutines, common blocks, equivalencies, etc.. Issues listed in categories 1 and 2 are dependent on the type of simulation code being tested. For example, numerical dispersion and oscillations in the simulation of sharp concentration fronts may occur in solute transport models. Non-convergence or exorbitant computation times may occur in unsaturated zone flow models or multi-phase flow models due to strong non-linear behavior or poorly-chosen initial conditions.

Based on the analysis of potential correctness and performance problems a test strategy is formulated which matches test issues with test problems in a comprehensive manner. Test problems are chosen to address specific functions of the code or to emphasize specific performance issues. Typically, test problems are based on the availability of adequate benchmarks, representative hypothetical situations, or independently observed physical systems (see section on test strategy later in this chapter). A detailed discussion of the elements of the test strategy is provided later in this chapter.

3.4.1.3. *Functionality Tables and Matrices*

Full evaluation of a ground-water simulation code requires taking a systematic approach to the design and reporting of test issues and performed tests. An adequate testing strategy addresses all functions and features of the code and related performance issues by formulating test objectives for each test and describing how the test will meet these objectives. Performance issues addressed in the functionality testing, test objectives for each test, and the benchmark solution (if available) should be included in functionality tables. Appendix C lists example functionality tables for three-dimensional saturated flow and solute transport codes.

To simplify the functionality analysis procedure, the two components of functionality analysis, functionality description and functionality testing, are combined in a *functionality matrix* (see Figure 3-4). The left column of the functionality matrix represents the functionality description by listing the code functions which are to be tested. The top row of the functionality matrix represents the functionality testing by listing benchmark solutions which are used to address the code functions. These two elements define a two-dimensional matrix that is used to provide a quick overview of tested functions. The matrix can also be used to determine the availability of benchmark solutions.

Each cell within the center of the matrix actually represents a series of specific questions and/or issues which must be evaluated before a simulation code is fully functionality tested. These questions and issues are summarized in a series of functionality tables presented in Appendix C. The functionality tables can be considered as a third dimension extension of the functionality matrix. This concept is illustrated in Figure 3-4; an example application of the functionality matrix is presented in section 4 of this report. The functionality matrix is shown as the basis of the chart with the corresponding background information overlain on it. The resulting three-dimensional figure integrates the functionality issues, test objectives, and benchmark solutions into a single illustrative figure. For practical reasons, the use of the functionality matrix is limited to the two-dimensional primary level shown in Figure 3-4. Each cell of the matrix is marked off when the function has been evaluated in accordance with the protocol and the associated issues have been addressed. The completed functionality matrix provides a kind of summary report structure showing in a glance deficiencies in the testing of a particular code.

3.4.2. Performance Evaluation

Performance evaluation is designed to characterize code behavior in terms of numerical accuracy, efficiency, sensitivity and reliability. This is accomplished by measuring the results of comparative testing and analyzing operational code characteristics during the execution of test problems.

Specifically, code responses are monitored for a realistic range of parameters and model configurations. The main modeling variables influencing code performance are: 1) spatial discretization and grid orientation; 2) time-stepping scheme; and 3) solution technique and related numerical parameters. Figure 3-4 shows the relationship between the main components of Fig. 3-3.

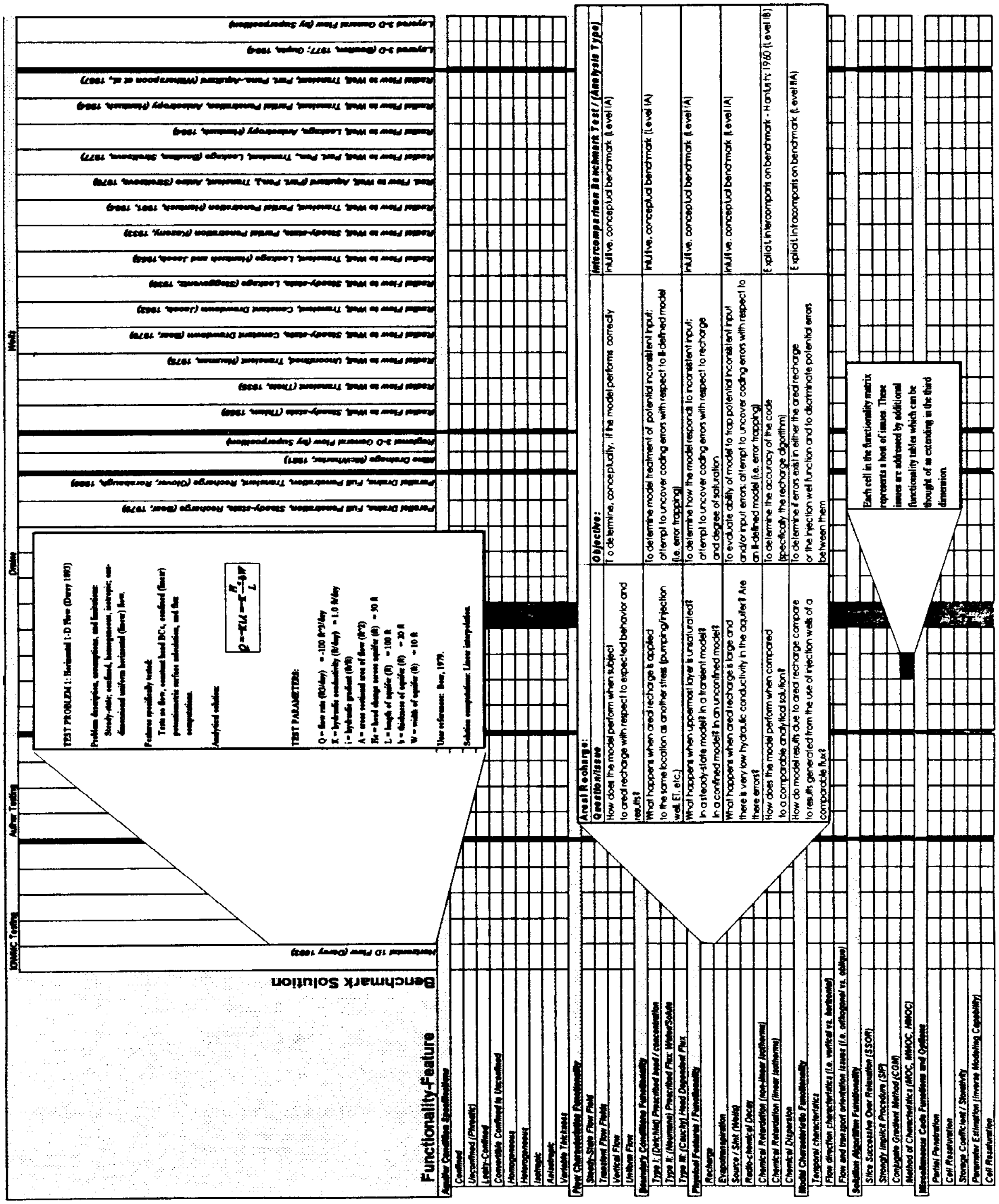


Figure 3-3: Generic Functionality Matrix

performance evaluation. The results of performance evaluation are expressed in terms useful for code selection, modeling resource allocation, and overall modeling project management. Table 3-5. shows the major steps in Performance Evaluation.

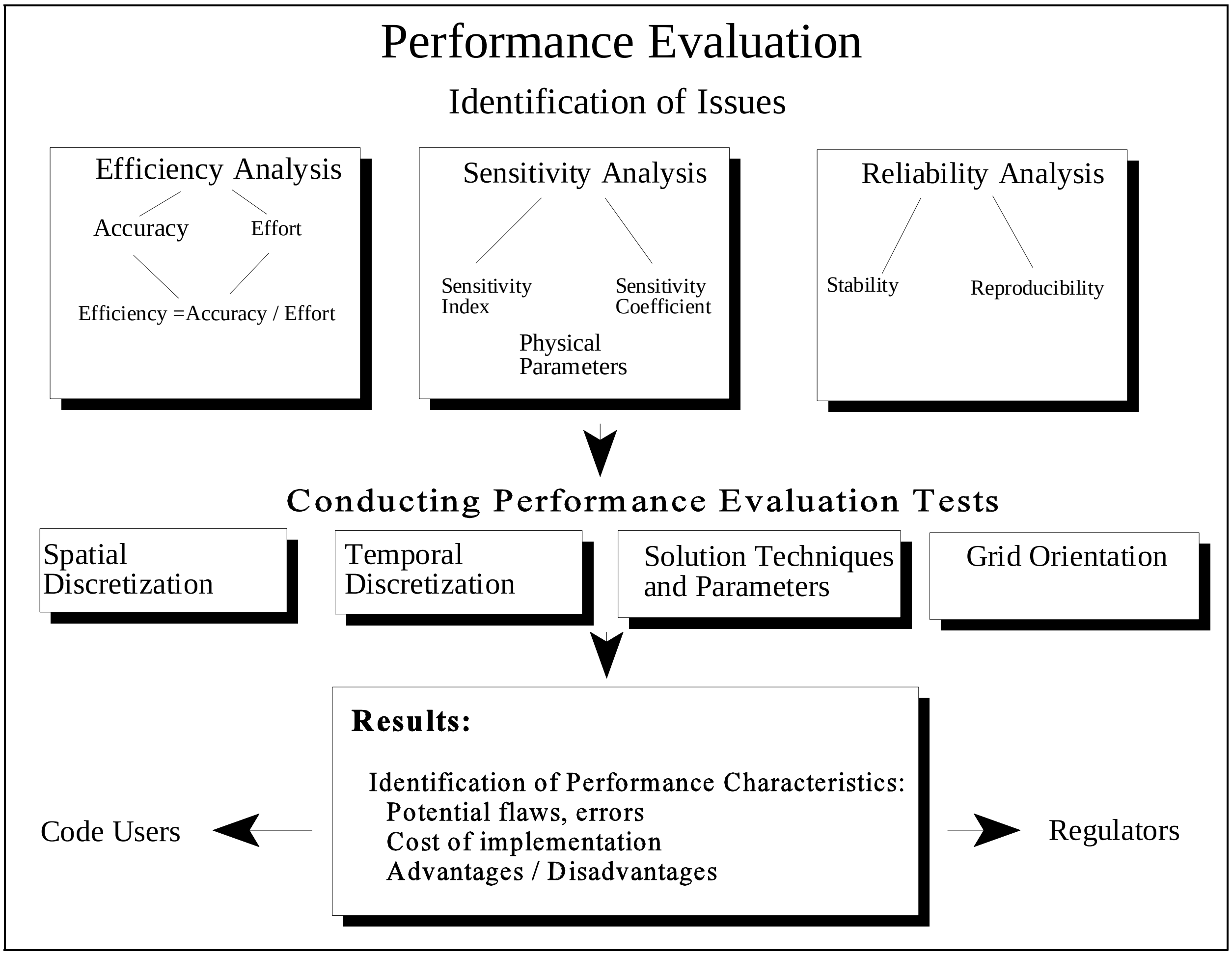


Figure 3-4. Overview of the performance evaluation procedure.

3.4.2.1. *Performance Evaluation Elements*

The main elements of performance evaluation are code accuracy, code efficiency in terms of code use resources required to achieve a specific accuracy, code sensitivity to input variations, and code reliability in terms of solution stability and reproducibility of results (van der Heijde and Elnawawy, 1992). Each of these elements is discussed in detail in the following section.

Table 3-5. Performance Evaluation as a four-step procedure.

Step 1:	Definition of the performance issues for the specific code;
Step 2:	Selection of appropriate test problems;
Step 3:	Producing performance evaluation measures while running the test problems; and
Step 4:	Evaluation of results using established graphical and statistical measures, and preparation of performance evaluation report.

3.4.2.2. *Code Accuracy*

One of the main objectives of the performance evaluation procedure is the determination of the *accuracy*, which may be obtained with a simulation code. Code accuracy is a quantitative measure for the correctness of the calculations made with the computer code. It is measured by comparing the result of a code based computation with an independently derived value for the calculated entity, assuming that this second value is the correct result of the calculation (*i.e.*, the benchmark). Code accuracy may quantitatively be expressed using statistical type measures. In the testing of analytical models, such quantitative evaluation of code accuracy is rather straightforward. However, in the testing of numerical modeling codes, evaluation of code accuracy is often more complicated, among others because an independent benchmark may not be available, and because the code based computations are inherently subject to schematization and discretization errors. Code accuracy can be measured for different discretization densities, time-stepping schemes, grid orientations and numerical parameters using the benchmarks and intercomparison tests developed as part of the functionality analysis. Alternative approaches to assess code accuracy under these latter conditions are discussed in the section on code testing strategy. Results are summarized in tables to provide the code users with relevant information and utilization the subsequent efficiency analysis.

3.4.2.3. *Efficiency Evaluation*

The *efficiency* of a simulation code is defined as the level of effort and computer resources required to obtain a user-specified code accuracy as given in equation 3.1.

$$\textit{Code Efficiency} = \textit{Code Accuracy} \,/\, \textit{Level of Effort} \qquad (3.1.)$$

The *level-of-effort* required to set up a model problem using a particular simulation code is an important and often unreported aspect of performance evaluation. This level-of-effort is primarily determined by the manpower, and thus cost, required for the simulation study. The major difficulty in determining the level-of-effort is where the distinction is made between field characterization and model preparation. In the terms of this protocol, site characterization and model conceptualization are basically independent of the selected code and therefore not included in the determination of the level-of-effort.

One of the main labor-intensive components of model preparation is the creation and editing of input files, reflecting the spatial and temporal variability of the modeled system. It is here that specifically spatial and temporal discretization play an important role.

The amount of effort required to use any simulation code includes two major components: human resources and computer resources. Each of these is made up of sub-components. The human resources component includes all human effort required to translate a conceptual model into a finished, interpreted simulation model; this includes the time and effort involved in data preparation and input, as well as the time and effort required for data reduction and analysis. It is assumed that the effort needed to understand code documentation, assess the code's capabilities, and install the code on user's platform is the same for all simulation codes. The protocol addresses only the effort involved in the actual set-up and execution of the test cases using standard measures and assuming an expert modeling team.

To quantify the level-of-effort, a new parameter has been developed, called the "Human Effort Parameter" (HEP). HEP consists of four major components:

HEP_1 = effort involved in model grid design (both manually or automatically);
HEP_2 = effort involved in spatial parameter allocation (both manually or automatically);
HEP_3 = effort involved in setting up time-varying parameters and stresses; and
HEP_4 = effort involved in the manipulation and analysis of results.

Each of these parameters can be defined by a semi-quantitative expression. The sum of these parameters is equal to the total required human effort.

The effort involved in model grid design, HEP_1, is directly related to the total number of grid cells or elements in the model. A code-specific factor, C_1, is used to characterize the ability of the code (or related peripheral software) to automate grid design (see equation 3.2).

$$HEP_1 = i*j*k*C_1 \qquad 0 \leq C_1 \leq 1 \tag{3.2}$$

where i is number of nodes in x-direction, j is number of nodes in y-direction, and k is number of nodes in z-direction. If the code contains an automatic grid generation algorithms, together with bandwidth optimizers, C_1 is small and therefore HEP_1 is small.

Similarly, the effort involved in parameter allocation, HEP_2, is related to the total number of spatially varying parameters in a code (see equation 3.3). Again, a code-specific factor, C_2, is used to characterize the level of automation of a code in parameter allocation or its ability to use parameter zoning. If the code contains a parameter allocation algorithm or preprocessor, that automates or reduces the effort required in spatial parameterization, HEP_2 can be significantly reduced. The greater the automation ability, the closer C_2 and, therefore, HEP_2 approach zero.

$$HEP_2 = p*C_2 \qquad 0 \leq C_2 \leq 1 \tag{3.3}$$

where p is number of spatially varying parameters.

The effort involved in allocating time-varying parameters and stresses, HEP_3, is related to the total number of temporally varying parameters in the model (see equation 3.4). Note, that in steady-state simulations, none of the parameters vary in time, thus HEP_3 is equal to zero. As before, HEP_3 can be modified by a code-specific factor, C_3, which is used to characterize the temporal automation ability of the simulation code. If the code is embedded in an interface which automatically uses time series information stored in a data base, or has a preprocessor to perform this function, HEP_3 can be reduced. The greater the automation ability, the smaller C_3 and HEP_3.

$$\begin{aligned} HEP_3 &= t * C_3 \\ 0 \leq\ & C_3 \leq 1 \end{aligned} \tag{3.4}$$

where t is number of time-varying parameters (t=0 for steady state simulations).

The level-of-effort needed to manipulate code output, and/or perform output analysis, HEP_4, is more difficult to quantify. Among others, it depends on the level of integration between simulation and postprocessing within a dedicated software environment and the structure and form of the files prepared by the simulation code. Furthermore, this level-of-effort depends on the graphic, statistical, word-processing, and spreadsheet software available to the analyst and his/her experience with this software. HEP_4 reflects these issues and is proportional to the total number of model nodes and total number of time steps of interest (see equation 3.5).

$$\begin{aligned} HEP_4 &= i * j * k * T * C_4 \\ 0 \leq\ & C_4 \leq 1 \end{aligned} \tag{3.5}$$

where i is number of nodes in x-direction, j is number of nodes in y-direction, k is number of nodes in z-direction, T is number of time steps of interest, and C_4 is a factor comparable to the factors C_1 - C_3 in the previous equations. It should be noted that C_1 -C_4 are empirical factors, chosen based on experience in pre- and postprocessing with the particular simulation code.

The total amount of effort required (HEP_{total}) to create and analyze a simulation model is defined as the sum of all the components previously described. This follows in Equation 3.6.

$$HEP_{total} = HEP_1 + HEP_2 + HEP_3 + HEP_4 \tag{3.6}$$

Another element that defines efficiency is *computer resources utilization.* Computer resources utilization is determined by the required platform (and its intrinsic cost), problem simulation time (cpu use and I/O time), random access memory (RAM) required to successfully run a data set, and mass storage requirements for data sets and result files. Objective comparison of computer resource needs of various simulation codes requires the use of a standard computer configuration. Computer hardware magazines regularly publish performance comparisons for various hardware platforms, which can be used to determine a code's computer resource utilization requirements on a user-specified platform. One way to present this type of information is given in Table 3-3a.

Using these calculated parameters for accuracy and effort, the code tester can define *code efficiency* in several ways. A derived measure of efficiency is computed by dividing the measured accuracy parameter of interest (using the statistical measures described later in this chapter) by the measured effort parameter of interest or their cost equivalent (see Table 3-6). Each separate efficiency measure provides a different type of code performance information. Using the defined efficiency measures, the performance of a code can now be evaluated by comparing its efficiency for various spatial and temporal discretizations, grid orientations, and solution algorithm parameters. For example, efficiency analysis, performed using the proposed procedure, can provide information on the cost-benefit ratio for different discretization or parameterization schemes and determine optimum grid densities, or time-stepping schemes.

3.4.2.4. *Sensitivity Analysis*

Sensitivity analysis is a significant component of code performance evaluation. Intera (1983) stated that it is important to quantitatively or semi-quantitatively define the dependence of a selected code performance assessment measure on a specific parameter or set of parameters. Sensitivity analysis is used to identify the most influential parameters, or code issues, that may affect the accuracy and precision of code results. This information is important for the code user because it allows the establishment of required code accuracy and precision standards as a function of data quantity and quality (Hern, *et al.* 1985). Sensitivity analysis can be used by code developers to improve code simplicity and, therefore, efficiency, and results may increase the understanding of the code by the user.

Table 3-6. Generic matrix of sample efficiency measures

Accuracy Measures[1]	**Effort measures**			
	RAM	**CPU time (CPUT)**	**Number of iterations (NITER)**	**Human Effort Parameter (HEP)**
RMS	RMS/RAM	RMS/CPUT	RMS/NITER	RMS/HEP
MAE	MAE/RAM	MAE/CPUT	MAE/NITER	MAE/HEP
ME	ME/RAM	ME/CPUT	ME/NITER	ME/HEP

Identification of the change in simulation model results caused by a known change in a specific input parameter provides the user with an understanding of the importance of that parameter. If a modest change in an input parameter causes a large change in output results, the code is considered to be sensitive to that parameter.

There are various ways to assess the sensitivity of model results for changes in model parameters, including the calculation of sensitivity coefficients or sensitivities (*e.g.*, Cooley *et al.*, 1986), the use of joint sensitivity equations (*e.g.*, Sykes *et al.*, 1985), and the application of stochastic modeling, for example using monte carlo analysis (Clifton and Neuman, 1982; Smith and Freeze, 1979; and Thompson *et al.*, 1989). Typical measures of this phenomenon is the sensitivity index, S_t, defined by Fjeld *et al.* (1987) and the relative sensitivity S_r used by Nofziger *et al.*, (1994).

To determine the sensitivity index, nominal, minimum, and maximum values for the selected input parameter are specified by the code evaluator. The values of the dependent variable are determined for these three values of the input parameter. The resulting values of the dependent variable are:

h_t^{nom} for the nominal input parameter value;

h_t^{pmin} for the minimum input parameter value; and
h_t^{pmax} for the maximum input parameter value.

This approach yields the upper and lower bounds for the values of the dependent variable based upon the upper and lower values of the input parameters. This information is used in the calculation of the sensitivity index S_t, which is defined as

$$S_t = \frac{h_t - h_t^{nom}}{h_{nom}^{\max}} \tag{3.7}$$

where

h_t = value of the dependent variable for either the minimum or maximum value of a given input parameter
h_t^{nom} = value of dependent variable determined for some nominal value
h_{nom}^{max} = maximum instantaneous value of the dependent variable

The maximum instantaneous value of the dependent variable, $h_{nom}^{\max}$ (*i.e.*, the nominal value of the dependent variable at the maximum time), is based upon nominal values of the parameter. The sensitivity index is most useful for evaluating the impact of individual input parameters on local variables, or for evaluating parameters that describe the overall simulation model configuration. These parameters can include: 1) Peclet and Courant numbers for spatial and temporal discretization; 2) solution parameters; and 3) global input parameters, such as dispersivity and degree of anisotropy, or spatially defined parameters in a homogeneous system. The sensitivity index cannot be used effectively for sensitivity analysis of input parameters that vary in space. To apply the sensitivity index approach within the performance evaluation procedure, the code is run against benchmarks selected for the functionality analysis procedure of the code testing and evaluation protocol.

The *relative sensitivity* is defined as $S_r = S*x/f$ where S is the sensitivity coefficient, f is the value of the model output, and x is the value of the model input parameter (Nofziger *et al.*, 1994). The sensitivity coefficient can be obtained from $S=\Delta f/\Delta x$ where Δf is the change in output f due to a change Δx in the input parameter. The relative sensitivity can be used to estimate the relative change in model output, $\Delta f/f$, from the relative change in input parameter, $\Delta x/x$, using the equation

$$\frac{\Delta f}{f} = S_r \frac{\Delta x}{x} \tag{3.8}$$

Nofziger *et al.* (1994) used this measure in evaluating the sensitivity in travel time, concentration, mass loading and pulse width of a contaminant at the water table for four unsaturated zone fate and transport models (RITZ, VIP, CLMS, and HYDRUS). Sensitivity was investigated for a wide variety of conditions including organic carbon content, bulk density, water content, hydraulic conductivity, organic carbon partition coefficient, degradation half-life, rooting depth, recharge rate, and evapotranspiration. The study included investigation of uncertainty in predictive capability of the models and found that large uncertainty exists due to the combination of sensitivity and high parameter variability in natural soils.

Zheng (1993) used the sensitivity coefficient, S_c , a measure of the effect that the change in one factor or parameter has on another factor or result. Practically, this represents the change in either some calibration criteria or relative accuracy measures, expressed as residual difference, R, (*e.g.*, RMSE, or comparable statistical measure) divided by the change in the input parameter, P (see equation 3.9).

$$S_c = \Delta R / \Delta P \tag{3.9}$$

where ΔR is change in accuracy measure of choice and ΔP is change in input parameter.

As part of the code performance evaluation, code sensitivity must be established not only for code-specific parameters, but also for model configuration and setup. For instance, changes in grid density can be expressed as a factor, ΔP. If grid density is doubled (*i.e.*, grid distances are halved), the value of ΔP is doubled. It should be noted that, for a finite difference type of grid, doubling the grid density will result in doubling, squaring or cubing the number of nodes for one-dimensional, two-dimensional, and three-dimensional models, respectively. If the change in code results due to this doubling of grid density is measurable using a statistical measure like RMSE, the sensitivity coefficient for spatial discretization can be calculated.

3.4.3.5. *Reliability Evaluation*

A reliable simulation code is one which: 1) is free of run-time errors and failures; 2) converges for a wide range of parameters; and 3) yields results which are fully reproducible from one execution to the next. Run-time errors are addressed in the functionality analysis and testing. Other failures might relate to convergence and stability problems. Stability problems have been discussed in Chapter 2. It is important to realize that sometimes stability problems are directly related to code use, such as selecting improper solution techniques, or using incorrect or unsuitable model configurations. For example, flow simulations are sensitive to correct setup of initial and boundary conditions. In a relatively unstressed system, initial conditions close to reality and a sufficient number of first-type boundary conditions are required to constrain the model enough to reach convergence. Stability (and convergence) is evaluated during functionality and applicability testing by keeping track of non-successful simulation runs and the conditions under which they occur (see Table 3-3b). These conditions include problem setup, parameter allocation, and solver selection.

Reproducibility refers to the code characteristic illustrating that results from a specific simulation model are identical between different runs on a specific computer platform. Often, this characteristic is extended to across-platforms comparisons. In the latter case, differences in computational precision among platforms might cause differences in round-off errors. The code should be designed such that this type of errors do not have a great influence on the simulation results. It should be noted that some solution techniques inherently prevent reproducibility. This is specifically the case with the random walk method for solute transport modeling.

3.4.2.6. *Performance Evaluation Factors*

There are four major factors which influence performance evaluation in terms of accuracy, effort, efficiency, sensitivity, and reliability. These factors are: 1) spatial discretization; 2) temporal discretization; 3) solution techniques and parameters; and 4) grid orientation. These factors should be investigated in conjunction with the functionality test problems. Selected functionality test cases should be altered to allow the sensitivity of the code for these factors.

The specification of spatial discretization has a very significant impact on code accuracy, effort required and, therefore, overall efficiency. Also, spatial discretization might influence convergence behavior in terms of stability and speed. If spatial discretization is too low, the accuracy of the code results can suffer; contrarily, if spatial discretization is too high, the overall effort required can be exorbitant and even prohibitive. Therefore, it is important to determine the optimum spatial discretization required to provide a stable solution with an acceptable efficiency level. This might require running selected test problems with increasingly dense grids and monitoring convergence and efficiency measures. For some codes, an acceptable level of spatial discretization can be derived from stability criteria (see Chapter 2), which is defined as the ratio of ground-water velocity times characteristic grid size over dispersion. For example, the degree of spatial discretization for codes that simulate advective-dispersive transport processes can be derived from the Peclet Number.

Temporal discretization impacts code convergence and efficiency in the same manner as discussed for spatial discretization. To evaluate this characteristic, test problems are set up using different time-stepping schemes. These differences might take the form of an increased number of time steps for the same simulation period, or the use of a non-linear time-stepping scheme to better reflect the behavior of the time-derivative of the dependent variable (e.g., time-stepping for the Theis equation test case by Prickett and Lonnquist, 1971). As is the case with spatial discretization, time-stepping for the simulation of solute transport can be expressed by a stability criterion, the Courant Number (see Chapter 2), which is defined as the ratio of ground-water velocity multiplied by the minimum time step divided by the characteristic distance between grid nodes.

Code performance is often highly dependent on the selection of the equation solver and the choice of solver parameters. If problems in stability occur during testing, or the code seems to be inefficient, selection of an alternative solver (if available) or adjustment of solver parameters might improve the situation. Solution parameters that might be investigated include: 1) error criterion or convergence tolerance for iterative solutions (expressed in terms of dependent variable and/or mass balance); 2) the maximum number of iterations allowed; 3) weighting factors; and 4) iteration and acceleration parameters. Although the required human resources, as expressed by the HEP, are not impacted by changes in solution techniques and/or solution parameters, required computer resources can be significantly affected by changes in solution techniques and parameters. The information on code performance based upon solution parameters and techniques is summarized in the performance evaluation checklist. This information might form the basis for guidelines on specification and implementation of solution techniques and parameters.

Many ground-water flow and transport models do not include cross-terms for hydraulic conductivity and dispersivity. Thus, the model grid is supposed to be oriented such that its principle axes are parallel to the ground-water flow and contaminant transport directions. In practice, nonuniform flow situations makes this requirement often difficult to meet. The degree of error, attributable to non-orthogonal flow and transport, needs to be characterized as part of the protocol. The effect of the absence of cross-terms can be explored by intercomparison with codes which include these cross-terms, and by intracomparison of results obtained with the tested code for different grid orientations, specifically using tests for which an analytical solution is available. Typically two grid orientations are used: parallel to flow (*i.e.*, orthogonal) and under 45 degrees with the flow direction (*i.e.*, oblique). It should be noted that to obtain comparable levels of accuracy oblique grids might require significantly longer computation times.

3.4.2.7. *Performance Evaluation Tables*

The results of the accuracy analysis for each of the performance evaluation categories are compiled into a summary table or checklist. For example, Tables 3-8, 3-9, and 3-10 show summary tables for accuracy, effort and sensitivity versus the four performance evaluation factors grid discretization and orientation, time-stepping, and solution technique setup.

3.4.3. Applicability Assessment

Model users, environmental regulators and model reviewers need to know if a particular code is appropriate for the specific site conditions and simulation scenarios of a project. This determination needs to be made during the code selection process, prior to the use of the selected code in the study. Commonly, the applicability of a code is determined from careful analysis of its functionality and evaluation of the needs of the project. Often, this process is enhanced by analysis of previous applications of the code, specifically for comparable site conditions and simulation

Table 3-7. Generic table of accuracy analysis results for a specific test problem.

Performance Evaluation Categories	Statistical Measures			
	RMS	**MAE**	**ME**	**Other**
SPATIAL DISCRETIZATION				
One half density				
Single density				
Double density				
Peclet number = 10				
Peclet number = 1				
Peclet number = 0.1				
TEMPORAL DISCRETIZATION				
One half density				
Single density				
Double density				
Courant number = 10				
Courant number = 1				
Courant number = 0.1				
SOLUTION TECHNIQUE AND PARAMETERS				
Tolerance = 0.0001				
Tolerance = 0.001				
Tolerance = 0.01				
SSOR, Acceleration parameter = 1.4				
SSOR, Acceleration parameter = 1.6				
SIP, Acceleration parameter = 1.0				
GRID ORIENTATION				
Parallel				
Oblique				

Table 3-8. Generic table of effort analysis results for a specific test problem.

Performance Evaluation Categories	Human Resources Use Measures				
	HEP_1	**HEP_2**	**HEP_3**	**HEP_4**	**HEP_{total}**
SPATIAL DISCRETIZATION					
One half density					
Single density					
Double density					
Peclet number = 10					
Peclet number = 1					
Peclet number = 0.1					
TEMPORAL DISCRETIZATION					
One half density					
Single density					
Double density					
Courant number = 10					
Courant number = 1					
Courant number = 0.1					
SOLUTION TECHNIQUE AND PARAMETERS					
Tolerance = 0.0001					
Tolerance = 0.001					
Tolerance = 0.01					
SSOR, Acceleration parameter = 1.4					
SSOR, Acceleration parameter = 1.6					
SIP, Acceleration parameter = 1.0					
GRID ORIENTATION					
Parallel					
Oblique					

Table 3-9. Generic table of sensitivity analysis results for a specific test problem.

Performance Evaluation Categories	**Sensitivity Measures**			
	Sensitivity C	**Sensitivity Coeff.**	**Sensitivity Index**	**Other**
SPATIAL DISCRETIZATION				
One half density				
Single density				
Double density				
Peclet number = 10				
Peclet number = 1				
Peclet number = 0.1				
TEMPORAL DISCRETIZATION				
One half density				
Single density				
Double density				
Courant number = 10				
Courant number = 1				
Courant number = 0.1				
SOLUTION TECHNIQUE AND PARAMETERS				
Tolerance = 0.0001				
Tolerance = 0.001				
Tolerance = 0.01				
SSOR, Acceleration parameter = 1.4				
SSOR, Acceleration parameter = 1.6				
SIP, Acceleration parameter = 1.0				
GRID ORIENTATION				
Parallel				
Oblique				

scenarios. Optimally, documentation of a simulation code should include discussion of the applicability of the code to various hydrogeological and contamination situations, and for the analysis of a variety of engineering and management issues. Such discussions should not only address the type of applications which can be performed, but also how to set up the model to optimally represent the application aspects of concern.

Applicability assessment is used to help determine the range of situations that can be simulated by the code, reflecting typical applications for which the code might be used. Typical questions raised during the applicability assessment include:

- is the code applicable to the problem/site-specific hydro(geo)logical system;
- can the code be used to analyze the engineering and management solutions of interest; and
- can the model application, developed using the code, yield results that are feasible and can be calibrated to real-world situations.

Usually, applicability assessment takes the form of comparative simulation of standard, real world problems or their simplified, synthetic representation.

The representative applications, expected to be analyzed with the code, are categorized using a three-level, hierarchical classification approach (see Figure 3-5). For example, in analyzing applicability issues of saturated flow and solute transport codes, four broad application categories of hydrogeological scenarios can be distinguished: 1) ground-water resource development (*i.e.*, water supply); 2) hydrogeological control (*e.g.*, construction site or mine dewatering); 3) pollution control (*e.g.*, remediation); and 4) ground-water protection (*e.g.*, recharge zone delineation). Each of these application categories may be further characterized, based upon the physical system being represented and the engineering and management scenarios supported. These primary components can be further divided into a number of individual elements, representing specific code options. These code options can be either represented by directly activating a particular code function, or by careful formulation of model conceptualization and model setup.

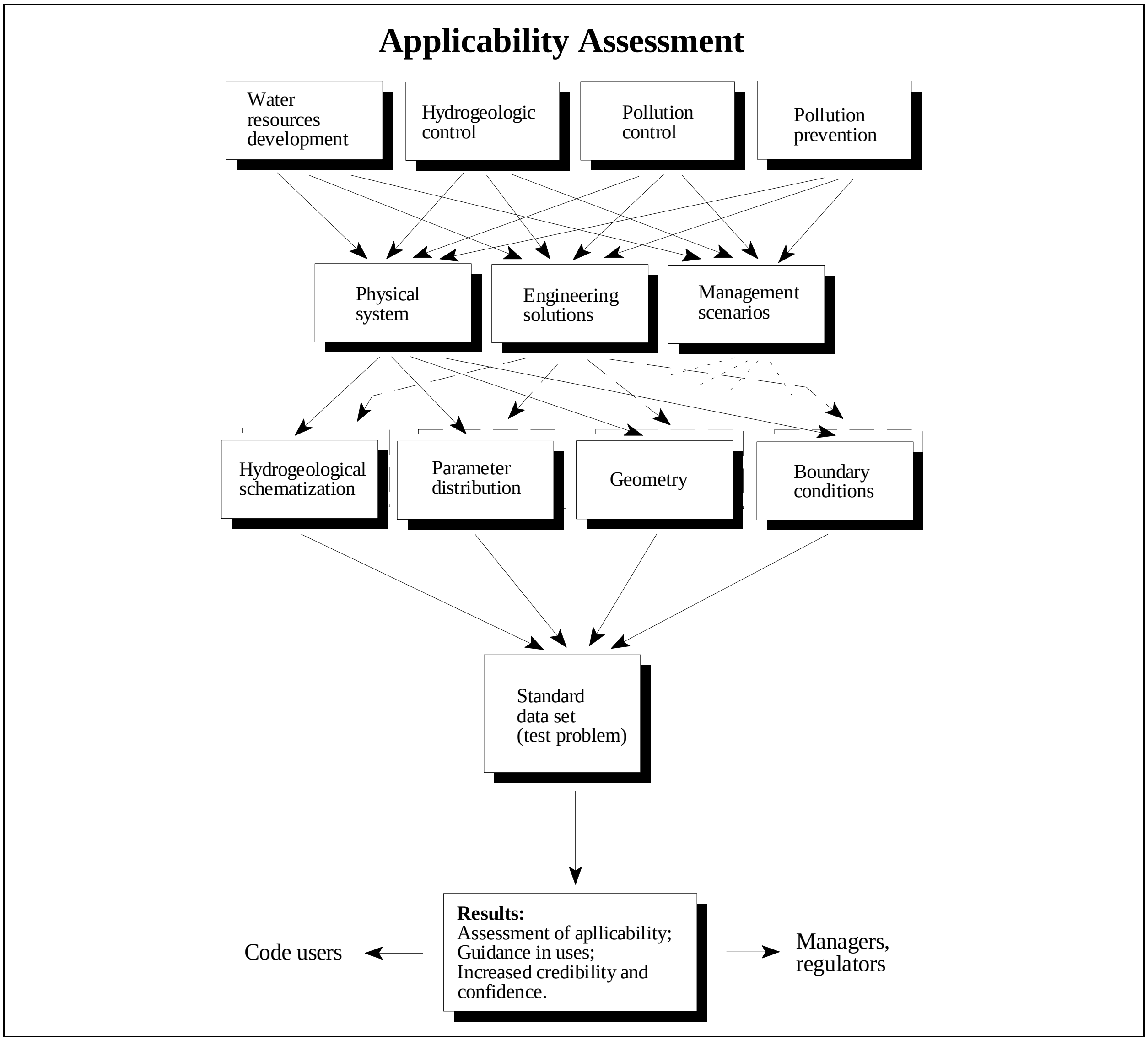


Figure 3-5. Overview of applicability assessment procedure.

Applicability assessment yields qualitative results which are illustrative for the code. Rather than objectively comparing code results to a benchmark solution, applicability assessment evaluates how well the simulation code represents representative, standard applications. To remove some of this subjectivity from applicability assessment, code intercomparison may be performed using the standard data sets. "Good" results are obtained when the code performs the applicability tests without causing run time errors, and when the results seem reasonable. In some cases, real-world application may be used. Then, more objective evaluation is possible, specifically when simulation-independent information regarding the behavior of the dependent variable(s) and the mass balance is available. An overview of the applicability assessment procedure is presented in table 3-10. Table 3-11 presents an example applicability assessment table. In this table, the applicability assessment issues are compared with the test design criteria. Actual applicability assessment tables will have more detail with respect to addressed issues than this example table.

Table 3-10. Applicability assessment as a four-step procedure.

Step 1:	Identification and description of applicability issues and related questions and problems;
Step 2:	Design and/or selection of representative sample applications;
Step 3:	Execution of test problems and evaluation of results as function of grid design, time-stepping, and general model formulation;
Step 4:	Summarizing results in applicability assessment tables.

The test data set design criteria are derived from the requirement that the data sets address the significant issues associated with typical code applications. For example, a ground-water pollution control application typically involves layered aquifer characteristic and complex physico-chemical soil interactions. The engineered remediation alternatives may require simulation of patch sources, vertical line barriers, distributed water supply wells, and horizontal shallow drains. The design criteria should be systematically formulated to ensure that the resulting standard data sets address the required characteristics.

The elements in the applicability assessment test data sets representing the physical system include the hydrogeologic configuration, system geometry, and host material properties. Each of these applicability elements can be difficult to implement depending upon their complexity and code functions. For example, elements of the hydrogeologic configuration which might cause problems in some codes include: 1) temporally and spatially varying stresses (*e.g.,* areal recharge due to precipitation, ET); 2) sloping layers; 3) aquifer or aquitard pinch out; 4) strong heterogeneity (*e.g.,* low permeability lenses in high-permeability formations, or high permeability channels in moderate to low permeability formations); and 5) highly anisotropic conditions. Applicability issues for system geometry include: 1) irregular model boundaries (*i.e.,*non-linear model boundary conditions), 2) sloping base (*i.e.,*variable thickness/transmissivity, aquifer/aquitard pinch out); and 3) internal boundary conditions (*e.g.,* specified flux, no flow cells, etc.).

Table 3-11. Generic applicability assessment table.

Test Cases	**Test Elements**[1]																					
	Physical system										Management/engineering design											
	System geometry			Stresses			Soil characteristics				Flow control			Water supply			Protection, planning			Remediation		
	1	2	3	1	2	3	1	2	3	4	1	2	3	1	2	3	1	2	3	1	2	3
1																						
2																						
3																						
4																						
5																						
6																						
7																						
8																						
9																						
10																						

1) Numbers 1, 2...,4 in columns indicate specific test issues, to be discussed in test report.

The applicability of a simulation code to different management and engineering scenarios is typically controlled by three groups of elements: 1) modification of hydrogeological characteristics (*e.g.,* enhancement or reduction of permeability; 2) implementation of hydraulic controls (*e.g.,* operation of sources and sinks, placement of barriers, imposed hydraulic gradients); and 3) modification of chemical characteristics (*e.g.*, introduction of nutrients and electron acceptors in bioremediation schemes).

3.4.4. Code Testing Strategy

The code testing strategy represents a systematic, efficient approach to the comprehensive testing of the code. The code testing strategy includes the following elements:

- Formulation of *test objectives* (as related to code functionality), and of *test priorities* (based on the performance issues identified in the functionality analysis and on available resources for testing);
- Selection and/or *design of test problems* and determination of *type and extent of testing* for selected code functions or application-dependent combinations of code functions;
- Determination of level of effort to be spent on *sensitivity analysis* for each test problem;
- Selection of the qualitative and quantitative *evaluation measures* to be used in the evaluation of the code's performance; and
- Determination of the *level of detail* to be included in the test report and the *format of reporting* (see section on reporting at the end of this chapter).

Typically, test cases are based on the selection of adequate benchmarks, representative hypothetical situations, or independently observed laboratory experiments or field systems. An efficient testing strategy combines the tests required for the functionality, performance, and applicability evaluation in an efficient manner, minimizing the number of test problems considered and the simulation runs made for each test problem. Therefore, the code testing protocol is implemented using a three-level *code testing strategy*.

At Level I, a code is tested for correctness of coded algorithms, code logic and programming errors by: 1) conducting step-by-step numerical walk-throughs of the complete code or through selected parts of the code; 2) performing simple, conceptual or intuitive tests aimed at specific code functions (Test Type 1 or Level 1A Testing; see Figure 3-6); and 3) comparing with independent, accurate benchmarks (Test Type 2 or Level 1B Testing; *e.g.*, analytical solutions or hand calculations). If the benchmark computations themselves have been made using a computer code, this computer code should, in turn, be subjected to rigorous testing by comparing computed results with independently derived and published data.

At Level II, a code is tested to: 1) evaluate functions not addressed at Level I; and 2) evaluate potentially problematic combinations of functions. At this level, code testing is performed by intracomparison (*i.e.*, comparison between runs with the same code using different functions to represent a particular feature) and intercomparison (*i.e.*, comparison between different codes simulating the same problem). Typically, synthetic data sets are used representing hypothetical, often simplified ground-water systems (Test Type 3 or Level 2 Testing).

At Level III, a code (and its underlying theoretical framework) is tested to determine how well a model's theoretical foundation and computer implementation describe actual system behavior, and to demonstrate a code's applicability to representative field problems. At this level, testing is performed by simulating a laboratory (Test Type 4 or Level 3A Testing) or field experiment (Test Type 5 or Level 3B Testing) and comparing the calculated and independently observed cause-and-effect responses. Because measured values of model input, system parameters and system responses are samples of the real system, they inherently incorporate measurement errors, are subject to uncertainty, and may suffer from interpretive bias. Therefore, this type of testing will always retain an element of incompleteness and subjectivity.

First, Level I testing is conducted (often during code development) and, if successfully completed, followed by Level 2 testing. The code may gain further credibility and user confidence by being subjected to Level 3 testing (*i.e.*, field or laboratory testing) and well-conducted, field demonstrations or routine field applications (Test Type 6 or Level 3C Testing). Level 1 and Level 2 testing is sometimes referred to as "verification." The selected conceptual and verification tests are designed and described in terms of test objectives (as related to code functions), problem description (including boundary conditions), input data, and numerical discretization and solution parameters.

Although, ideally, code testing should be performed for the full range of parameters and stresses the code is designed to simulate, in practice this is often not feasible due to budget and time constraints. Therefore, prospective code users need to assess whether the documented tests adequately address the conditions expected in the target application(s). If previous testing has not been sufficient in this respect, additional code testing may be necessary.

3.4.4.1. *Test Types*

Conceptual or *intuitive tests* use highly simplified problems which have intuitive or “obvious” solutions. For saturated zone testing, these tests are often based on gradient analysis, symmetry considerations, simple application of Darcy's law and computation of mass balances. In general, these solutions are qualitative in nature. They are mostly used during the development of a code to test code sections, subroutines, and local algorithms. This type of testing, although often used, is seldom documented in a published form. Sometimes, very simple analytical solutions are used for this purpose, such as solutions for one-dimensional steady-state flow in various aquifer types subject to simple boundary conditions. Because, in most cases, an independently obtained solution is not available, conceptual tests are not considered benchmarks. They are very useful for testing in the early stages of the development of complex codes with many features, functions and options, as well for reviews of a code’s capabilities and performance.

In ground-water modeling, *benchmarks* are often represented by closed-form solution to the governing partial differential equation (*i.e.*, analytical solutions in terms of piezometric head, ground-water flux, seepage velocity, travel times, capture zones, concentration, or solute flux). The numerical model to be tested provides solutions to the same equation at a limited number of discrete points in space and time. Assuming that the coding is correct and the problem conceptualization and model setup is optimal, differences between the system responses described by the analytical solution and the numerical solution of the governing equation are due primarily to the approximate nature of the numerical method involved and to the limitations in computer accuracy, and are generally not randomly distributed. In many instances, the magnitude of these differences is related to the resolution in the discretization used in the computational scheme (Lapidus and Pinder, 1982). Theoretically, if the resolution increases such that the spatial and temporal step sizes approach zero, the differences between the numerical and the closed-form solution should disappear. In practice, due to computer round-off errors and discretization trade-offs, some measurable differences prevail.

It should be noted that, if a computer code implementation of the analytical solutions has been used in this type of testing, the resulting analytical modeling code should first be subject to appropriate testing. Often, analytical solutions are presented in the form of complicated integrals, which, in turn, need to be numerically evaluated either by series approximation or numerical integration *(e.g.,* the well function in the Theis equation). Verification of a coded analytical solution is restricted to comparison with independently calculated results using the same mathematical expression; *i.e.,* manual calculations, comparison with the results from computer programs coded independently by third party programmers, or using general mathematical computer software systems such as Mathematica®[1] and Mathcad®[2]. One of the most common approaches to check the numerical evaluation of analytical solutions is performing hand calculations using published values of the approximated functions.

Often, when more complex code functionality issues need to be assessed, appropriate analytical benchmark solutions are not available. In such cases, Level 2 benchmarking may be more appropriate. Unlike Level 1 testing which yields quantitative intercomparison results and may be considered a rather "objective" form of code testing, Level 2 benchmarking is more subjective. Level 2 testing uses test problems for which the solution is basically unknown. The results of Level 2 testing are inspected for “obvious” problems, such as physically inappropriate behavior, mass balance errors, instability and slow or non-convergence. Often, the results obtained with the test code are compared with those obtained with another, comparable numerical model using high-resolution spatial and temporal discretization schemes. If major differences between the codes occur, the results of one or both codes might be incorrect. On the other hand, when the results for a well-designed Level 2 test are (almost) identical, both codes gain in credibility. As the absolute "truth" for these hypothetical problems is unknown, only a comparative verification of a model can be obtained. Using this approach provides a "relative" benchmark. This form of testing can be used to study the treatment of a number of naturally occurring conditions, including various hydrogeologic conditions (such as aquifer stratification and heterogeneities), physico-chemical processes and ranges of their respective

[1]Registered Trademark of Wolfram Research Inc., Champaign, Illinois.

[2]Registered Trademark of Mathsoft, Inc., Cambridge, Massachusetts.

parameters, boundary and initial conditions, large variations in the gradient of the dependent variable *(e.g.,* solute fronts), and sources and sinks. Some of the conditions are summarized in Table 3-12.

Level 2 test problems should be solved using a critical range of Peclet and Courant numbers. Accurate numerical solutions should be generated using codes that are known to effectively handle critical parameter values, high resolution numerical grids, and small time steps. This approach is based on the idea that the smaller the discretization is in space and time, the better the approximate numerical solution will represent the real (unknown) solution of the governing partial differential equation (Huyakorn and Pinder, 1983). The resulting benchmarks are developed in a step-wise fashion, going from coarse resolution grids and large time steps to higher resolution grids and smaller time steps. After each run, computational differences should be evaluated. When further refinement, for example with a factor 2, does not provide significant changes in the computational results, the relative benchmark is established. If the simulation results in a Level 2 code intercomparison test do not deviate significantly, the "relative" or "comparative" test is considered successful. However, if significant differences occur, in-depth analysis of the results of simulation runs, performed with both codes, should be performed.

At Test Level 3, the model (and its code) is compared with independently obtained field or laboratory data, determining the "degree of correlation" between calculated and independently observed cause-and-effect responses (van der Heijde and Elnawawy, 1992). This type of testing is sometimes referred to as "field or laboratory validation." The role of Level 3 testing in the protocol is two-fold: 1) determining how well a model's theoretical foundation and computer implementation describe actual system behavior; and 2) assessment of a code's applicability to real-world systems and management problems. The first goal is met by both laboratory experiments (Test Type 4) and field experiments (Test Type 5); the second goal is met by comparing modeling results with high-quality field experiments and successful field applications (Test Type 6). However, evaluation of successful field applications is not incorporated in the testing strategy. It should be noted that the actual measured data of model input, system parameters and system response are samples of the real system and inherently incorporate errors (NRC, 1990). An additional complexity is that often the data used for field validation are not collected directly from the field but are processed in an earlier study.

Program Name:	HOTWTR
Program Title:	Simulating Coupled Three-Dimensional Steady-State Ground-Water Flow and Heat Transport in Saturated Media
Version:	1.1
Release Date:	September 1993
IGWMC Number:	FOS 67
Institution of Development:	U.S. Geological Survey, Denver, Colorado

TEST 03D:

Geometry: multi-layer profile model (2-D cross-sectional); homogeneous aquifer of 13 by 1 cells horizontally, and 10 layers

Processes: internal heat conduction; heat conduction through overburden to land surface; no ground-water flow

Boundary conditions: given heat flux condition at lower boundary (natural geothermal gradient at bottom boundary; second-type b.c.); fixed temperature at opposite lateral boundaries (first-type b.c.); given temperature at surface boundary (third-type b.c.); no areal ground-water recharge from precipitation; no pumping or injection of water in wells; zero ground-water flux at lower, lateral, and upper boundaries.

Objective: to qualitatively evaluate conductive heat flow through aquifer resulting from first-, second- and third-type heat flow boundary conditions.

Results: Problem has zero ground-water flow; heat in-flux occurs along lower and upper boundaries, and along upper part of high temperature boundary; heat out-flux occurs along lower part of high temperature boundary and along low temperature boundary (see contour graph).

Evaluation: results are conform expected behavior (qualitative conceptual test).

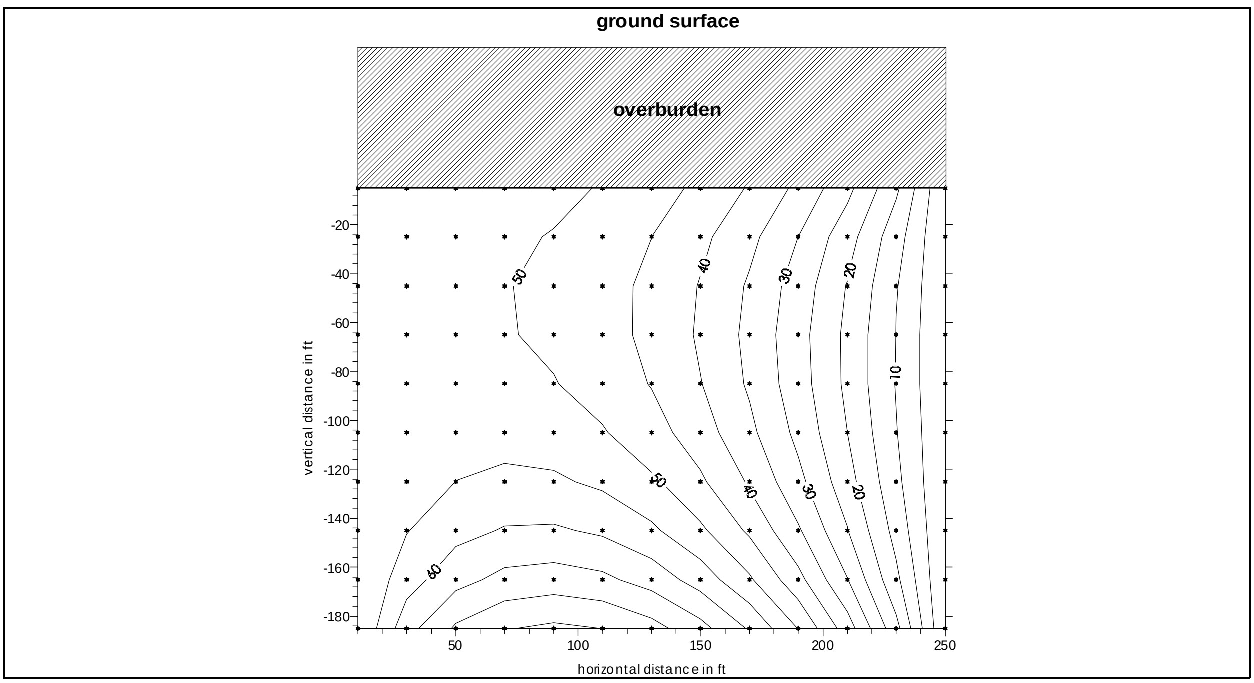


Figure 3-6. Example of a conceptual test problem: temperature distribution in a homogeneous aquifer.

Therefore, they are subject to inaccuracies, loss of information, interpretive bias, loss of precision, and transmission and processing errors, resulting in a general degradation of the data to be used in this type of testing.

Table 3-12. Example test scenario for three-dimensional solute transport codes (from van der Heijde and Elnawawy, 1992).

1. Solute transport in a steady-state uniform flow field in a large homogeneous isotropic aquifer (conceptual and analytical solutions are available);
   - 1.1. advection only (various boundary conditions, source locations, source strength)
   - 1.2. advection and dispersion (various boundary conditions, source locations, source strength, various ratios for longitudinal and transverse dispersion)
   - 1.3. advection, dispersion and decay
   - 1.4. advection, dispersion, and retardation
   - 1.5. advection, dispersion, decay, and retardation

2. Solute transport to sink in a non-uniform steady-state flow field in a large homogeneous aquifer (analytical solutions available);
   - 2.1. advection and dispersion for various source/sink scenarios

3. Solute transport in a non-uniform flow field in a large homogeneous aquifer (analytical solutions not available):
   - .1. steady-state flow field:
     - 3.1.1. different solute source/sink conditions
     - 3.1.2. different boundary conditions
   - 3.2. non-steady flow field with:
     - 3.2.1. constant source rates
     - 3.2.2. time-varying source rates
     - 3.2.3. time-varying boundary conditions

4. Non-uniform flow field in a heterogeneous anisotropic aquifer (no analytical solutions available):
   - 4.1. layered system
     - 4.1.1. steady-state flow field
       - 4.1.1.1. sources/sinks in various layers
       - 4.1.1.2. different boundary conditions
     - 4.1.2. non-steady flow field with:
       - 4.1.2.1. sources/sinks in various layers
       - 4.1.2.2. different boundary conditions
   - 4.2. lens heterogeneities
     - 4.3. random heterogeneities

3.4.4.2. *Potential Problems in Code Testing*

There are some potential pitfalls associated with the functionality testing procedures. Differences between the ground-water code being tested and the benchmark solution may have various reasons, such as:

- the assumptions made in developing the simulation code may differ from those made to derive the benchmark solution;

- the level of discretization used in testing a numerical code;
- the mathematical nature of the governing partial differential equation;
- the methods involved in obtaining a numerical solution;
- the limitations in computer accuracy; and
- limitations in accuracy (or even errors) of the benchmark solution implementation.

Furthermore, the magnitude of some of these numerical differences can be related to the resolution in the spatial and temporal discretization used in the computational solution scheme (Lapidus and Pinder, 1982). In theory, if the benchmark solution uses a closed-form solution of the governing partial differential equation, the differences between the numerical and the closed-form solution of a particular mathematical problem (*i.e.*, governing equations, and boundary and initial conditions) should become negligible as spatial and temporal step-sizes approach zero. Overall, residuals between analytical and numerical results tend to decrease when the spatial discretization is increased near localized aquifer stresses (van der Heijde *et al.*, 1993). This is also true for temporal discretization refinement directly after a change in stresses (*e.g.*, Prickett and Lonnquist, 1971). In general, if the simulation code is free of errors, and functionality has been correctly established, any deviation from the benchmark should be attributable to grid discretization and computer precision issues. Consequently, test problems should be carefully designed to minimize deviation due to discretization issues to increase the effectiveness and quality of the test case.

3.4.5. Test Evaluation Tools

An important aspect of code testing is the definition of illustrative, informative and efficient measures. Typically, such measures are statistical or graphical in nature. Acceptance of code testing results to date has been primarily based on visual inspection of the graphical representation of the dependent variable as computed with the simulation code and the benchmark solution (see Figure 3-7). Although graphical comparison is an appropriate measure, acceptance should also be based on quantitative measures of the goodness-of-fit. There are three general procedures, coupled with standard linear regression statistics and estimation of error statistics, to provide such quantitative code performance assessment (Donigian and Rao, 1986).

- paired-data performance -- the comparison of simulated and observed data for exact locations in time and space;
- time and space integrated, paired-data performance -- the comparison of spatially and temporally averaged simulated and observed data;
- frequency domain performance -- the comparison of simulated and observed frequency distributions.

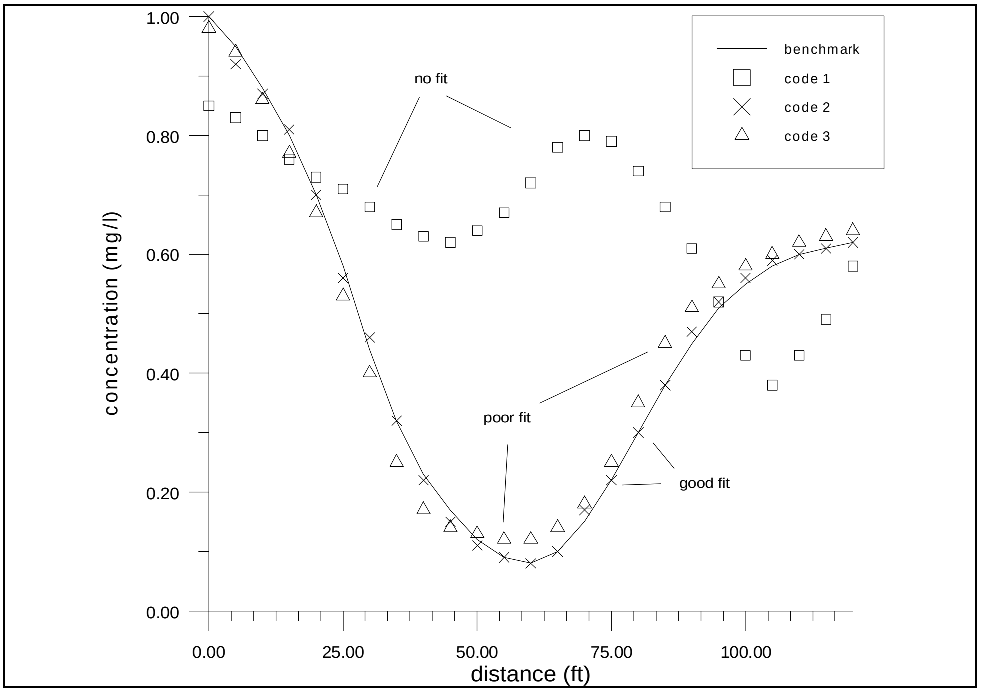


Figure 3-7. Visual inspection of goodness-of-fit between benchmark and tested models.

Of these three methods, paired-data analysis is the most appropriate technique for use in the code-testing protocol. Intercomparison of data generated at the same point in time and space provides the most explicit and objective analysis. Using spatially averaged or integrated representations, or frequency distributions of the test variable for the intercomparison analysis can result in biased or subjective analyses due to undesirable data smoothing and weighting.

These paired-data intercomparison results are best manipulated, calculated, and analyzed using computer-aided techniques. Spreadsheet software is well-suited for the reduction of protocol results because it provides a variety of data editing, analysis and graphing capabilities for both spatially and temporally distributed data generated during code-testing. Often, the understanding of the data processed in spreadsheet software can further be enhanced by using line graph, contour graph, surface display or animation software.

Typically, test variables for saturated flow codes include hydraulic head (in space and time), head gradients, global water balance and segmented internal or boundary fluxes, flow velocity patterns (direction and magnitude), flow path lines, capture zones, and travel times. For solute transport codes, performance evaluation will focus on the spatial concentration distribution of the tracer of interest, the global mass balance (per species) and specific mass fluxes, and breakthrough curves at observation points and sinks (wells, streams).

3.4.5.1. *Statistical Evaluation Techniques*

The code-testing protocol employs a series of statistical measures, called *evaluation or performance measures*, to characterize quantitatively the differences between the results derived with the simulation code and the established benchmark, or between the results obtained with two comparable simulation codes (van der Heijde and Elnawawy, 1992). Some of these measures are comparable to the measures typically used in the calibration of site-specific simulation models (Anderson and Woessner, 1992). The main statistical measures, included in the code testing protocol, are mean error, mean absolute error and root-mean-squared error. Variations of these common measures, such as positive and negative mean error, and the ratios between them, can also be valuable in evaluating code-testing results. In addition, simple quantitative measures such as minimum and maximum deviation, and their spatial location within the model domain, can provide meaningful information on code performance.

The organization and evaluation of code intercomparison results can be cumbersome due to the potentially large number of data-pairs involved, specifically if every computational node is included in the analysis. This can be mitigated by analyzing smaller, representative sub-samples of the full set of model domain data pairs. The representativeness of the selected data pairs is often a subjective judgment. For example, in simulating one-dimensional, uniform flow, the data pairs should be located at least on two lines parallel to the flow direction, one in the center of the model domain and one at the edge to capture the effects of asymmetrical results due to the used solver (see Figure 3-8a). Another example is the simulation of the Theis problem using a finite difference formulation in Cartesian coordinates; here, two lines of data pairs should be chosen parallel to the two horizontal principal hydraulic conductivity axes, while a third set of data pairs should be on a line under 45 degrees with these axes to address effects of the rectangular grid on the radial-symmetric response of the aquifer on the imposed stress (see Figure 3-8b). Test cases that are symmetrical can be analyzed for a smaller portion of domain based upon the type of symmetry present. For example, test cases that have radial symmetry can be divided into four equal representative radial slices; this can significantly reduce the number of data pairs in the analysis and simplify the analysis considerably.

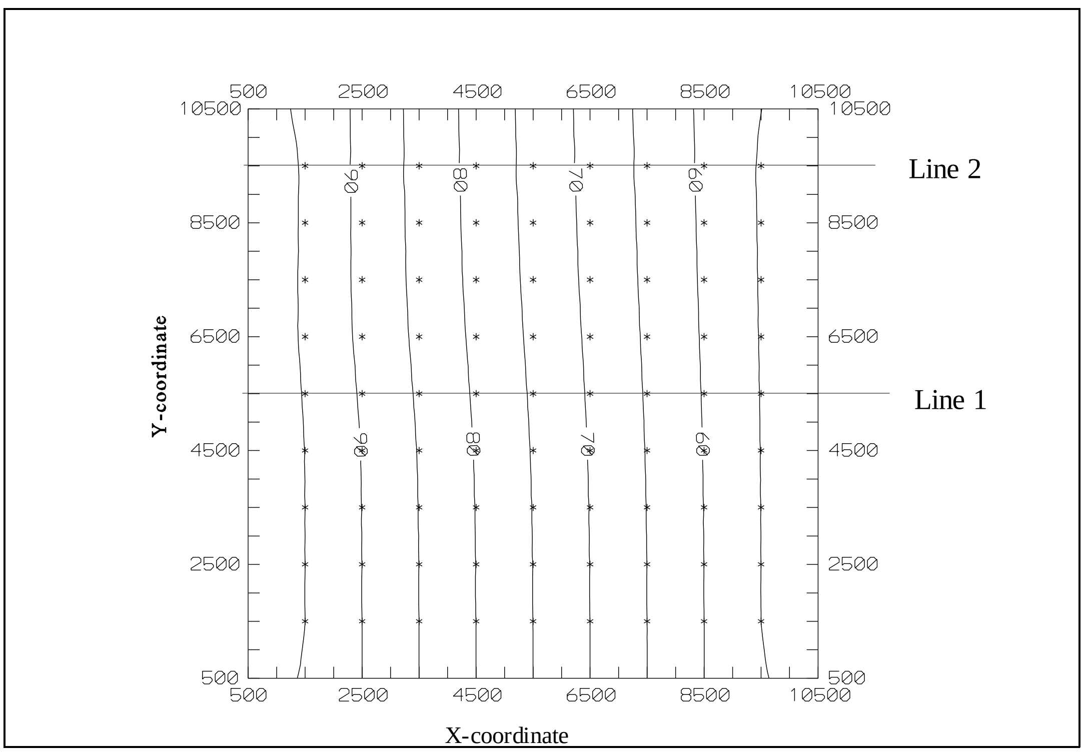


Figure 3-8a. Representative sets of spatially-defined data pairs for intercomparison: one-dimensional, uniform flow case.

As part of the measurement and analysis of paired-data, it is important to define a sign convention to ensure standardization. The measures used in the developed protocol are *positive* when the simulation code under investigation exceeds, or *overestimates*, the benchmark solution. Contrarily, a *negative* statistical measure indicates a situation where the simulation code *underestimates*, or generates results that are less than those of the benchmark solution.

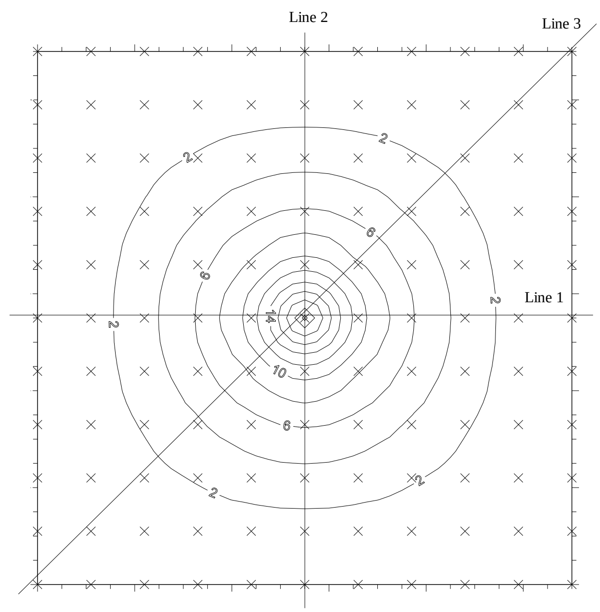


Figure 3-8b. Representative sets of spatially-defined data pairs for intercomparison: radial, confined flow case.

The statistical measures used in the testing protocol are organized, discussed, and briefly illustrated in the following sections. Each statistical measure is individually described and defined. Although ***h***, which generally denotes hydraulic head, is used in the following expressions as the symbolic notation for the dependent variable, it may represent any other dependent scalar variable of interest (*e.g.*,contaminant concentration, directional ground-water velocity).

The first paired-data measure used as an evaluation tool in the protocol is the *Deviation Coefficient* (DC). It can be calculated at any single point in space or in time by using the following expression (ASTM; 1984):

$$DC\ \% = [\ (h_{nm} - h_{am})\ /\ h_{bm})\ ]\ * 100 \tag{3.10}$$

where h_{nm} is the value of the dependent variable calculated by the numerical model, and h_{bm} is the value of the dependent variable calculated with the benchmark solution (*e.g.*, analytical model). To gain a more general measure of code intercomparison, the *Average Deviation Coefficient* (ADC) can be calculated for the entire model domain. The ADC is calculated for every point in the model domain and then averaged:

$$ADC\ \% = \frac{1}{n} \sum_{i=1}^{n} [\ (h_{nm} - h_{bm})_i\ /\ (h_{bm})_i\ ]\ * 100 \tag{3.11}$$

where i is the individual model point, ranging from 1 to n, n the total number of calculation points (data pairs), and other terms are as defined for expression 3.10.

The *Mean Error* (ME) is defined as:

$$ME\ = \frac{1}{n} \sum_{i=1}^{n} (h_{nm} - h_{bm})_i \tag{3.12}$$

Because ME includes both positive and negative values which cancel each other, ME may not be the best indicator of an acceptable match (Anderson and Woessner, 1992). The *Mean Absolute Error* (MAE) may provide a better indicator of agreement between code and benchmark, because it computes the absolute value of the residuals:

$$MAE\ = \frac{1}{n} \sum_{i}^{n} \left|\ (h_{nm} - h_{bm})_i\ \right| \tag{3.13}$$

To further characterize the residuals with respect to their mathematical sign, two other measures may be used. The *Positive Mean Error* (PME) is a quantitative indicator of the overestimation of the numerical code because it analyzes only the positive residuals. It is computed by averaging the positive differences as follows:

$$PME = \frac{1}{n_{pos}} \sum_{i}^{n_{pos}} POS(h_{nm} - h_{bm})_i \tag{3.14}$$

where $POS(h_{nm}-h_{bm})_i$ is the value of the differences when $h_{nm} > h_{bm}$ and n_{pos} is the number of grid points having such positive differences. Similarly, the *Negative Mean Error* (NME) is a quantitative indicator of the underestimation of the numerical model because it analyzes only the negative residuals. It is computed by averaging the negative differences between the dependent variable values calculated by the numerical model and the benchmark solution. NME is defined such that it is always positive:

$$NME = \frac{1}{n_{neg}} \sum_{i}^{n_{neg}} NEG(h_{bm} - h_{nm})_i \tag{3.15}$$

where $NEG(h_{bm}-h_{nm})_i$ is the value of the difference when $h_{bm} < h_{nm}$, and n_{neg} is the number of model points having such negative differences. When used alone, the PME and NME measures are often inadequate. These criteria only describe how the code differs from the benchmark, they do not account for the locations where agreement to the benchmark is perfect and residuals are zero. This can be described by the *Root Mean Squared Error* (RMSE) measure. RMSE is the square root of the average of the squared differences between the values for the dependent variable calculated by the numerical model and the benchmark solution:

$$RMSE = \sqrt{\frac{1}{n} \sum_{1}^{n} (h_{nm} - h_{bm})_i^2} \tag{3.16}$$

As defined above, these measures provide the protocol user with an estimate of the overall, or average, difference between the simulation code results and the benchmark solution. However, these measures can be more useful to protocol users if they are reported as a percentage of the originally calculated dependent variable. For example, if a simulation code predicts a maximum total drawdown of 30 feet in an aquifer subject to pumping and the calculated RMSE is 1.5 feet, then the protocol user may be better able to relate these two values if the RMSE is also reported as a *Relative Error* (RE) of five per cent (*i.e.*, 1.5 divided by 30). RE can be calculated for any of the measures discussed according to:

$$RE\ \% = \frac{Measure}{h_{nm}^{\max}} * 100 \tag{3.17}$$

where *Measure* is the statistical measure of choice, and $h_{nm}^{\max}$ is the maximum value calculated by the numerical code. The use of relative error measures can effectively characterize the amount of overall error or residual which can be attributed to a ground-water simulation code. This provides a measure for the entire simulation and differs from the DC which is a measure of error relative to the value of the system at a single measurement point.

To further describe the nature of the agreement between the numerical model and the associated benchmark, a new mathematical ratio called *Mean Error Ratio* (MER) was used in this code-testing study. The MER quantifies the comparative agreement of the code being tested in terms of under- or overestimation. The value of the MER may be either positive or negative. Positive MER values represent situations where the PME equals or exceeds the NME; in these cases the MER has a value of 1.0 or greater and the MER indicates the magnitude or degree of over- or underestimation of the code being tested:

$$MER = \frac{|ME|}{ME} \frac{PME}{|NME|} \qquad (for\ \ |NME| < PME) \tag{3.18}$$

When the MER is equal to one, the NME equals the PME and the amount of positive deviation from the benchmark is equal to the negative deviation from the benchmark. When the NME exceeds the PME, the MER is negative and indicates the degree of underestimation of the code being tested:

$$MER = \frac{|ME|}{ME} \frac{|NME|}{PME} \qquad (for\ \ |NME| > PME) \tag{3.19}$$

3.4.5.2. *Graphical Evaluation Techniques*

As part of the code-testing protocol, this section presents a set of graphical evaluation tools to effectively analyze and clearly and concisely illustrate code-evaluation results. Graphical techniques are especially significant for test results that do not lend themselves to statistical analysis. For example, graphical representation of solution convergence characteristics may indicate numerical oscillations and instabilities in the iteration process. As is the case with the computation of statistical measures, practical considerations may prevent the use of all generated data pairs when using graphical techniques. Often, a representative or illustrative subset of data pairs may be selected for use with graphical evaluation techniques of code performance. The selection of a set of representative sample data pairs may be based on symmetry considerations, or focused on model domain areas with potential higher deviations or other specific test issues (*e.g.*, vertical or horizontal slices of the model domain).

Graphical representation of test results should include graphs of the dependent variable(s), the comparison deviations (or residuals), and other computed entities (*e.g.*, mass balance, aquifer-stream fluxes) versus distance and, if appropriate, versus time. Two-dimensional graphs depicting the spatial distribution of each dependent variable and the deviations in that variable may also prove useful for evaluation of code testing results (van der Heijde and Elnawawy, 1992). Such spatial graphs may cover the entire model domain, or focus on a specific subregion(s). In general, the conclusions from visual inspection of graphic representations of testing results are described qualitatively using, for example, such terms as "poor," "reasonable," "acceptable," "good," and "very good" (Beljin, 1988).

Most of the graphical analyses used in previous code testing studies have typically utilized simple line graphs, (*e.g.*, head versus time or head versus linear distance). Multi-dimensional graphs that illustrate the areal distribution of dependent variables (for example, contoured hydraulic heads or residuals in X-Y space) have also been used to support code performance tests. Expanding the application of multi-dimensional graphical techniques in the code-testing process will enhance the visual judgment of residuals, deviations, and goodness-of-fit (van der Heijde and Elnawawy, 1992). Tables 3-13 and 3-14 provide an overview of recommended graphical evaluation techniques. They are discussed in detail in the following paragraphs.

The protocol specifies five types of graphical evaluation techniques (see table 3-14):

1) X-Y plots or line graphs of spatial or temporal behavior of dependent variable and other computed entities;
2) one-dimensional column plots or histograms (specifically to display test deviations);
3) combination plots of line graphs of dependent variable and column plots of deviations;
4) contour and surface plots of the spatial distribution of the dependent variable; and
5) three-dimensional, isometric, column plots or three-dimensional histograms.

Table 3-13. Overview of graphical code testing evaluation techniques.

Type of variable	Type of graph	Optional graph
distribution of the dependent variable in space and time	line graph versus distance for selected times, line graph versus time for selected locations, two-dimensional contour plot, two-dimensional histograms	two- and three-dimensional iso-surfaces
distribution of deviations in the dependent variable in space and time	line graphs versus distance for selected times, line graph versus time for selected locations, two-dimensional contours (for large number of nodes), two-dimensional histograms	
combination graphs	line graph of dependent variable and deviations versus distance/time	
global mass balance	line graph versus time	
iteration error	line graph versus number of iterations for selected times	

X-Y plots are very useful in illustrating the general shape of the solution in terms of the dependent variable of interest, and to obtain an impression how major differences between the results obtained with the tested code and the benchmark relate to the shape and values of the solution. This is the conventional approach used in most code-testing efforts. These commonly used plots are also very helpful in sensitivity analysis, which is a significant part of the performance evaluation procedure of the code testing protocol. An example of this display technique is shown in Figure 3-9. It is obvious from the graph that for shorter distances and higher values of the dependent variable the tested code is underpredicting, while for longer distances and lower values of the dependent variable the code is overpredicting. Furthermore, there is some oscillation in the benchmark for very short distances. This might indicate problems in generating the analytical solution for values of the independent variable near zero. X-Y graphs can be easily prepared using spreadsheet programs with graphic capability, and with dedicated scientific graphics packages.

Table 3-14. Use of graphical evaluation techniques.

Test problem dimen-sionality	**Graph Type**				
	contours of spatial distribution	line graph of spatial distribution	line graph of behavior in time	1-D histogram of spatial distribution	2-D histogram of spatial distribution
1-D	-----	yes	at selected locations	yes	-----
2-D horizontal	areal	for selected lines parallel to axes in middle of model domain and at edges and for lines under 45 degrees with axes (separate graphs for each data pair set)	at selected locations (dependent variable)	at same locations as line graph (deviations; combine with line graph for data pair set)	for rectangular grids only
2-D vertical	profile	for selected lines parallel to axes in middle of model domain and at edges and for lines under 45 degrees with axes (separate graphs for each data pair set)	at selected locations (dependent variable)	at same locations as line graph (deviations; combine with line graph for data pair set)	for rectangular grids only
radial-symmetri-cal	areal	for 2 axes and for a line under 45 degrees with the axes (combination plot of all three data pair sets in separate graphs for variable and deviation)	at selected locations	at same locations as line graph (deviations; combine, in separate graph for each data pair set, with line graph)	for rectangular grids only
3-D	selected slices and profiles	for selected lines parallel to axes and under 45 degrees angles with axes	at selected locations	at same locations as line graph (deviations; combine with line graph for each data pair set)	for rectangular grids only; same slices and profiles as used for contours
transient	at selected times	at selected times	for linear, logarithmic or user-defined time-stepping	at selected times	at selected times

Combination plots provide an excellent way to depict two types of data in one graph. For example, the results for the dependent variable, obtained with the tested code, may be plotted together with residual results (*i.e.*, deviations) to illustrate their inter-relationship. An example of a combination plot is shown in Figure 3-10, where an X-Y plot of the simulation results is overlain by a column plot of the intercomparison residuals. It should be noted that two different vertical scales (Y-axes) have been used to plot the disparate data. Figure 3-10 shows, among others, where the maximum residual occurs in relationship to the spatial distribution of the dependent variable. It also shows that all residuals are positive and they are asymmetrically distributed in space.

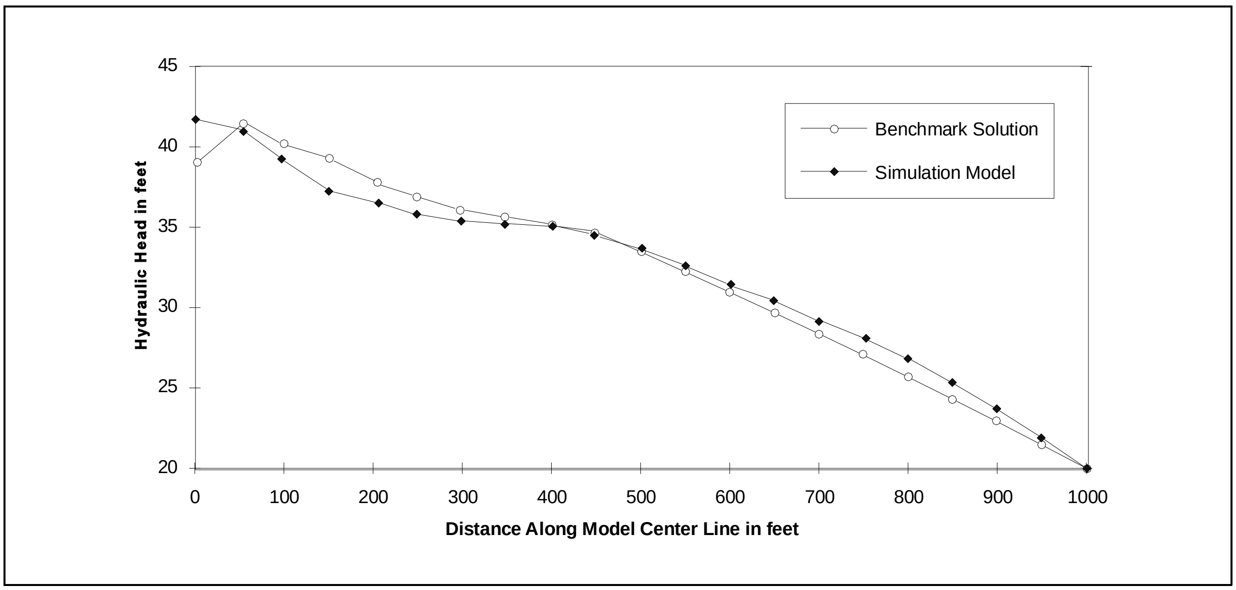


Figure 3-9. X-Y plot of dependent variable computed by tested code and benchmark.

Another, very illustrative graphic display technique is provided by three-dimensional isometric column plots or histograms. This type of plots is not a true three-dimensional technique because the data is characterized by a two space coordinate or a time and space coordinate, and some computed value, which corresponds to the Z coordinate. Isometric column plots are very effective for the depiction of layer-wise spatially distributed data sets, specifically for hydraulic heads, contaminant concentrations, and intercomparison residuals. Figure 3-11 depicts a generic isometric, column plot. It provides a rapid impression of the spatial distribution of the data set. Such a plot can be valuable in illustrating the location where maximum or minimum values occur, and the spatial extent of high values for the plotted variable. For example, this graphical technique will highlight artificial high concentrations in stagnation zones occurring when using certain random walk techniques. It is also very useful to provide a quick impression of the distribution of residuals. Isometric column plots can be produced rapidly with modern spreadsheet software. They do not require additional interpolation or smoothing and thus provide a more direct representation of the spatial distribution of a variable than, for example, two-dimensional contouring. This is especially true for simulations which use a regularly-spaced grid; results can be directly imported into the graphical spreadsheet software and plotted.

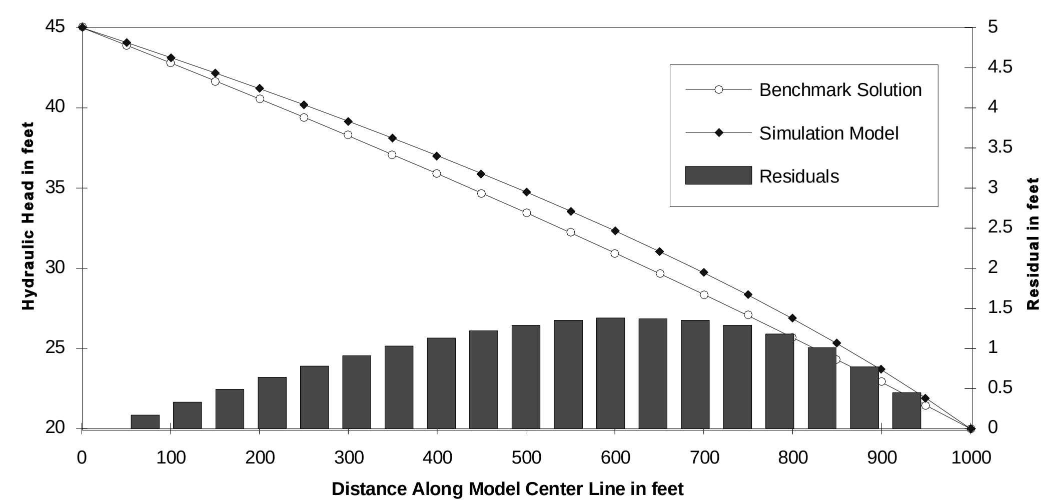


Figure 3-10. Combination plot of X-Y graph of dependent variable
and column plot of residuals.

Three-dimensional histograms have some disadvantages. Most conventional software packages will produce some level of visual distortion when variably-spaced data are plotted using isometric columns. In addition, some isometric column plots may be difficult to interpret due to their blocky, discretized nature, especially plots that represent low grid resolutions. The three-dimensional perspective and axis scales that are selected for the graphs can also visually distort the data depending upon the angles, elevations, and scales chosen. Effects of such relative distortion may be decreased by use of standardized perspective and scale. Overall, isometric graphical techniques provide an effective graphical method for data presentation and analysis; they can be used to easily identify maxima, minima, general trends, as well as potential errors in the data.

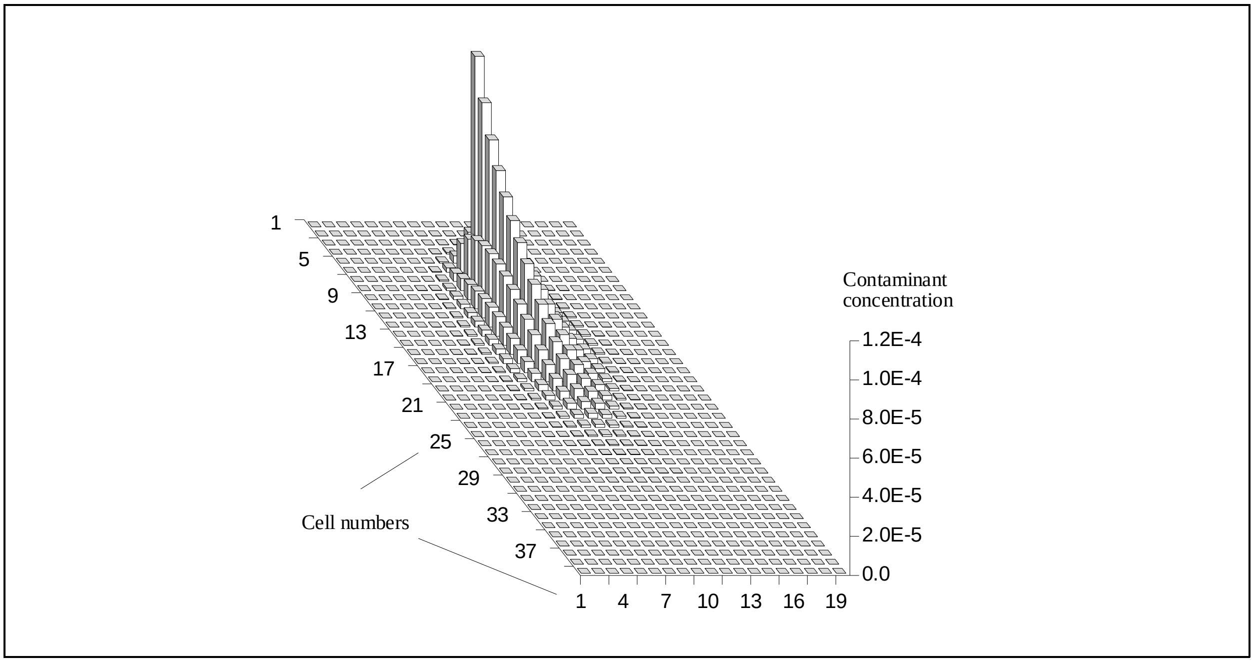


Figure 3-11. Example of an isometric column plot or three-dimensional histogram; produced with Microsoft® Excel for Windows.

Two- and (quasi-) three-dimensional contour and surface graphs provide an overview of the spatial distribution of the dependent variable (Figures 3-12, 3-13, 3-14 and 3-15). These graphs can also be used for display of the spatial distribution of benchmark deviations. They are very useful to discover irregularities in the spatial distribution of the test variables and unacceptable high deviations in computed deviations.

Contour maps are two-dimensional graphs of lines of equal value (contours) of a variable defined in two dimensions (Figure 3-14). Surface graphs are three-dimensional graphs of the distribution of a variable defined in two dimensions (Figure 3-12). If the surface formed by the variable is represented by lines parallel to the horizontal axes of the graph, it is called a wire mesh plot; if the surface is represented by contours of the variable, it represents a series of slices. A wire mesh plot can be combined with contours into a single plot (Figure 3-12). Figure 3-13 shows a series of slices

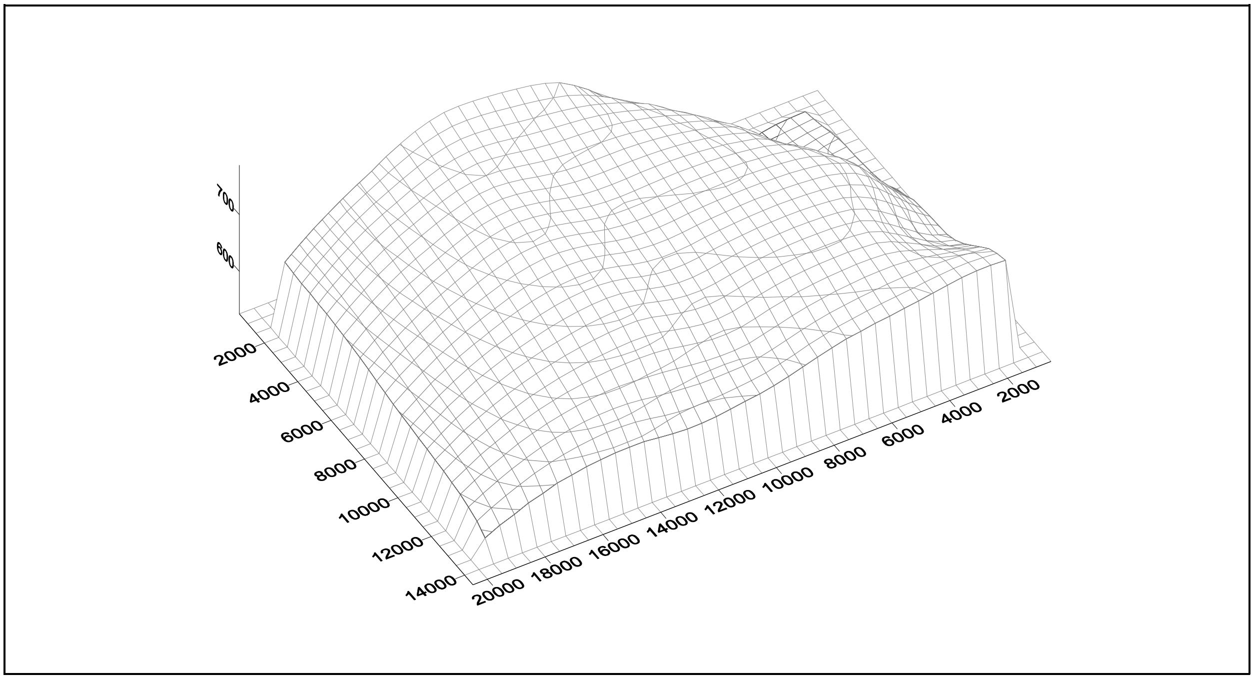


Figure 3-12. Surface plot of hydraulic head showing wire mesh and contour lines; produced with Golden Software's Surfer® for Windows.

representing lines of equal value of the variable of interest, filled in to produce a solid surface. This figure also shows the combination of a quasi-three-dimensional presentation with a regular two-dimensional contour plot.

Contour maps and surface plots are well-suited for qualitative assessment of test results. However, many user-introduced decisions may significantly alter the representation of computational results using these graphs. For example, smoothing provides a graph which may highlight main features of the response surface, but hide some irregular computational behavior (Figure 3-14). Also, the method of interpolation in contouring programs is subject to user-manipulation. Figure 3-15 shows some options available from a widely used commercial contour and surface graphing program. Except for the graph prepared with the method of inverse distance using a power equal to 3, all graphs have been produced using default program settings.

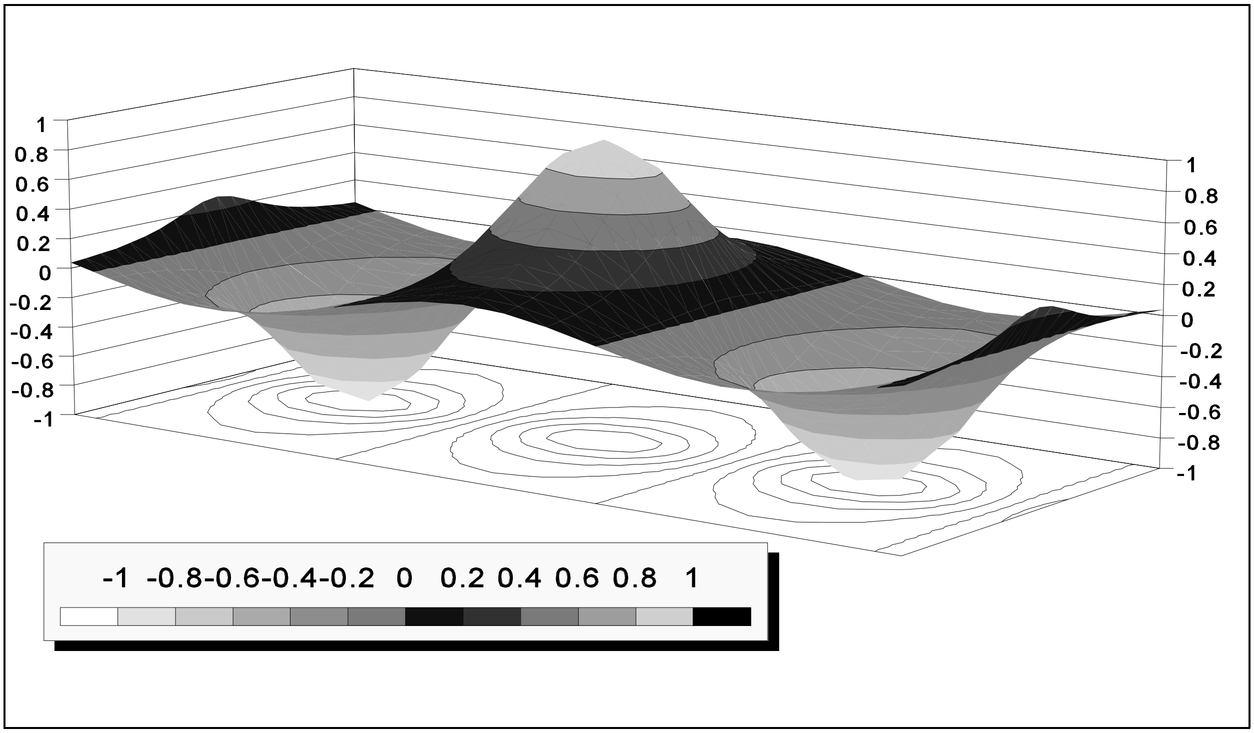


Figure 3-13. Combination plot of solid surface and projected contours; prepared with DeltaPoint's DeltaGraph® for Windows.

3.4.5.3. *Notes on the Use of Evaluation Tools*

This section illustrates the use of the statistical evaluation techniques, in combination with the graphical techniques used in the code-testing protocol. The examples illustrate effectiveness and ineffectiveness of various measures and techniques in case of persistent overestimation, persistent underestimation, and a spatially-characterized combination of both.

The first example illustrates how the statistical and graphical evaluation tools can be combined to identify the case where the numerical simulation code overestimates the benchmark solution. Figure 3-16 shows the graphic comparison of the results obtained with the numerical code plotted against the benchmark solution. In addition, this figure includes the statistical measures ME, NME, RMS, and ADC for the comparison of the two data sets. In this case, the simulation code overestimates the benchmark solution at almost every point along the center line of the model

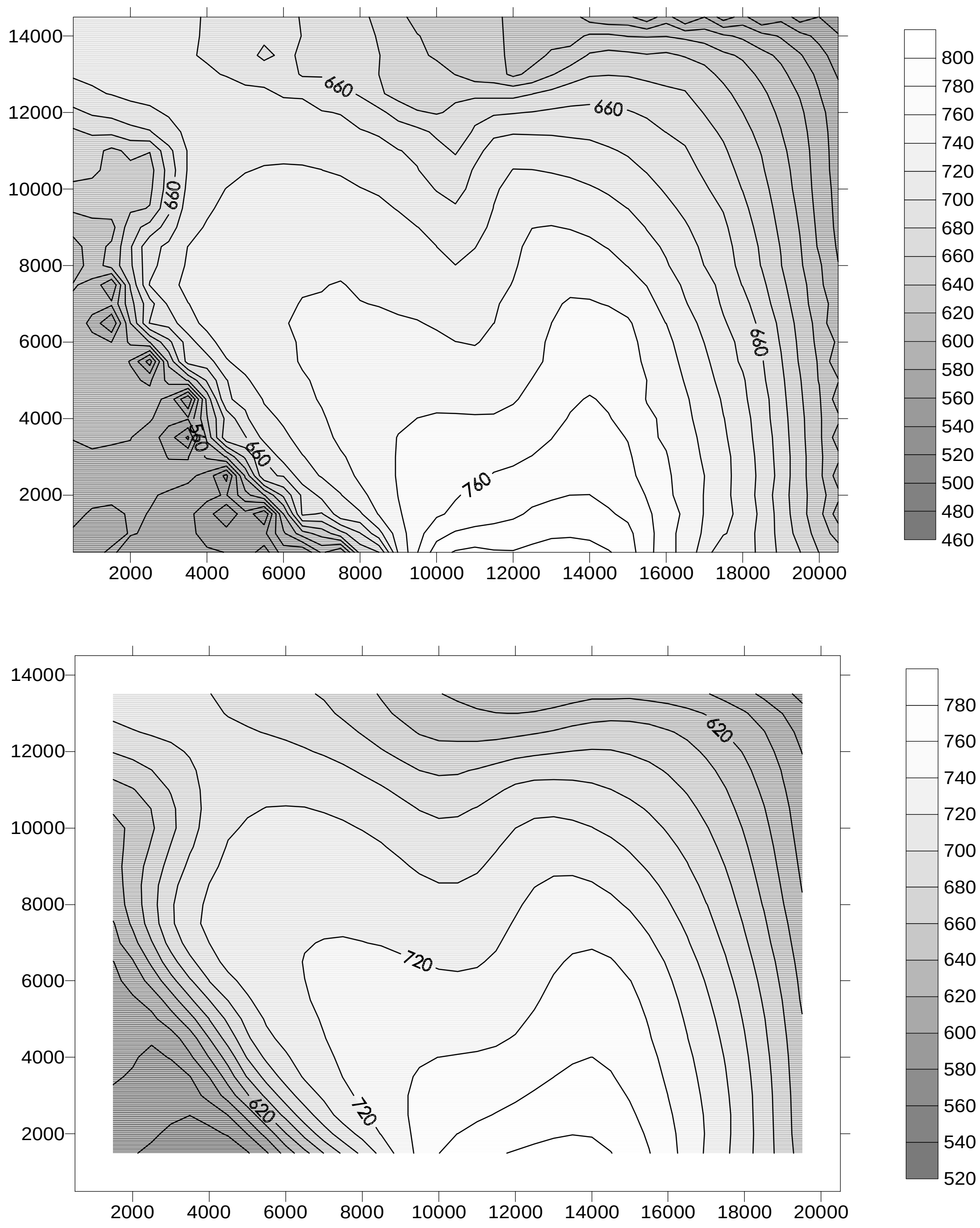


Figure 3-14. Contour plots of hydraulic head showing effects of smoothing of interpolation grid; prepared with Golden Software's Surfer® for Windows.

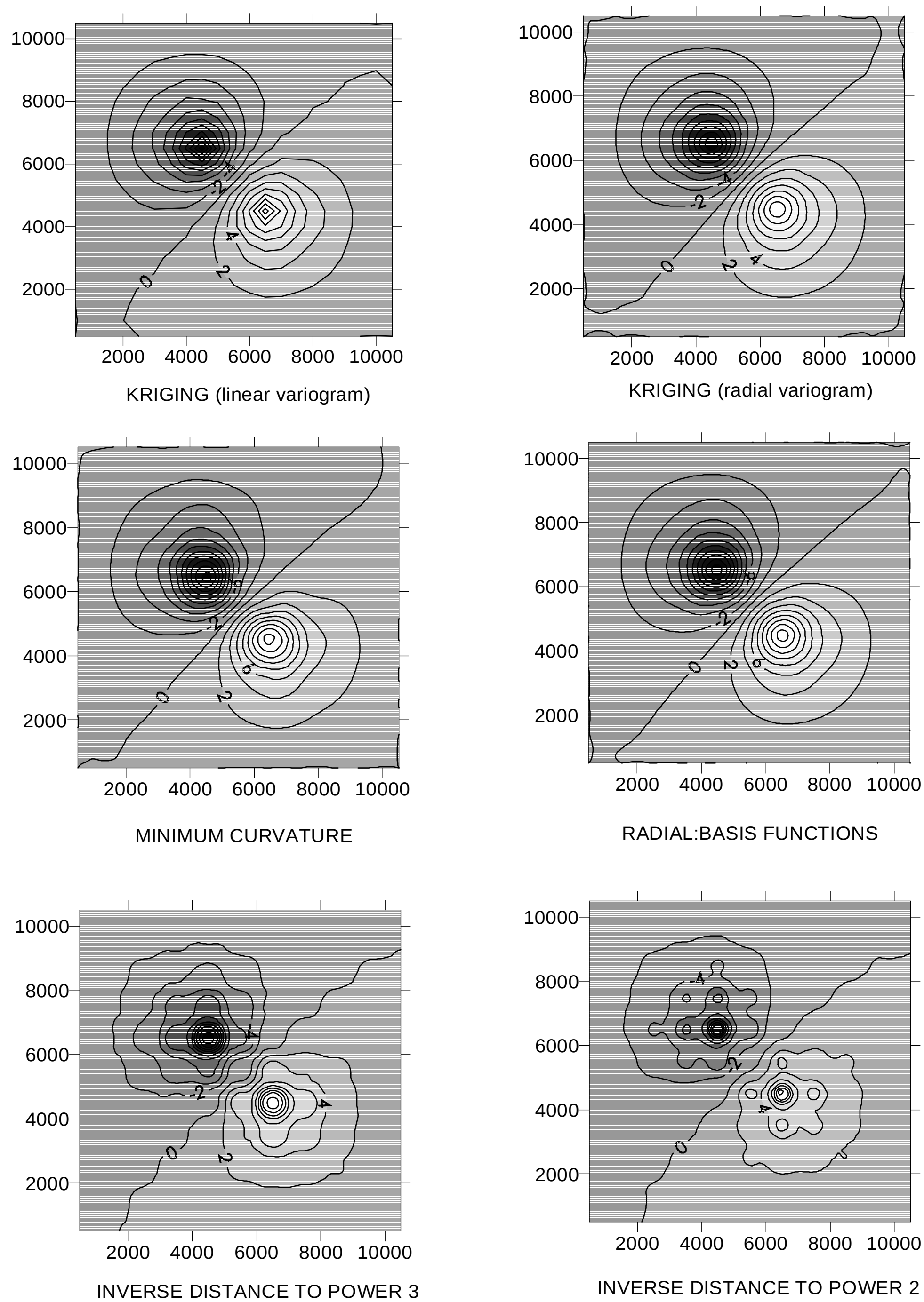


Figure 3-15. Contour maps of drawdown caused by injection-pumping well pair showing effect of grid interpolation algorithm; prepared with Golden Software's Surfer® for Windows.

domain. All of the residuals are positive and the statistical measures reflect this. The ME, MAE, and the PME are all identical and equal to 0.82 feet. Because all of the residuals are positive, the NME and the MER are not applicable measures. The ADC is 2.8 per cent. Additional information and conclusions may be drawn from inspecting the graph. The plot clearly shows that the agreement is greatest at the edges of the model domain, which may be an artifact of the closeness to specified boundary conditions. It can also be seen that there is a non-symmetric distribution of residuals which may be a significant indication of code performance. There is no obvious relationship between the magnitude of the deviations and the value of the dependent variable. The statistical measures do not provide indication where in space (or time) the major deviations occur. Graphical techniques are needed to illustrate this test characteristic.

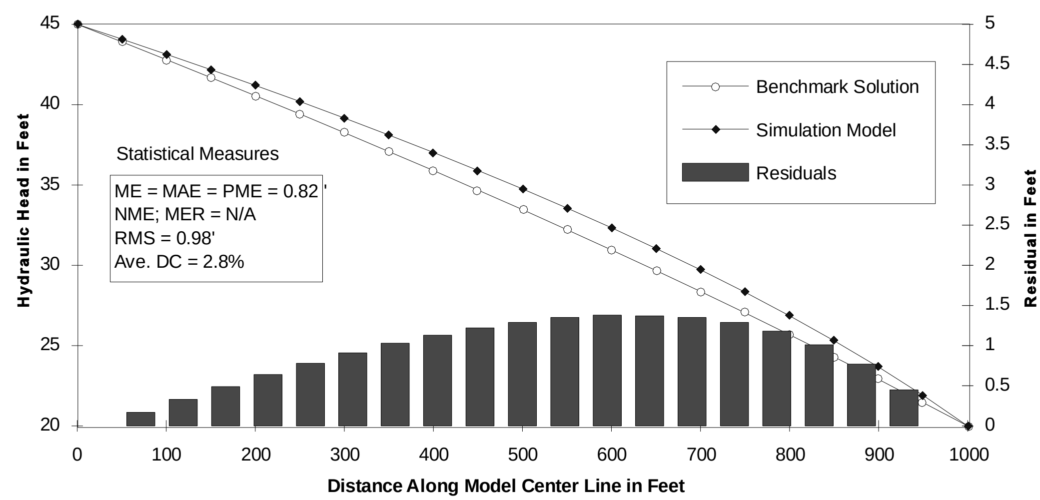


Figure 3-16. The use of statistical measures and graphical techniques to illustrate consistent overprediction of the simulation code.

The second example, shown in Figure 3-17, illustrates a case where the simulation code predominantly underestimates the benchmark solution. The statistical measures effectively summarize this situation. Unlike the example shown in Figure 3-16, which featured no negative residuals due to consistent overestimation, this case is characterized by both positive and negative residuals. Thus, all statistical measures, including NME, PME and MER, may be calculated. Because the simulation code primarily underestimates the benchmark solution, the values of ME and MER are negative and the NME is greater than the PME. The degree of the underestimation can be characterized by the magnitude of the MER. In this case, a MER equal to -12.8 indicates that the simulation model results in 12.8 times the amount of average negative residual than average positive residual. The graphical display clearly shows the distribution of residuals. It is apparent that the residuals are strongly negative in the left hand part of the diagram, indicated by the unshaded columns on the chart. The positive residuals, plotted as shade columns, exist only at distances of greater than 2000 feet along the center line of the simulation model. The plot also shows that the larger deviations occur at higher values of the dependent variable. As is the case with the first example, the statistical measures do not indicate where in space (or time) the major deviations occur. Graphical techniques are needed to illustrate this test characteristic.

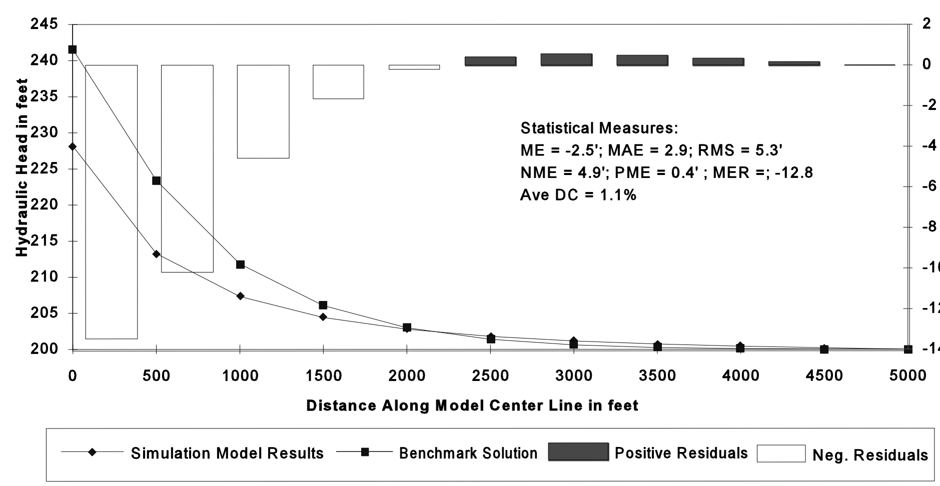


Figure 3-17. The use of statistical measures and graphical techniques to illustrate trends in over- and under-prediction of the simulation code.

The third example, illustrated in Figure 3-18, pertains to a situation where global statistical measures are not sufficient to characterize the overestimation or underestimation tendency. Residuals are almost evenly distributed between negative and positive deviations. The statistical measures indicate that there is a significant error and that the simulation code overestimates the benchmark solution. The PME (3.6 feet) is slightly greater than the NME, resulting in a MER of +1.1 feet. The ME is only 0.5 feet. Note that if the residuals were evenly distributed with an equal number of positive and negative residuals, the ME would be equal to zero and the MER would be equal to one. So, although some of the statistical measures may suggest that the global agreement is reasonably balanced between negative and positive space, there is locally considerable variation from the benchmark solution. The statistical measures do not provide indication where in space (or time) the major deviations occur. Again, graphical techniques are needed to illustrate this test characteristic.

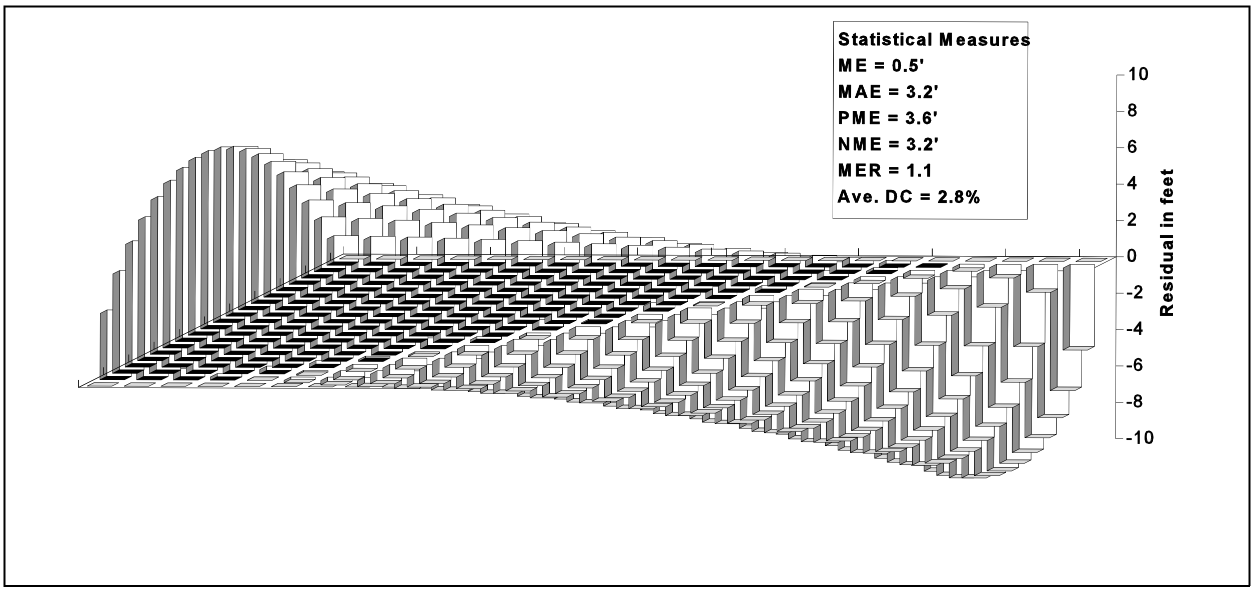


Figure 3-18. The use of statistical measures and graphical techniques to illustrate spatial distribution of over- and underprediction of the simulation code.

3.4.6. Documentation of Test Results

The results of a code testing exercise should be documented, addressing all steps of the code testing and evaluation protocol in a manner that the testing is reproducible and the conclusions well-founded. The report should contain an introductory section, a section describing the performed testing and test results, and a section on recommendations and limitations covering code theory, documentation, functionality, performance and applicability as encountered by the reviewer/tester. A detailed table of contents for the test report is presented in Table 3-15. The test details to be included in the report are listed in Table 3-16. An example of the type of illustrative figures for the test problems is given in Figure 3-19.

Table 3-15. Elements of a test report.

Introduction	
Program name	Test environment (computer, operating system, etc.)
Program title	Reviewed materials/documentation
Tested version	Installation review
Release date	Discussion of general operation (batch, interactive, graphics)
Author/custodian	Terms of availability (legal status, etc.)
Reviewer (name, organization)	Type/level of support
Review date	
Short description	
Computer and software requirements	
Testing	
Analysis of code functions and preparation of functionality description	Presentation and discussion of functionality analysis matrix
Overview and discussion and re-evaluation of testing performed by code authors	Presentation and discussion of performance tables
Overview and detailed description of additional tests performed	Optional discussion of applicability issues both from a theoretical point-of-view, as well as based on applicability testing
Conclusions	
Testing (performance, limitations, cautions)	Installation and general operation
Documentation (completeness and correctness of functionality description, correctness of theory, consistency of mathematical description and coded functionality, correctness and completeness of user's instructions)	Code setup (how easy/difficult it is to run the code)
	Specific hints/tricks learned during testing, not present in documentation

Finally, an executive summary of the code testing effort should be prepared. This summary should function as a stand alone document describing the main code features, providing an overview of the performed tests, discussing major strengths and weaknesses of the code, and listing some key recommendations regarding the code's use. Table 3-17 lists the main components of such an executive summary.

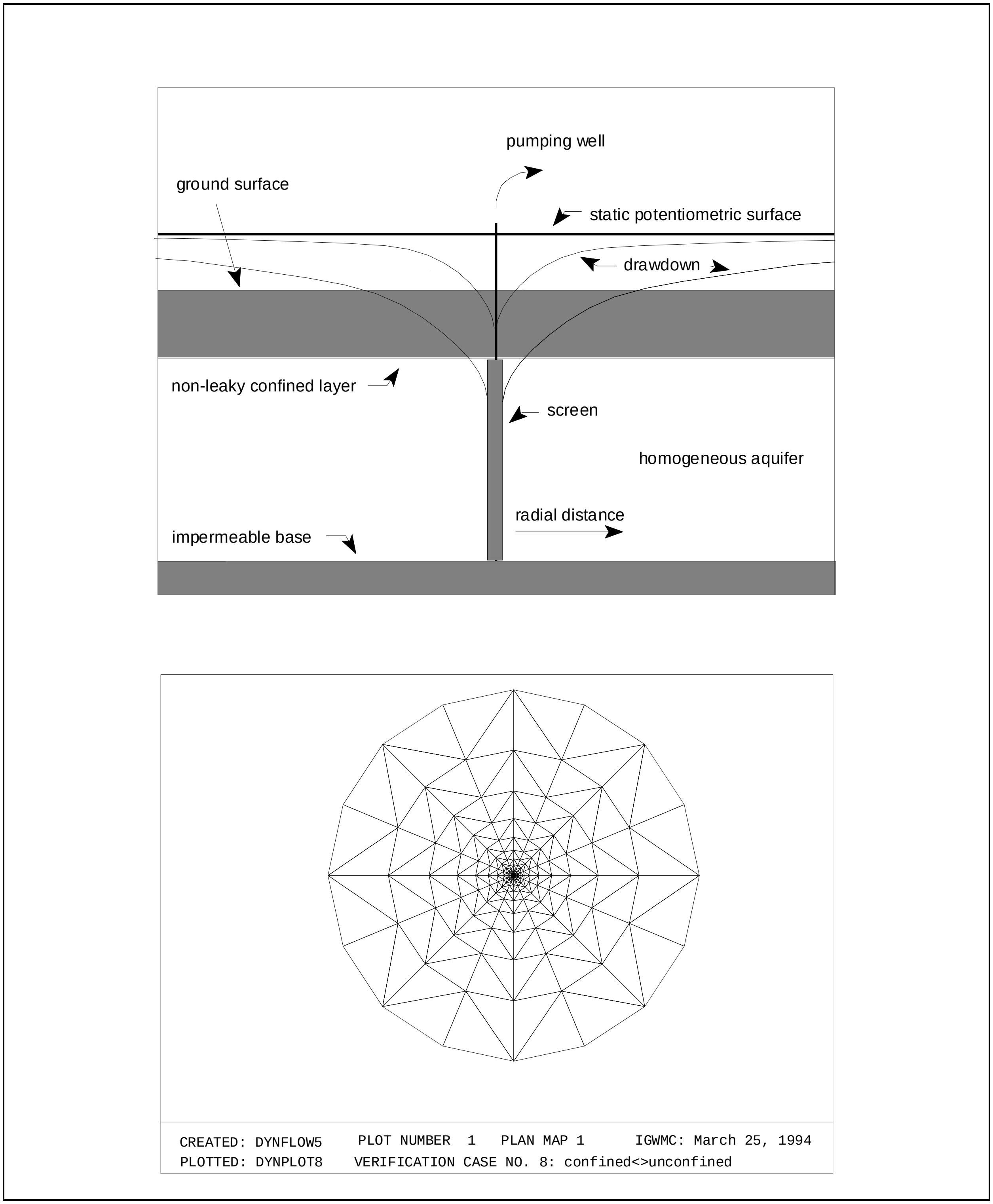


Figure 3-19. Illustration of test problem situation and model grid used in test problem.

Table 3-16. Test details to be discussed in test report

- general problem description (including assumptions, limitations, boundary conditions, parameter distribution, time-stepping, figures depicting problem situation)
- test objectives (features of simulation code, specifically tested by test problem)
- benchmark reference
- if feasible, benchmark solution (*e.g.*, analytical solution)
- reference to benchmark implementation (hand calculation, spreadsheets, dedicated software, etc.)
- test data set
- model setup, discretization, implementation of boundary condition, representation of special problem features (for both tested code and benchmark code; electronic input files)
- results (table of numerical and benchmark results (if available) for the dependent variable at selected locations and times; mass balances; statistical measures and supporting figures; electronic results files)
- sensitivity analysis strategy and results
- discussion of results

Table 3-17. Elements of the executive summary of the test report

- Program name, title, version, release date, authors, custodian
- Reviewer (name, organization)
- Detailed program description (functionality)
- Computer/software requirements
- Terms of availability and support
- Overview of testing performed by authors
- Overview of additional testing performed
- Discussion of specific test results (illustrating strengths and weaknesses)
- Discussion of completeness of testing (functionality matrix)
- Representative performance information
- Main conclusions on test results
- Comments on installation, operation and documentation
- List of main documentation references
- Tables providing overview of performed tests and performance information
- Figures illustrating key results

4. APPLICATION OF THE CODE TESTING PROTOCOL TO THREE-DIMENSIONAL FLOW AND TRANSPORT CODES

4.1. GENERAL COMMENTS

Successful implementation of the code testing protocol depends upon the design of effective test cases, correct implementation of the selected test cases, and unbiased analysis and reporting of test results. The selected test problems should be subject to the following considerations:

- designed to meet specific objectives as well as the needs of particular audiences;
- designed to address multiple issues to increase test efficiency;
- designed in conjunction with other tests to limit redundancy;
- designed to address all three protocol elements, where possible;
- implemented in a standard fashion (problem description, model setup, benchmark description, analysis, reporting); and
- subjected to impartial analysis procedures to eliminate subjectivity, whenever possible.

In this report section, the code testing and evaluation protocol is applied to simulation codes which use rectangularly discretized model domains to simulate steady-state and transient three-dimensional flow and solute transport under saturated hydrogeological conditions. It focuses on the development and execution of the code testing strategy, including the selection of test problems. Except in cases where symmetry exists, the protocol requires simulation of the entire model domain, be it in one, two, or three dimensions, dependent on the dimensionality of the test problem. For the analysis of the results, the model domain might be divided into horizontal or vertical two-dimensional slices; statistical measures and graphical techniques are then applied to each of these slices separately. It is often impractical to analyze the results for the entire model domain. In such cases, representative portions (slices, lines, points) of the model domain should be selected for analysis. To ensure meaningful analysis of results, line-graph analysis and supporting statistical evaluation should be based on a significant number of data points (typically using 25 - 50 data pairs). The selection of slices and lines should follow the recommendations in section 3.4.5.1. Note that choosing a non-representative portion of the model domain can result in erroneously optimistic conclusions regarding the functionality of the tested simulation code.

The three-dimensional finite-difference simulation code, FTWORK, has been used to evaluate implementation of the code testing protocol. FTWORK is a public domain software, originally developed by GeoTrans, Inc, Sterling, Virginia, to model the ground-water flow and mass transport regimes encountered at the U.S. Department of Energy Savannah River Site (Faust *et al.*; 1990). An overview of the capabilities and limitations of FTWORK is presented in Appendix D. The following discussion addresses some relevant issues regarding the implementation of the code testing protocol and the development of a code testing strategy.

4.1.1. <u>Analysis of Functions and Features</u>

The establishment of code functionality is the most basic and essential requirement of code evaluation and, thus, it is the highest priority element of the protocol. Code testing starts with the analysis of the code's functionality, followed by functionality evaluation. The results of this analysis are summarized in the functionality matrices. Functionality testing consists of three steps: 1) identification of functionality issues and test objectives; 2) design test strategy to meet objectives; and 3) perform and analyze test runs. Identification of test objectives (*i.e.*, correctness of the implementation of particular functions in the code), and test issues (*i.e.*, potential problems in specific functions) is crucial to successful evaluation (see Appendix C). When each functionality issue is addressed and each test objective is met through the execution of the test strategy, functionality testing is complete.

Where possible, functionality tests should be based on the availability of a benchmark solution. There are a variety of analytical and numerical solutions which may be used as benchmark. Many analytical solutions can be found in text books and compilations, such as Bear (1979), van Genuchten and Alves (1982), Hunt (1983), Walton (1984); Luckner and Schestakow (1991), Beljin (1992), Wexler (1992), and Beljin and Murdoch (1994). In selecting analytical solutions as benchmark, care should be taken with respect to their correct computer implementation. Many analytical solutions are complex in nature and include functions which require numerical approximation.

4.1.2. Performance Issues

Performance evaluation establishes the performance characteristics of a simulation code by evaluating run-time performance characteristics. The performance characteristics can be used to differentiate between codes of identical functionality, and to estimate resource requirements in project planning. Performance evaluation should be an integral part of the code testing strategy, using the same tests as in the functionality analysis. To ensure compatibility and comparability among different simulation codes, it is important that performance evaluation of simulation codes is conducted using a standard computer configuration. Performance issues related to human variability (*e.g.*,user skills and knowledge) are not part of performance evaluation. Results should be analyzed and presented using standard measures and summary structures (*i.e.*, performance evaluation checklists).

4.1.3. Applicability Issues

Applicability assessment is most significant when identified applicability issues cannot be easily assessed from a code's functionality description. Thus, standard data sets are developed representing typical application environments. These data sets are specifically designed to demonstrate the capability of simulation codes to represent specific real-world issues of concern, as well as to uncover problems encountered in model setup. Applicability assessment is not aimed as much at code intercomparison as demonstrating the code's ability to simulate practical, real-world problems. The results, where possible, should be compared with established numerical benchmarks (*e.g.*, obtained with other simulation codes), using statistical and graphical residual analysis.

4.2. EXAMPLE TESTING AND EVALUATION USING THE CODE "FTWORK"

To demonstrate the use of the code testing protocol, the following steps have been taken, featuring the FTWORK code:

1) identifying and examining code functionality
2) determining type and objectives of tests performed and documented by the code developers;
3) evaluating the suitability of performed tests for use in protocol demonstration;
4) compiling protocol summary structures (*i.e.*, checklists, matrices) using performed tests;

5) designing and conducting new tests, to address some gaps in the test strategy used by the code developers; and
6) summarizing the combined results of tests performed by code developers and tests performed as part of the protocol demonstration.

Most of the tests originally performed by the code developers were adapted, augmented, and re-analyzed to ensure consistency with the protocol. The additional tests designed during this study demonstrate how to eliminate gaps in code-evaluation.

The code-evaluation tests for FTWORK were performed on 50 and 60 MHZ Intel 80486 and 90 MHZ Pentium™ based personal computers using Microsoft MS-DOS™ operating system (version 6.20) and on IBM RISC™ 6000 workstations using the Unix operating system (AIX version 2.2). The protocol demonstration was performed using version 2.8B of the FTWORK source code, compiled and linked by IGWMC using the Lahey F77L/EM 32™ FORTRAN compiler (version 5.0; Lahey, 1992). Evaluation measures were calculated using the Microsoft spreadsheet program Excel™ (versions 5.0; Microsoft, 1994), and plotted using Excel and Golden Software's Grapher™ for Windows (version 1.0; Golden Software, 1992).

4.2.1. Code Description

To simplify and classify the functionality description process, the code functions are organized into four functionality categories, including code options, methods and capabilities. These four categories, and their principal components are:

1) general code characteristics, which include code discretization options, spatial orientation options, restart options, and code output options;
2) flow system characteristics, which include hydrogeologic zoning options, hydrogeologic media options, flow characteristics options, boundary condition options, source/sink functions, and mathematical solution methods;
3) solute fate and transport characteristics, which include water quality constituents, transport and fate processes, boundary conditions, and mathematical solution methods; and

4) parameter estimation characteristics (where appropriate), which include: input options, output options, and solution methods.

The functionality of FTWORK has been determined using the generic functionality description form of Appendix B; the results are presented in Appendix D. A short description is of FTWORK is given in the following paragraphs (Faust *et al.*, 1993).

Purpose and General Features

FTWORK is a block-centered finite difference code designed to simulate transient and steady-state three-dimensional saturated ground-water flow and transient transport of a single dissolved component under confined and unconfined conditions. It supports both areal and cross-sectional two-dimensional simulations. Its primary use is to simulate the migration of contaminants at low concentrations to assess impacts of contamination and to aid in developing a remediation strategy. The code may be used for characterizing large, complex, multi-layered, fully-saturated, porous hydrogeologic systems. The code can be used in a quasi-three-dimensional mode.

The flow equation is posed in terms of hydraulic head, the transport equation in terms of concentration. It is assumed that fluid density is independent of concentration, and density and porosity changes due to changes in hydraulic head have negligible effect on the transport of solutes. FTWORK includes the calculation of a comprehensive, model-wide mass balance for both flow and mass transport. The code supports variable grid block lengths in X-, Y-, and Z-direction and deformed coordinate approximation for variable thickness layers.

Boundary conditions include prescribed head, prescribed concentration, prescribed flux of water (*e.g.*, recharge) or solute mass, and head-dependent flux (*e.g.*, for leakage to or from streams, flow to drains). It also handles time-varying single- and multi-aquifer wells, and chemical sources and sinks. The default boundary condition is no-flow and zero solute flux. The code achieves the default condition by setting the transmissivity and dispersivity to zero along such boundaries. A prescribed flux boundary is specified by using source terms or recharge rates. Inflow is simulated by specifying the concentration of an injection well fluid or recharge to determine the solute influx. For outflow, the solute mass flux is determined using the product of the grid block concentration and the ground-water pumping rate. If a well is simulated in more than one layer, flow is apportioned to the open layers on the basis of layer transmissivity. The code assigns recharge to the uppermost active grid block and is apportioned based on grid block dimensions. A prescribed head boundary is specified at the center of the grid block adjacent to the boundary, along with concentrations so that advective solute mass fluxes may be computed. A third boundary condition, head-dependent flux, can be used to simulate three different cases: a leaky boundary, a leaky boundary with potential for dewatering below the base of the semi-pervious boundary, and a drain boundary. The standard leaky boundary can apply leakage through an adjacent aquitard without storage or to leakage through a stream bed. A provision for dewatering below a stream bed or leaky aquitard is the function of the modified leaky boundary. For a drain boundary, flow is approximated as head-dependent flux that occurs only if the head in the grid block containing a drain is higher than the specified head in the drain.

Spatially variable flow parameters include hydraulic conductivity, specific storage or porosity, recharge, and evapotranspiration. The code handles anisotropy for flow assuming that the hydraulic conductivity tensor is aligned with the Cartesian coordinate axes. It supports the conversion from confined to unconfined conditions, and dewatering of a grid block. For unconfined conditions the transmissivity is a function of the saturated thickness in adjacent blocks.

Transport and fate processes supported by the code include advection, hydrodynamic dispersion, linear and non-linear (Freundlich) equilibrium sorption by using a nonlinear retardation coefficient, and first-order (chemical, biological, and radioactive) decay. Cross product terms for dispersion can be included in the transport calculations. Longitudinal and transverse dispersivity, retardation factors, and decay factors are considered spatially variable transport parameters.

The model includes a parameter estimation option (semi-automatic history matching) of the steady-state flow equation, using a Gauss-Newton, non-linear least-squares technique for global minimization of the differences in observed and computed heads, together with a Marquardt correction. This option may be used to estimate horizontal and vertical hydraulic conductivity, and recharge.

FTWORK has an option to use either central or upstream weighting of the advection term and central or backwards weighting of the time derivative. For general three-dimensional problems, an iterative method, the Slice Successive Over-Relaxation (SSOR) method, is used to solve the non-coupled flow and transport equations. The resulting matrix equations are solved using the Gauss-Doolitle method for banded coefficient matrices. FTWORK includes two other solvers to be used for problems of reduced complexity.

FTWORK creates a cell-by-cell flux file which is compatible with the USGS particle tracking code, MODPATH. Using MODPATH, however, requires modification of the input data file. An MS-Windows™ based preprocessor, PRE-FTW, has been prepared by IGWMC. In this preprocessor, array entry and editing is performed using a spreadsheet format. FTWORK provides restart capabilities which can be used to continue computations from previously completed simulations or from previous time steps. FTWORK's output options include:

main output file: an ASCII text file containing a summary of the input data (control parameters, grid block data, flow and/or transport parameters, initial conditions, time parameters, source/sink data, recharge data, and evapotranspiration data), convergence error, array data (head and/or concentration, Darcy velocity, and saturation index), and, if parameter estimation is performed, summary statistics and parameter multipliers, and residuals.

plot file: MODFLOW-type binary or ASCII files of head- and concentration distribution for graphic postprocessing.

sensitivity coefficient file: results of sensitivity calculations for each grid block for each calibrated parameter in the parameter estimation procedure.

observation block file: heads and/or concentration as function of time for selected nodes.

residuals file: observed heads, computed heads, computed residuals.

restart file: head and/or concentration at end of simulation to be used as initial conditions for a subsequent run.

cell-by-cell flux files: various types of cell-by-cell fluxes from steady-state simulations using MODPATH compatible method and formats.

The users manual contains additional specific information on model theory, code structure, user instructions, code listing, verification by analytical solutions, as well as sample input and output for example problems and tests.

Limitations of FTWORK include: 1) water density is independent of concentration; 2) flow is independent of density and viscosity; 3) for water table conditions, free surface must not be too steep; 4) treatment of dispersive processes is based on uniform (non-scale-dependent) longitudinal and transverse dispersivity concepts; and 5) FTWORK does not support resaturating a grid block once it has gone dry, limiting its use for thin aquifers subject to significant head changes).

4.2.2. Test Issues

Based on the analysis of code functions, a list of major test issues has been compiled (see Table 4-1). This list includes functionality, performance, and applicability issues. Major issues are those that might have incorrectly implemented or cause problems in their use. Selection of issues is based on theoretical and empirical considerations. Separate test issues have been formulated for FTWORK's parameter optimization option related to the sensitivity of the generated distributions of hydraulic conductivity and recharge for various stress conditions and numerical parameter settings.

4.2.3. Tests Discussed in Documentation

The identified test issues should be evaluated through a well-chosen set of benchmark and intercomparison tests. To evaluate the comprehensiveness of the testing performed by the FTWORK authors, published tests have been analyzed with respect to the issues stated in Table 4-1. The FTWORK documentation (Faust et al., 1993) presents eleven code verification problems (Test Level

1B; see section 3.4.4.1). In addition, the documentation discusses eight code intercomparison cases (Test Level 2B), and two intracomparison cases (Test Level 2A). Table 4-2 provides an overview of the performed tests, benchmark solutions, and test type and level. Finally, the documentation presents two examples for which neither a benchmark exists nor intercomparison has been used.

Table 4-1. Major test issues for three-dimensional finite-difference saturated ground-water flow and solute transport codes.

General Features
- mass balances (regular versus irregular grid)
- variable grid (consistency in parameter and stress allocation)

Hydrogeologic Zoning, Parameterization, and Flow Characteristics
- aquifer pinchout, aquitard pinchout
- variable thickness layers
- storativity conversion in space and time (confined-unconfined)
- anisotropy
- unconfined conditions
- dewatering
- sharp contrast in hydraulic conductivity

Boundary Conditions for Flow
- default no-flow assumption
- areal recharge in top active cells
- induced infiltration from streams (leaky boundary) with potential for dewatering below the base of the semi-pervious boundary
- drain boundary
- prescribed fluid flux
- irregular geometry and internal no-flow regions

Transport and fate Processes
- hydrodynamic dispersion (longitudinal and transverse)
- advection-dominated transport
- retardation (linear and Freundlich)
- decay (zero and first-order)
- spatial variability of dispersivity
- effect of presence or absence cross-term for dispersivity

Boundary Conditions for Solute Transport
- default zero solute-flux assumption
- prescribed solute flux
- prescribed concentration on stream boundaries
- irregular geometry and internal zero-transport zones
- concentration-dependent solute flux into streams

Sources and Sinks
- effects of time-varying discharging and recharging wells on flow
- multi-aquifer screened wells
- solute injection well with prescribed concentration (constant and time-varying flow rate)
- solute extraction well with ambient concentration

Reviewing the suite of published tests (see Appendix E), it appears that some of the test issues stated in Table 4-1 have not been addressed. The objectives of the individual tests are not always clearly stated, and have to be deducted from the test set up and test conclusions (if present). The intercomparison and analysis procedures would have benefited from a more consistent use of graphical and statistical evaluation techniques. The tests performed by the FTWORK authors have been compiled in a functionality matrix (see Figure 4-1). Cross marks identify FTWORK functions addressed by the documented tests. Important functions not addressed in the testing include aquifer and aquitard pinchout, storativity conversion, anisotropic hydraulic conductivity, partially penetrating wells and solute sources, vertical transverse dispersion, and non-point, diffusive sources (e.g., from precipitation). Another potential problem, not addressed by the reported tests, is solute transport in a system with strongly curving flow lines, such as around an injection-extraction well pair.

Table 4-2: List of code tests and example applications presented in FTWORK documentation (Faust *et al.*, 1993)

IGWMC Reference Number	Section and Page in FTWORK Manual	Description	Type of Benchmark	Type of Test (see section 3.4.1.2)
GROUND-WATER FLOW PROBLEMS				
FTW-TST-1.1	4.1.1/59	steady-state one-dimensional flow to parallel drains in unconfined aquifer with vertical recharge	analytical solution	functionality level 1B
FTW-TST-1.2	4.1.2/61	transient one-dimensional flow to a fully-penetrating drain in a semi-infinite confined aquifer due to a step-change in head	analytical solution	functionality level 1B
FTW-TST-1.3	4.1.3/70	transient radial flow to a fully-penetrating well near a fully-penetrating straight-line recharge boundary in a confined aquifer	analytical solution, superposition	functionality level 1B
FTW-TST-1.4	4.1.4/70	transient radial flow to a fully-penetrating well in a non-leaky confined aquifer	analytical solution	functionality level 1B
FTW-TST-1.5	4.1.4/70	transient radial flow to a fully-penetrating well in a leaky confined aquifer	analytical solution	functionality level 1B
FTW-TST-1.6	5.1/133	transient response of a regional two-aquifer flow system to increased pumping from additional wells in lower aquifer near center of model domain	intercomparison	functionality, applicability level 2B
FTW-TST-1.7.1	5.2/136	steady-state flow in a three-aquifer system with areal recharge, and outflow into buried drains, through wells, and at specified head boundary cells; using drain option	intercomparison	functionality, applicability level 2B

IGWMC Reference Number	Section and Page in FTWORK Manual	Description	Type of Benchmark	Type of Test (see section 3.4.1.2)
FTW-TST-1.7.2	5.2/136	transient flow in a three-aquifer system with areal recharge, and outflow into buried drains, through wells, and at specified head boundary cells; using drain option	intercomparison	functionality, applicability level 2B
FTW-TST-1.7.3	5.2/136	transient flow in a three-aquifer system with areal recharge, and outflow into buried drains, through wells, and at specified head boundary cells; using stream leakage option	intercomparison	functionality, applicability level 2B
FTW-TST-1.8	5.5.2/172	two-dimensional transient flow in a homogeneous, isotropic unconfined aquifer with depth-limited evapotranspiration and well-pumping	intercomparison	functionality, performance, applicability level 2B
FTW-TST-1.9	5.5.1/164	two-dimensional steady-state flow in two-aquifer system; the shallow confined aquifer is subject to recharge, depth-limited evapo-transpiration, pumping, and upward leakage from the underlying confined aquifer.	intercomparison	functionality, performance, applicability level 2B
SOLUTE TRANSPORT PROBLEMS				
FTW-TST-2.1	4.2.1/81	transient one-dimensional advective-dispersive transport from a first-type inlet boundary in an infinite porous medium with a uniform flow field (steady-state one-dimensional flow)	analytical solution	functionality level 1B
FTW-TST-2.2.1	4.2.2/87	transient two-dimensional advective-dispersive transport of a conservative tracer from a fully-penetrating point source with constant release rate in a uniform flow field in a homogeneous confined aquifer of constant thickness using a parallel grid; cross-products included	analytical solution	functionality level 1B
FTW-TST-2.2.2	4.2.2/87	transient two-dimensional advective-dispersive transport of a conservative tracer from a fully-penetrating point source with constant release rate in a uniform flow field in a homogeneous confined aquifer of constant thickness using a skewed grid; cross-products included	analytical solution	functionality level 1B
FTW-TST-2.2.3	4.2.2/87	transient two-dimensional advective-dispersive transport of a conservative tracer from a fully-penetrating point source with constant release rate in a uniform flow field in a homogeneous confined aquifer of constant thickness using a skewed grid; lumped cross-products	analytical solution	functionality level 1B

IGWMC Reference Number	Section and Page in FTWORK Manual	Description	Type of Benchmark	Type of Test (see section 3.4.1.2)
FTW-TST-2.3	4.2.3/105	transient two-dimensional advective-dispersive transport of a nonconservative tracer from a fully-penetrating point source with constant release rate in a uniform flow field in a homogeneous confined aquifer of constant thickness using a parallel grid; the tracer is subjected to retardation and first-order (radio-active) decay	analytical solution	functionality level 1B
FTW-TST-2.4.1	4.2.4/105	transient one-dimensional advective-dispersive transport of a non-conservative tracer in a uniform flow field with non-linear adsorption as defined by Freundlich isotherms	intercomparison	functionality level 2B
FTW-TST-2.4.2	4.2.4/105	transient one-dimensional advective transport of a non-conservative tracer in a uniform flow field with non-linear adsorption as defined by Freundlich isotherms and molecular diffusion	intercomparison	functionality level 2B
FTW-TST-2.5	4.2.5/114	transient two-dimensional advective-dispersive transport of a non-conservative tracer from a constant flux-type source (third type or Cauchy condition at the inlet boundary); uniform flow field in a homogeneous porous medium; vertical plane source from top to bottom of aquifer, perpendicular to the flow direction.	analytical solution	functionality level 1B
FTW-TST-2.6	5.4/156	simulation of three-dimensional steady-state flow and transient transport in a three-aquifer system with variable thickness; the aquifers are separated by aquitards; model includes streams, seeplines, seepage basins, ground-water divides, and near-impermeable confining layers at part of the boundary.	no benchmark	applicability
FTW-TST-2.7.1	5.5.3/174	two-dimensional transient flow and transport in a homogeneous, isotropic unconfined aquifer with depth-limited evapotranspiration or drain-discharge, and well-pumping; an injection well creates solute mass in the model	intra-comparison	functionality applicability level 2A
FTW-TST-2.7.2	5.5.3/174	drain transport problem to test the evapotranspiration transport function; problem set up identical to 2.7.2 with evapotranspiration nodes replaced by drain nodes	intra-comparison	functionality applicability Level 2A
INVERSE FLOW PROBLEMS				

IGWMC Reference Number	Section and Page in FTWORK Manual	Description	Type of Benchmark	Type of Test (see section 3.4.1.2)
FTW-TST-3.1	5.3/148	simulation of steady-state three-dimensional flow in a four-aquifer/three-aquitard system subject to pumping and uniform areal recharge; hydraulic conductivity is homogeneous within each layer but transmissivity varies with layer thickness	manual calibration	functionality (qualitative) applicability level 3C

FTWORK Code Function	1.1	1.2	1.3	1.4	1.5	1.6	1.7.1	1.7.2	1.7.3	1.8	1.9	2.1	2.2.1	2.2.2	2.2.3	2.3	2.4.1	2.4.2	2.5	2.6	2.7.1	2.7.2	3.1
Hydrogeologic zoning																							
confined aquifer		x	x									x	x	x	x	x	x	x	x				
semi-confined (leaky- confined)				x	x	x	x	x	x	x	x									x			x
unconfined (phreatic) aquifer	x					x	x	x	x	x	x									x	x	x	x
1D/single aquifer	x	x															x	x					
2D/single aquifer-aquitard system; areal view			x	x	x							x	x	x	x	x			x		x	x	
2D/single aquifer-aquitard system: profile view																							
quasi-3D/ multiple aquifer/aquitard systems							x	x	x	x	x									x			x
fully 3D multiple aquifer/aquitard systems						x																	
variable thickness aquifers						x														x			x
variable thickness aquitards						x																	
aquifer pinchout																							
aquitard pinchout																							
storativity conversion																							
Hydrogeologic Media																							
anisotropic hydraulic conductivity																							
horizontal anisotropy																							
vertical anisotropy										x													
nonuniform, heterogeneous hydraulic properties																							
horizontal heterogeneity							x	x	x	x	x												
vertical heterogeneity						x	x	x	x	x	x												x
Flow Characteristics																							
steady-state flow	x						x				x	x	x	x	x	x	x	x	x	x			x
transient (non-steady-state) flow		x	x	x	x	x		x	x	x											x	x	
dewatering (desaturation of cells)										x													
Boundary Conditions																							
regular bounded domain	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x	x		x	x	
irregular bounded domain											x									x			x
fixed/specified head	x	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
zero flow (impermeable barrier)	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
fixed cross-boundary flux											x									x			x
areal recharge	x						x	x	x	x	x									x	x	x	x
head-limited drain cells							x	x												x		x	
stream cells with head-dependent flux									x											x			
stream cells with g.w.level beneath bottom of st									x											x			
depth-limited evapotranspiration										x	x										x		
Flow Sources / Sinks																							
point sources/sinks (recharge/pumping wells)																							
constant flow rate			x	x	x	x	x	x	x	x	x									x	x	x	x
variable flow rate																							
head-specified																							
partially-penetrating																							
multi-layer well						x																	

Figure 4-1a. Functionality matrix of testing performed by FTWORK developers

FTWORK Code Tests by Developers

FTWORK Code Function	1.1	1.2	1.3	1.4	1.5	1.6	1.7.1	1.7.2	1.7.3	1.8	1.9	2.1	2.2.1	2.2.2	2.2.3	2.3	2.4.1	2.4.2	2.5	2.6	2.7.1	2.7.2	3.1
Fate and Transport Characteristics																							
steady-state advection																							
uniform												x	x	x	x	x	x	x	x				
non-uniform																				x			
transient advection																					x	x	
dispersion																							
longitudinal												x	x	x	x	x	x		x	x	x	x	
hor. transverse													x	x	x	x			x	x	x	x	
vert. transverse																							
homogeneous (constant in space)												x	x	x	x	x	x		x		x	x	
heterogeneous																				x			
grid parallel to flow													x			x	x	x	x				
grid skewed with respect to flow														x	x					x	x	x	
internal dispersivity cross-terms														x						x	x	x	
chemical fate																							
zero-order production																							
first-order decay																x	x	x		x			
molecular diffusion																		x					
solid-liquid phase transfers (sorption)																							
linear equilibrium isotherm																x				x			
Freundlich equilibrium isotherm																	x	x					
Transport Boundary Conditions																							
prescribed concentration												x					x	x		x			
zero solute flux												x	x	x	x	x	x	x		x	x	x	
prescribed solute flux																			x	x			
Transport sources / sinks																							
source with constant concentration and flow rate													x	x	x	x			x	x	x	x	
source with time-varying concentration and flow rate																							
fully-penetrating sources													x	x	x	x			x	x	x	x	
partially-penetrating sources																							
sink with concentration dependent solute flux																				x	x	x	
point sources (injection wells)													x	x	x	x				x	x	x	
point sinks (pumping wells, springs)																				x	x	x	
line sources (infiltration ditches or canals)																			x	x			
line sinks (drains, streams)																				x		x	
horizontal areal or patch sources (landfills, feedlots)																				x			
vertical patch sources																			x				
non-point, diffuse sources (agricultural spraying)																							
plant uptake (evapotranspiration)																					x		
Inverse modeling (flow)																							
parameters to be estimated																							
hydraulic conductivity																							x
areal recharge																							x
Numerical Solution Methods																							
restart option								x	x														
variable grid spacing											x												
horizontally varying				x	x	x					x									x			
vertically varying						x	x	x	x	x	x									x			x

Figure 4-1b. Functionality matrix of testing performed by FTWORK developers

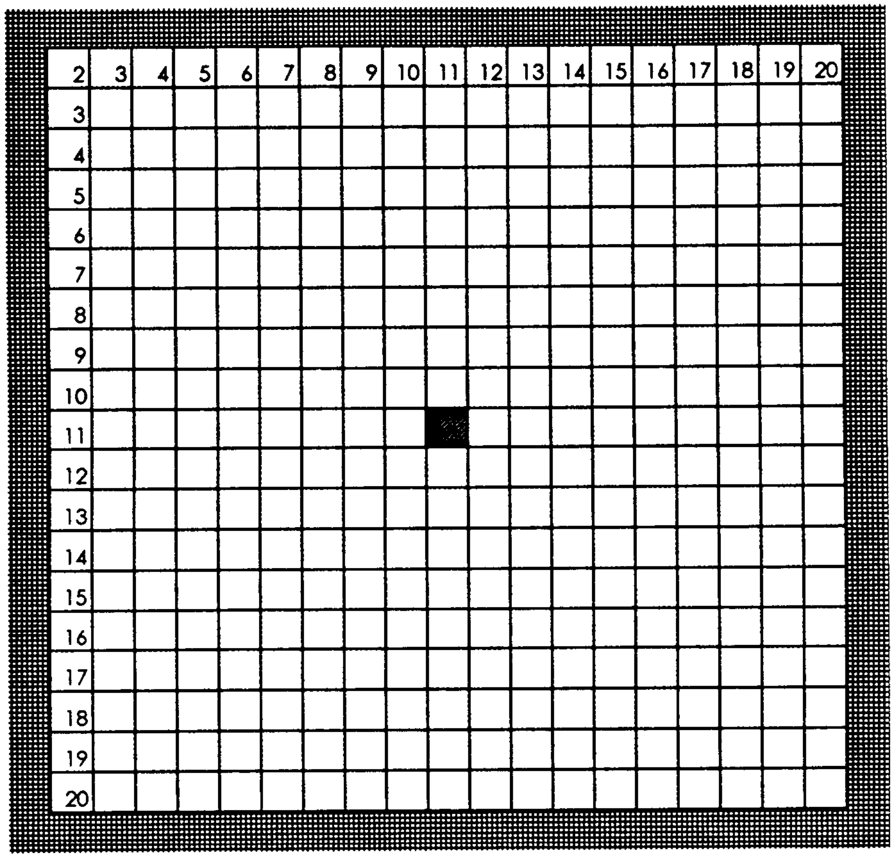
As is indicated in the last column of Table 4-2, the verification tests and example problems presented in the FTWORK documentation cover both functionality and applicability aspects of the testing protocol. Most tests include some evaluation of accuracy. A few of the tests actually address other performance issues. However, most tests and example problems do not provide the necessary information for in-depth performance evaluation. It should be noted that additional intercomparison testing of FTWORK was performed by Sims *et al.* (1989), comparing FTWORK results with those obtained using the numerical simulation models, SWIFT II, MODFLOW, SWICHA, and CFEST.

4.2.4. Additional Tests Performed by IGWMC

To evaluate capabilities and characteristics of the FTWORK code, not addressed in the documentation, additional tests have been designed and executed. This exercise is also aimed at assessing the procedures for the development of such tests. To evaluate functionality testing, three problems were designed focused on areal recharge, radioactive decay, and anisotropy of flow parameters, respectively. The latter problem has been specifically formulated to study effects of grid orientation on anisotropy. Various performance issues have been studied by executing the test problems provided by the FTWORK authors (Faust *et al.*, 1993), and evaluating the results using specific performance measures.

Areal Recharge

To evaluate the functionality of FTWORK with respect to areal recharge, various test issues have been identified (see Appendix C, Table C-3). The functionality issues are translated in test objectives, which in turn determine the type of tests required or available. To illustrate this procedure, a few areal recharge issues are selected for detailed discussion using two simple problem configurations representing a rectangular shaped aquifer with homogeneous aquifer parameters (*i.e.*, saturated thickness, hydraulic conductivity, and storativity), bound at all sides by constant-head boundaries. The first (single layer) model consists of 21 by 21 square cells of 500 by 500 ft each (see Figure 4-2). Recharge is introduced at the centermost cell of the model domain, creating a symmetrical situation with respect to the main axis. The edge of the recharge area is 250 ft from the model center.



Summary of Hydraulic Parameters:

hydraulic conductivity K = 10 feet/day

storage coefficient (unconfined) S_y = 0.2

aquifer thickness b = 200 feet

total recharge Q_t = 250,000 ft^3/day

time for comparison:

unconfined case: t = 121 days

Model Setup:

domain is 10,500 x 10,500 ft

single layer model of 21 x 21 cells of 500 x 500 ft each

Specified Head Boundary Cell

Recharge Area

Figure 4-2. Problem definition and model setup for the constant grid areal recharge functionality test

In one of the tests, FTWORK is compared with an analytical solution for mounding due to recharge in a rectangular area based on Hantush (1967), as modified by Warner *et al.* (1989). The solution has been programmed in MathCad® (Mathsoft, 1994; see files MND-EPA1.MCD and MND-EPA2.MCD, respectively, in appendix F). The solution assumes that the mounding is small compared with the saturated thickness of the unconfined aquifer. The results are summarized in Figure 4-3 and Figure 4-4. Figure 4-3 shows the results along a line extending from the model center to the boundary along the principal grid axis (note that residuals have been shifted to the center of the plot for display purposes). Often, this is the only analysis discussed in a code's documentation, biased towards small deviations from the benchmark. Both the graphical representation and the statistical measures suggest that areal recharge is accurately simulated by the code. However, this conclusion may not be representative for the entire model domain. To further explore this issue, a radial slice representing one eighth of the symmetrical model domain is analyzed (see Figure 4-4). All computed heads in this slice are used for comparison, including those present on a line under 45 degrees with the coordinate axes. The degree and nature of the deviations between the code and the benchmark differ significantly from those found along the coordinate axis.

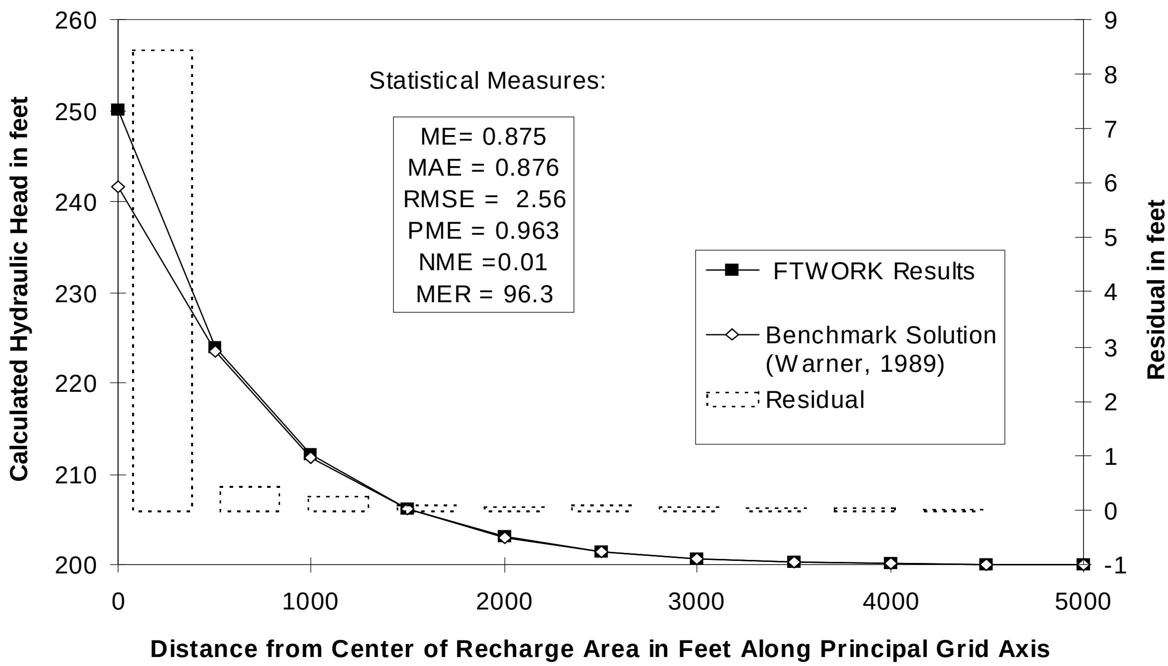


Figure 4-3. Combination plot of heads and residuals versus distance from center of recharge area, measured along one of the grid axis.

The statistical measures presented in Figure 4-4 are based upon 56 points; this is more representative than the statistical measures presented in Figure 4-3, which were calculated using only 11 intercomparison points. Although the statistical measures are generally smaller for the radial slice analysis than for the linear slice analysis, this may be deceiving. The great number of small residuals calculated at large distances from the recharge area causes a downward weighting effect to the statistical measures. This suggests that statistical measures, when used alone, can be misleading and should always be used in conjunction with graphical measures.

To further explore test design influence on test results, the same problem was executed using a variable grid with higher density of cells in the center of the model domain than near the edges (see Figure 4-5). This model setup provides greater resolution in the area with steeper gradients as well as greater flexibility in the distribution of areal recharge. The grid consists of 49 by 49 cells, covering a model area of 29,000 ft. by 29,000 ft. Areal recharge is introduced through a 500 foot by 500 foot area in the center of the domain, discretized in twenty-five 100 foot by 100 foot cells.

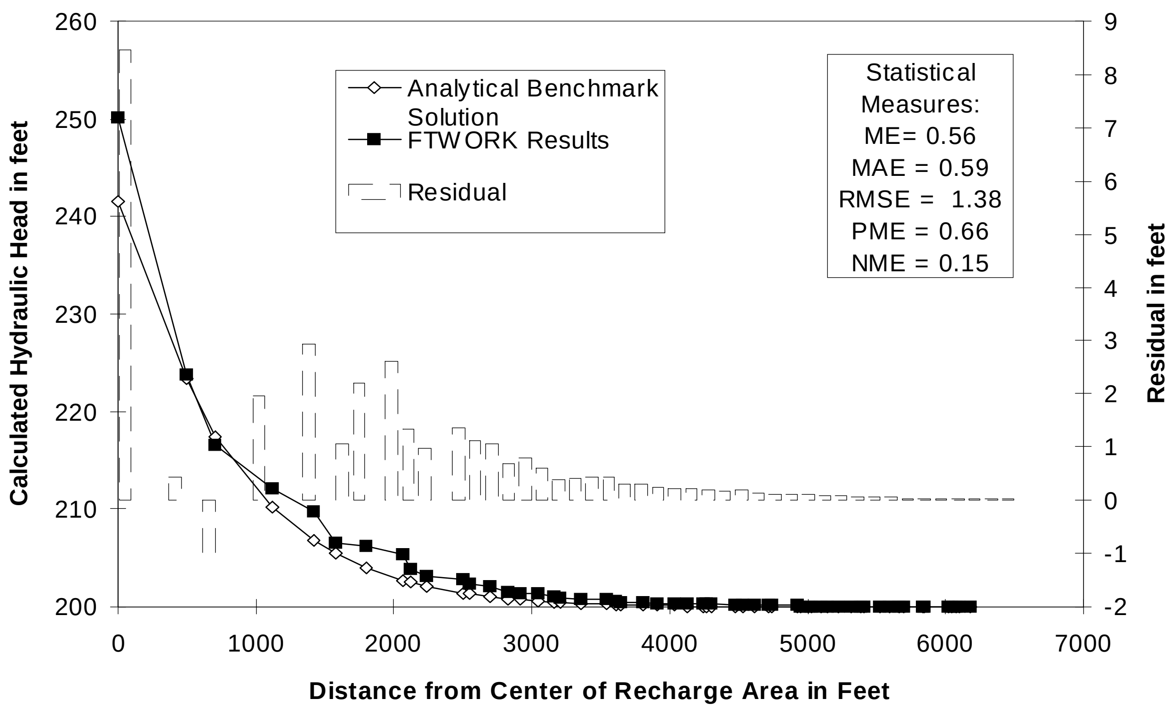
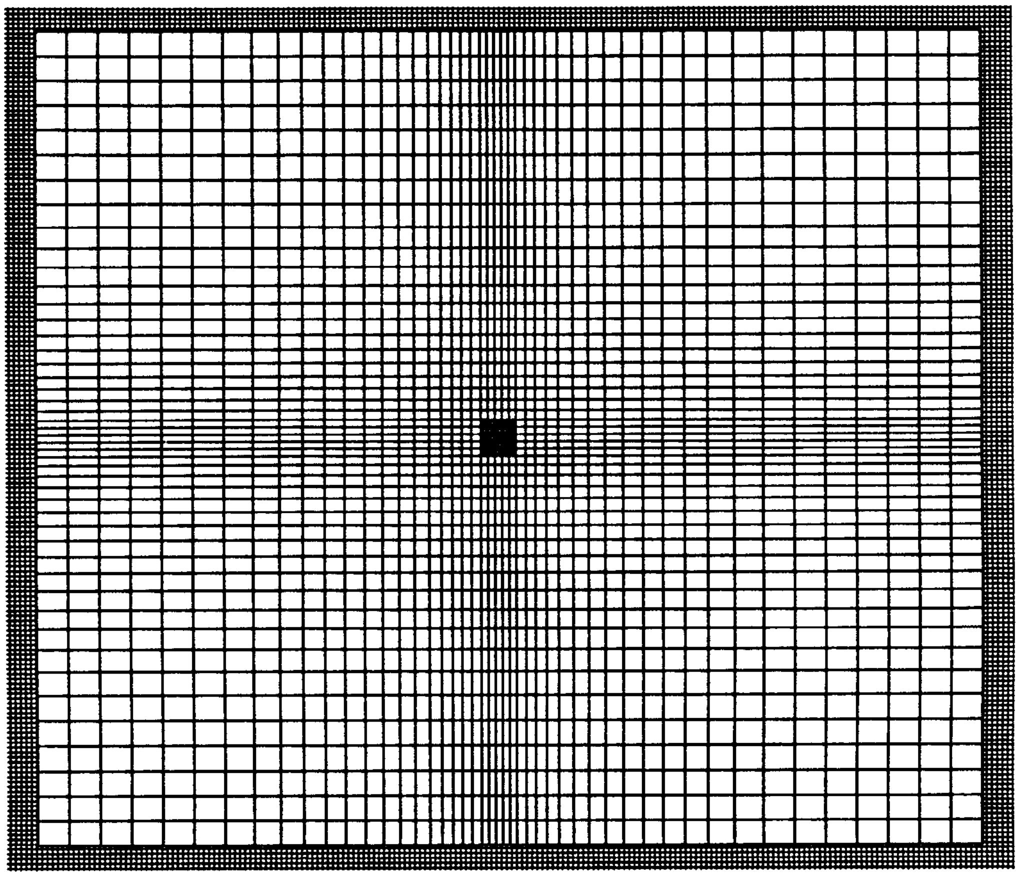


Figure 4-4. Combination plot of heads and residuals versus distance from center of recharge area for all cells in a one-eighth section of the model domain.



Summary of Hydraulic Parameters:
hydraulic conductivity K = 10 feet/day
storage coefficient (confined) S = 1E-3
aquifer thickness b = 200 feet
total recharge Q_t = 250,000 ft^3/day
time for comparison:
confined case: t = 21 days

Model Setup:
domain is 29,000 x 29,000 ft
single layer model of 49 x 49 cells

Specified Head Boundary Cell

Recharge Area

Figure 4-5. Problem definition and model setup for the variable-spaced grid areal recharge functionality test

The numerical results were compared to two different benchmark solutions along one of the principal grid centerlines (Glover, 1960; Warner *et al.*, 1989; see Appendix F, file MND-EPA3.MCD). Figure 4-6 shows the results for the comparison with the Warner *et al.* (1989) solution. The differences in the recharge area and at the domain edges are caused by approximations made to represent the problem in the numerical code.

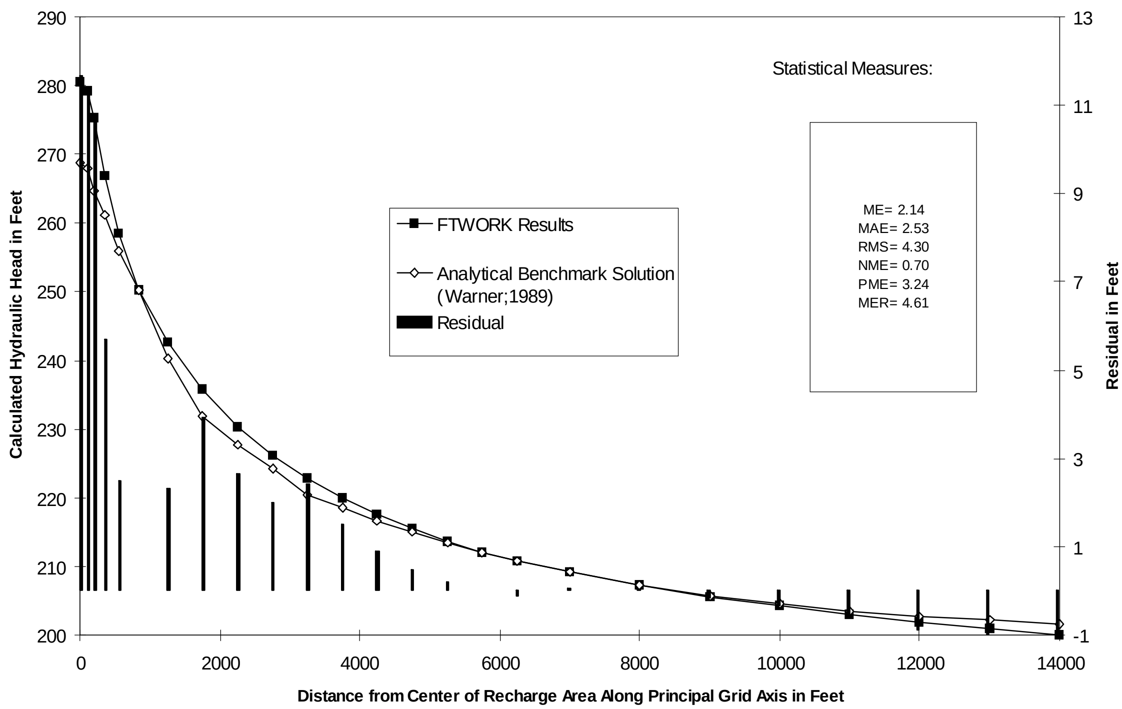


Figure 4-6. Combination plot of heads and residuals versus distance from center of recharge area for variably spaced points along centerline of grid using the Warner *et al.* (1989) solution.

Figure 4-7 shows that the magnitude of the deviations are not always due to inaccuracies inherent to the use of a numerical model. The same numerical results presented in Figure 4-6 are plotted against the Glover (1960) version of the benchmark. Using the original Glover solution improves significantly the agreement between the simulation code and the benchmark, illustrated by smaller statistical measures. The statistical measures indicate that the FTWORK results approximate the Glover (1960) benchmark solution much more precisely than the Warner *et al.* (1989) benchmark solution. The RMSE of 1.67 feet is 63% smaller than the RMSE calculated by the Warner *et al.* (1989) benchmark solution. In addition, the MAE for the Glover solution was calculated to be 1.0 foot, a reduction of close to 60% over the Warner *et al*. (1989) results. Overall, the MER of -1.3 indicates that FTWORK slightly underestimates the Glover (1960) benchmark solution. The agreement, especially near the recharge area was significantly improved (within one foot).

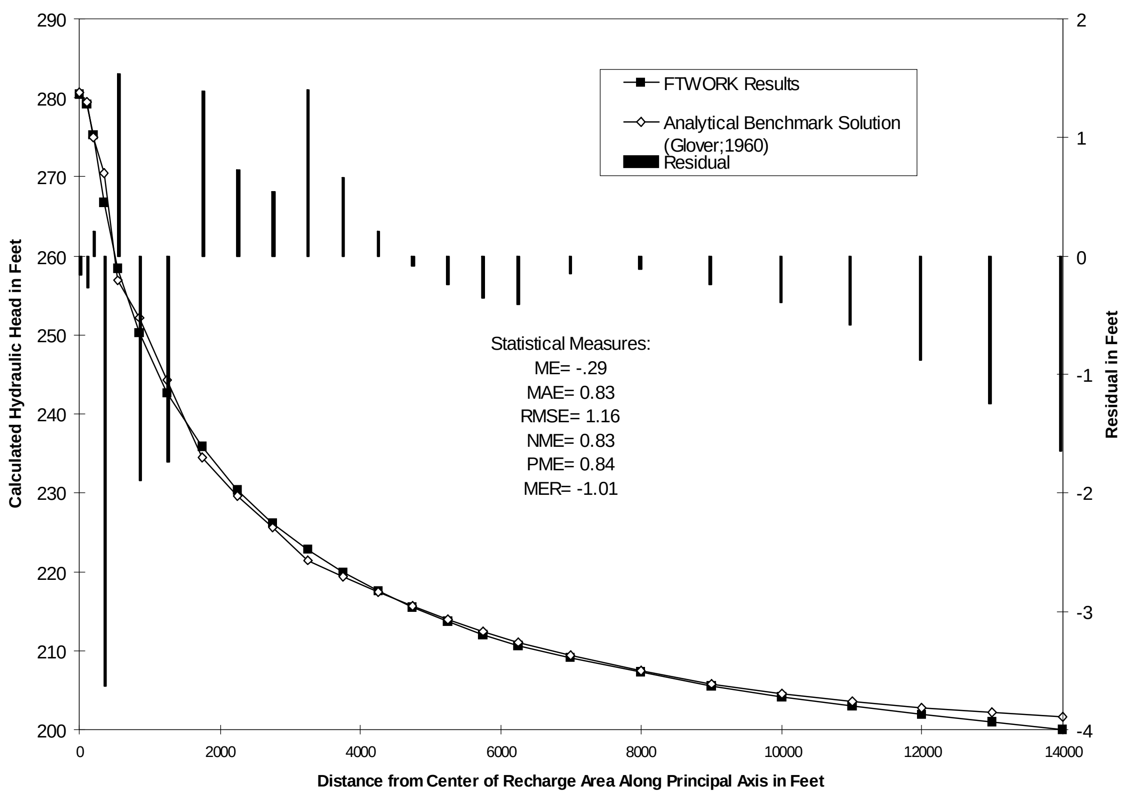


Figure 4-7. Combination plot of heads and residuals versus distance from center of recharge area for variably spaced points along centerline of grid using the Glover (1960) solution.

Using the problem setup of Figure 4-5, additional functionality testing focused on intra-comparison (Level 2A) techniques. Among others, the results generated by the areal recharge function of FTWORK were compared to results produced by the injection well function. These results indicate that FTWORK responds identically to both functions. In other words, the calculated hydraulic heads are identical when the model is subjected to areal recharge or when it is subjected to recharge introduced by an injection well with the same volumetric flux.

First-Order Decay

The documentation of FTWORK presents a test case for first-order (radioactive) decay using a decay factor of $0.0019 d^{-1}$ (Faust *et al.*, 1993, p. 105; see Appendix E). The results are compared with an analytical solution and with a zero-decay solution. To illustrate sensitivity analysis aspects of the protocol, IGWMC has performed additional runs using the same model setup as presented in Faust *et al.* (1993), decay factors ranging from λ=0.0 d^{-1} and λ=0.001 d^{-1} to λ=10.0 d^{-1}, and time steps Δt=100d and Δt=200d. Results are presented for node 10,6 (source) and node 10,10 (along plume centerline downstream of source)(see Table 4-3 and 4-4). All calculations were performed using the same numerical parameters. If the program terminates because changes in concentrations between time steps are less than a preset criterion, it advises to take a larger time step. Doing so introduces oscillations which are small for small values of the decay factor, but increase for larger values of this coefficient.

Table 4-3. Comparison of concentrations in Kg/m^3 for node 10,6 (source) of FTWORK (v.2.8B) test 4.2.3 using time steps of Δt=100 days (RADTST00-05) and Δt=200 days (RADTST10-15).

IGWMC File Name	Decay Factor [d^{-1}]	time [days]						
		200	400	600	800	1000	1200	1400
RADTST00.DAT RADTST10.DAT	0.0	8.97E-5 1.03E-4	1.04E-4 1.02E-4	1.08E-4 1.09E-4	1.10E-4 1.10E-4	1.10E-4 1.10E-4	1.11E-4 1.10E-4	1.11E-4 1.11E-4
RADTST01.DAT RADTST11.DAT	0.001	8.40E-5 9.86E-5	9.44E-5 9.25E-5	9.69E-5 9.81E-5	9.76E-5 9.74E-5	9.79E-5 9.80E-5	9.80E-5 9.79E-5	9.80E-5 9.80E-5
RADTST02.DAT RADTST12.DAT	0.010	5.09E-5 6.97E-5	no change 4.48E-5	-- 5.46E-5	-- 5.05E-5	-- 5.23E-5	-- 5.15E-5	-- 5.19E-5
RADTST03.DAT RADTST13.DAT	0.100	5.08E-6 2.19E-6	7.55E-6 4.27E-8	8.76E-6 2.14E-6	9.34E-6 8.38E-6	9.63E-6 2.11E-6	no change 1.23E-7	-- 2.06E-6

RADTST04.DAT RADTST14.DAT	1.000	8.38E-8 2.18E-6	1.61E-7 4.27E-8	2.32E-7 2.14E-6	2.99E-7 8.38E-8	3.60E-7 2.10E-6	4.16E-7 1.23E-7	4.69E-7 2.06E-6
RADTST05.DAT RADTST15.DAT	10.00	no change no change	-- --	-- --	-- --	-- --	-- --	-- --

Note: The term “no-change” relates to an FTWORK computational progress message, indicating that the calculations have been ended because the changes between two successive times are less than a set criterion or approaching zero.

Table 4-4. Comparison of concentrations in Kg/m^3 for node 10,10 (along plume centerline downstream from source) of FTWORK (v.2.8B) test 4.2.3 using time steps of Δt=100 days (RADTST00-05) and Δt=200 days (RADTST10-15).

IGWMC File Name	Decay Factor [d^{-1}]	time [days]						
		200	400	600	800	1000	1200	1400
RADTST00.DAT RADTST10.DAT	0.0	2.97E-6 2.25E-6	1.75E-5 1.57E-5	3.45E-5 2.37E-5	4.59E-5 4.67E-5	5.22E-5 5.28E-5	5.56E-5 5.60E-5	5.73E-5 5.75E-5
RADTST01.DAT RADTST11.DAT	0.001	2.46E-6 2.57E-6	1.33E-5 1.18E-5	2.38E-5 2.37E-5	2.95E-5 3.03E-5	3.20E-5 3.24E-5	3.31E-5 3.33E-5	3.36E-5 3.37E-5
RADTST02.DAT RADTST12.DAT	0.010	6.67E-10 4.68E-7	no change 1.49E-6	-- 1.84E-6	-- 1.56E-6	-- 1.61E-6	-- 1.65E-6	-- 1.61E-6
RADTST03.DAT RADTST13.DAT	0.100	5.97E-10 8.86E-10	4.94E-10 4.78E-10	4.18E-10 3.42E-10	4.12E-10 5.77E-10	4.29E-10 3.27E-10	no change 5.46E-10	-- 3.78E-10
RADTST04.DAT RADTST14.DAT	1.000	2.79E-15 1.46E-14	4.97E-15 1.53E-15	6.64E-15 1.31E-14	7.89E-15 2.89E-15	8.80E-15 1.19E-14	4.44E-15 4.09E-15	9.86E-15 1.07E-14
RADTST05.DAT RADTST15.DAT	10.00	no change no change	-- --	-- --	-- --	-- --	-- --	-- --

Note: The term “no-change” relates to an FTWORK computational progress message, indicating that the calculations have been ended because the changes between two successive times are less than a set criterion or approaching zero.

Effects of Grid Orientation on Flow

One of the known problems with numerical simulation codes which do not include cross terms for hydraulic conductivity is the sensitivity of the results for grid orientation when significant anisotropy is present. This problem can be illustrated for the case of one-dimensional flow in a single-layer, square, two-dimensional model domain, representing a confined aquifer with a thickness of 100 ft. A steady-state, uniform, one-dimensional flow field is created by specifying the head at the opposite boundaries (45 ft and 20 ft respectively), while the other two boundaries are impermeable. The resulting hydraulic gradient is 20 ft / 1000 ft. The problem is represented by three grid configurations. In configuration I, the grid of square 50 ft x 50 ft cells is oriented parallel to the flow direction. In configuration II, the grid is rotated 45 degrees with respect to the flow direction (see Figure 4-8). To be able to compare the two cases, the cells for configuration II have been set at 35.35 ft by 35.35 ft, resulting in intercell distances in the flow direction of 50 ft. The active model area consists of 21 x 21 cells, while the total number of cells is 43 x 43. In the third configuration, anisotropy is introduced by banded heterogeneity.

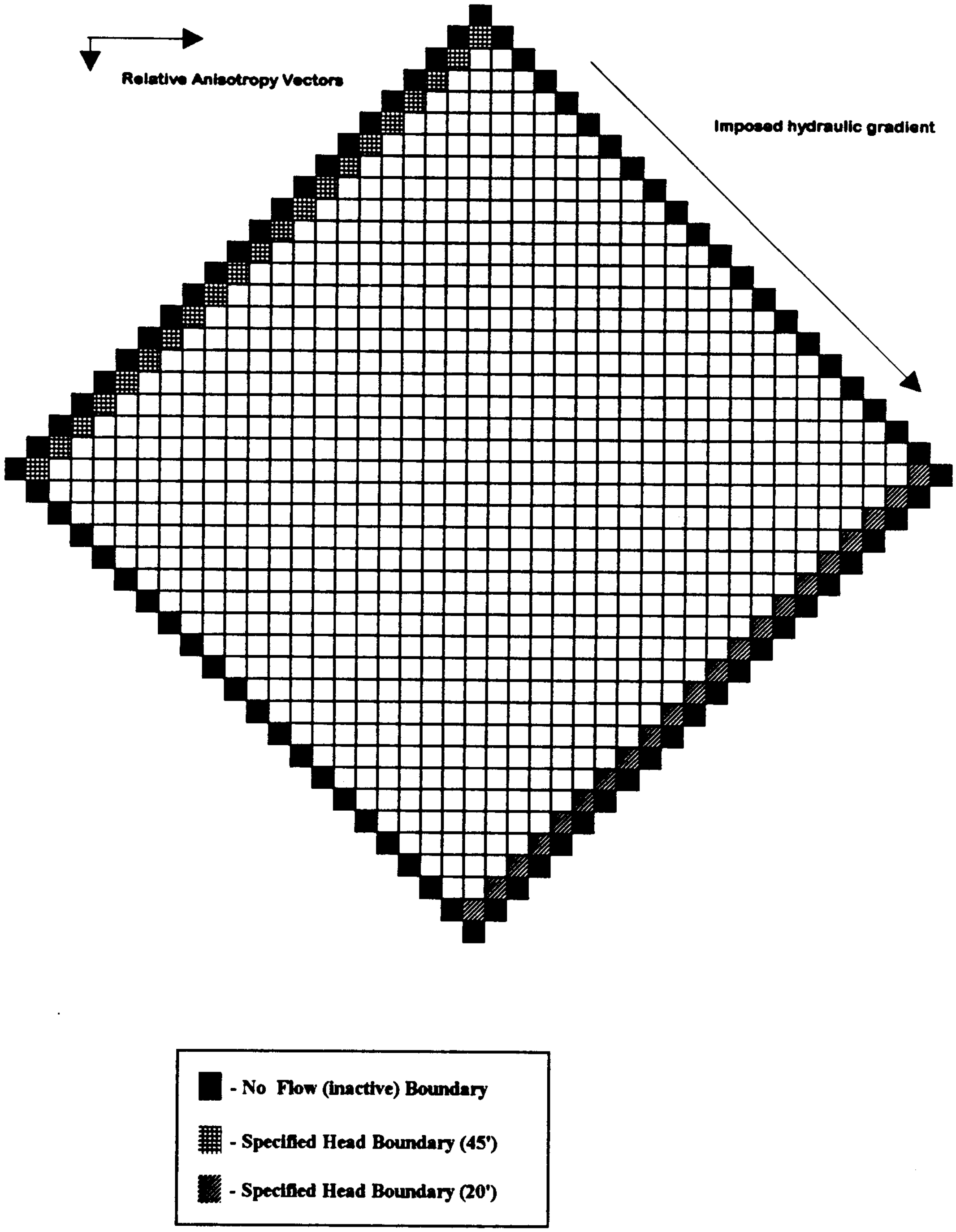


Figure 4-8. Oblique grid configuration used in anisotropy test

In the first set of simulations, hydraulic conductivity is isotropic ($K_x = K_y = K_z = 10$ ft/day). The numerical parameters are set as: the SSOR tolerance for heads = 0.001 ft, the non-linear tolerance for heads = 1.0, the non-linear weighing factor = 1.0, and the over-relaxation factor for parallel grid and oblique grid = 1.5 and 1.0 respectively. The results are shown in Figure 4-9.

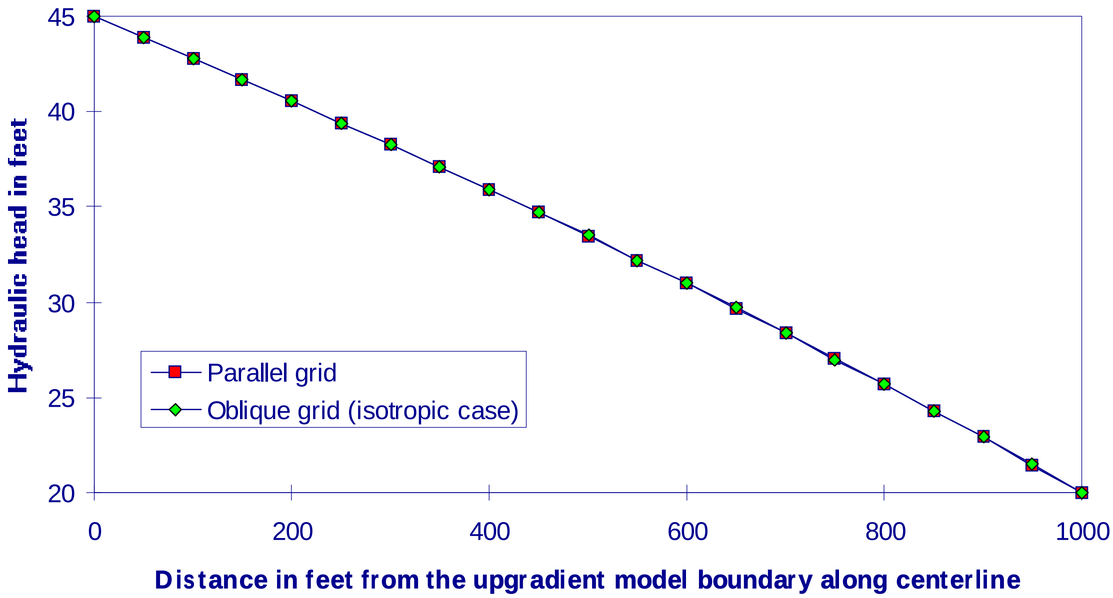


Figure 4-9. Comparison between parallel and oblique grid orientation for isotropic hydraulic conductivity.

Conceptually, the imposed hydraulic gradient results in a potentiometric surface that is uniformly sloping from the upper to the lower boundary. FTWORK approximates this very well for isotropic conditions. However, the FTWORK-produced results depart significantly from the benchmark when anisotropic conditions exist as is illustrated in the second simulation where K_x=10 ft/d, K_y = 1 ft/d, and K_z = 10 ft/d (see Figure 4-10 and Figure 4-11). Especially near the no-flow boundaries, the deviations with the benchmark are considerable.

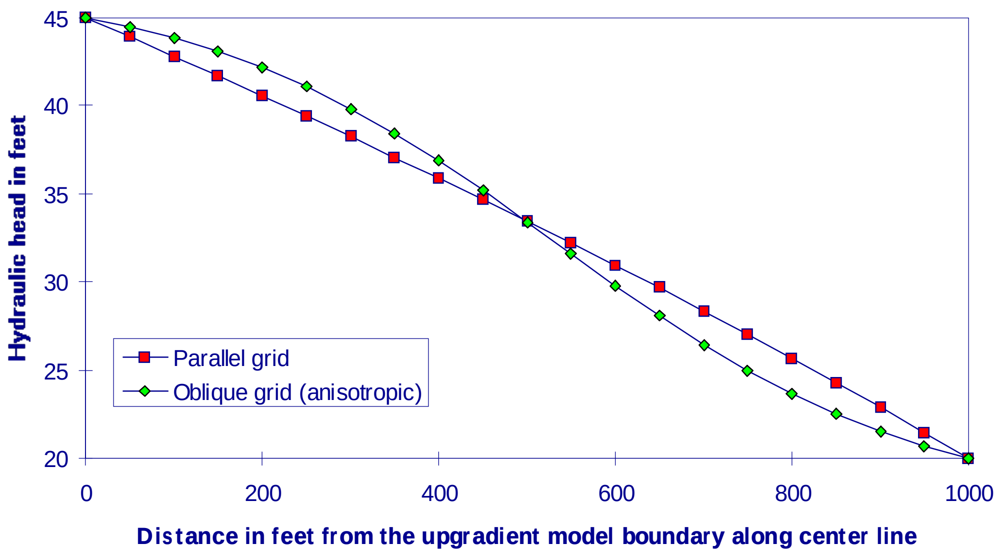


Figure 4-10. Comparison between parallel and oblique grid orientation for anisotropic hydraulic conductivity

To further investigate this issue, IGWMC has run the same test using the SIP, SSOR and PCG2 solvers in the USGS MODFLOW model (McDonald and Harbaugh, 1988). Results were almost identical, indicating that the sensitivity to grid orientation under anisotropic flow conditions is an artifact of the finite difference schemes used in these models.

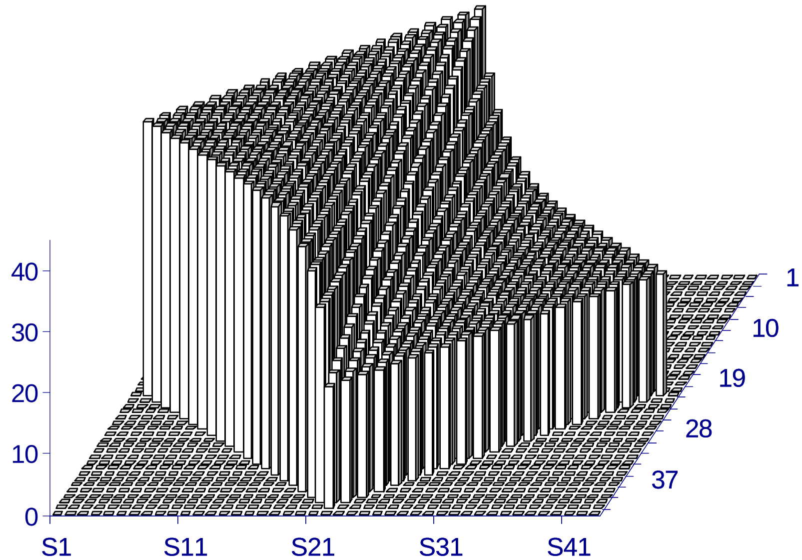


Figure 4-11. Distribution of hydraulic heads for oblique grid orientation and anisotropy (K_x = 100 ft/d, K_y = 1 ft/day, and K_z = 10 ft/d.

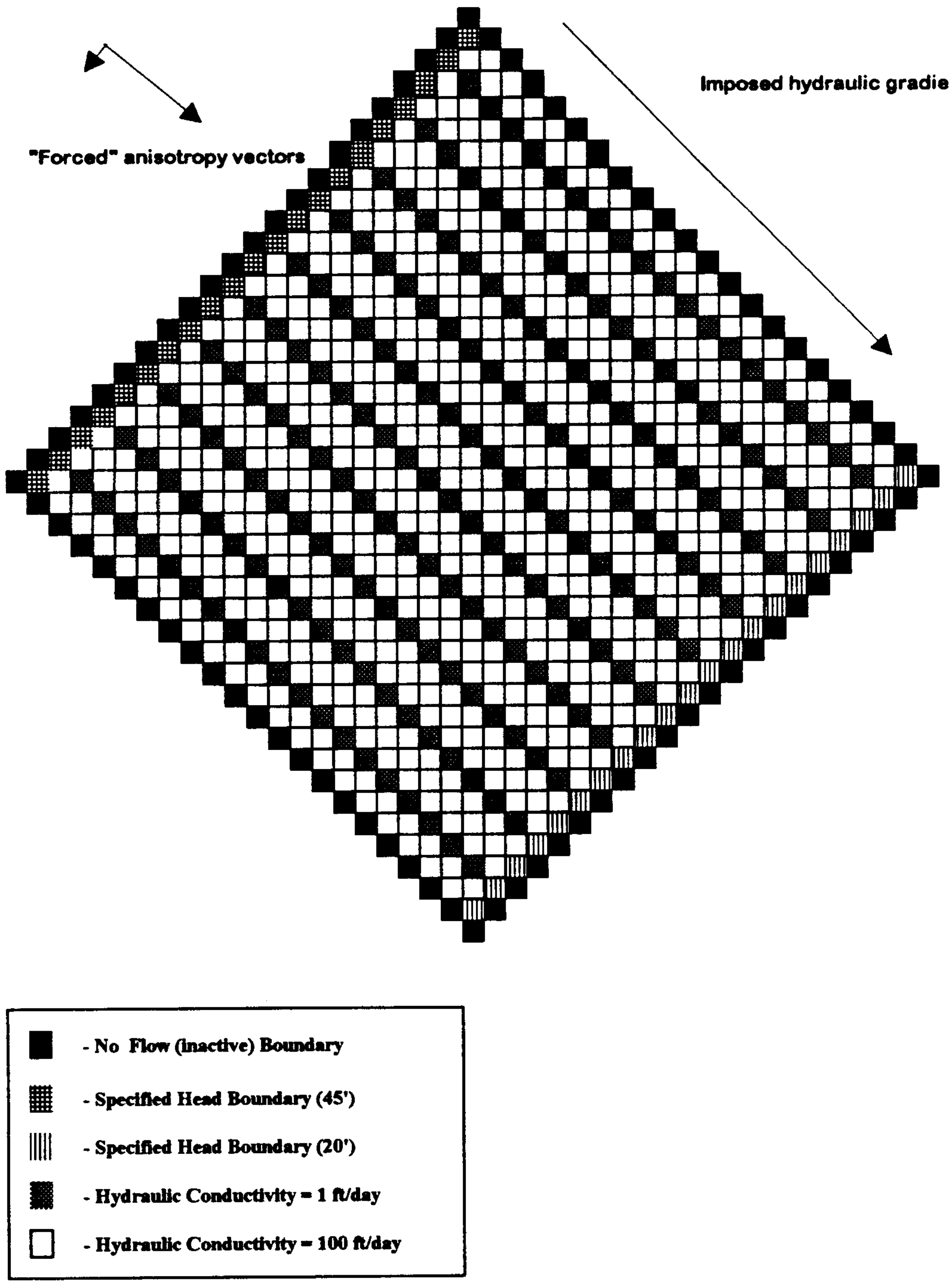


Figure 4-12. Grid design and orientation used in "forced" anisotropy test

Grid configuration III provides a different approach to simulating a scenario where principal ground-water flow direction is oblique to the grid orientation, and hydraulic anisotropy is present. Rather than explicitly defining the anisotropy properties for each cell, banded heterogeneity can be introduced to emulate directional anisotropy. For example, one can simulate directional anisotropy by defining a sequence of diagonal cells, parallel to the imposed hydraulic gradient, that have markedly lower (or higher) permeability (see Figure 4-12). Such bands of heterogeneity will result in a "forced" anisotropic pattern to the permeability distribution. The hydraulic heads computed for this configuration are presented in Figure 4-13.

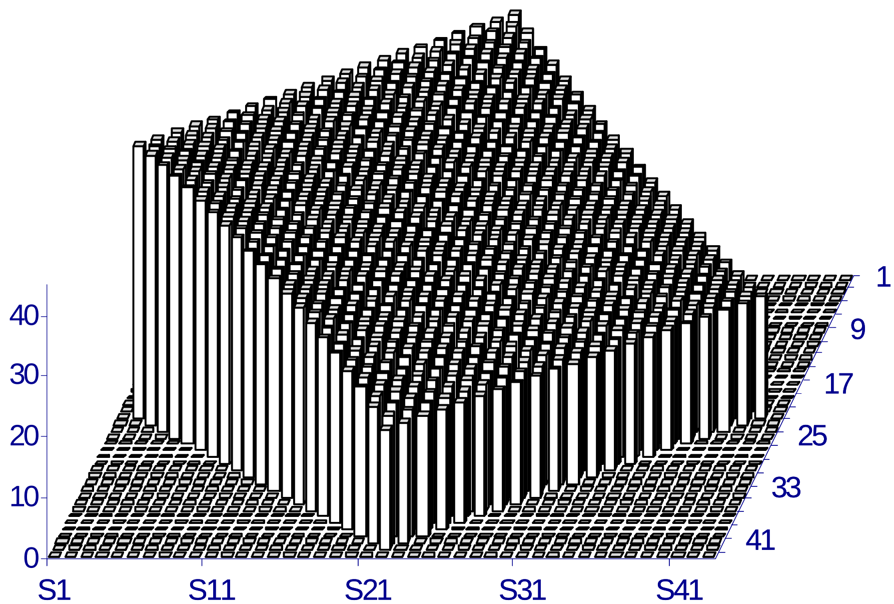


Figure 4-13. Distribution of hydraulic heads for oblique grid and anisotropy using banded heterogeneity.

5. DISCUSSION AND CONCLUSIONS

Historically, reporting on simulation code-testing has been limited to the use of author-selected verification problems. Few studies have focused on author-independent evaluation of a code, or at code intercomparison. Main deficiencies in reported code testing efforts include incompleteness of the performed testing, absence of discussion regarding tested code functions as compared with available code functions and features, and lack of detail in test problem implementation. This makes it difficult to recreate the data sets for additional analysis. The protocol presented in this report aims to address these issues. In addition, the protocol covers many other test issues, ranging from performance and resource utilization to usefulness as a decision-making support tool.

The code testing protocol consists of three components: functionality analysis, performance evaluation and applicability assessment. Functionality analysis is designed to determine the code's functions and features and to evaluate each code function for conceptual and computational correctness. Performance evaluation focuses on computational accuracy and efficiency of the code, parameter-range consistency, sensitivity of the results for model parameter uncertainty and model design, and reproducibility. Applicability assessment provides information regarding the code's ability to represent typical field problems, and the effectiveness of the code in handling such problems. The formulation of an efficient and adequate test strategy is a critical element of the protocol. Summary structures provide a quick overview of the completeness of the performed testing, while standardized statistical and graphical techniques add necessary quantitative detail to the evaluation of the results.

The code-testing protocol is designed to be applicable to all types of simulation codes dealing with fluid flow transport phenomena in the unsaturated and saturated zones of the subsurface. Selection and implementation of test problems will differ for the different types of codes. Although the preferred approach to code testing is benchmarking, for more complex codes, benchmarks are scarce and alternative test approaches, such as code intercomparison using synthetic test problems, need to be adopted. Test results are presented in a form that is unbiased by the requirements posed by specific applications. The reporting requirements of the protocol were developed to provide enough detail to establish confidence in the code's capabilities and to efficiently determine its applicability to specific field problems, without unduly burdening code developers and testers.

Because users of code-testing results may differ in terms of objectives, the protocol leaves it to the users to determine if a tested code is suitable to their needs.

The most critical element of the code testing protocol is the design of the test strategy. Many different test configurations can be used, and for some code types a large number of benchmark solutions may be available. For other code types new test problems may have to be conceived. Selection of benchmarks and design of test problems should be guided by test objectives derived during the functionality analysis step of the protocol. Specific performance evaluation issues may further determine the type of testing needed. Protocol tools such as functionality tables and functionality matrices are effective aids in the design of test problems. Well-designed tests not only identify code functionality problems, but should also provide important information on correct implementation of code features.

The practicality and usefulness of the various discussed functionality and performance evaluation measures have been assessed using the FTWORK code. Graphical evaluation measures are very illustrative for code behavior. Deviations between code results and benchmarks are easy to spot and analyze in the context of spatial location, temporal discretization, absolute value of the dependent variable, as are spatial and temporal trends in the deviations. However, graphical evaluation techniques can also be used in a very subjective way, either illustrating only elements of good performance by focusing on selected areas of the spatial or temporal domain, or highlighting problem areas. Proper use of graphical techniques means addressing both performance aspects of code behavior (*i.e.*, "the good and the bad"), if present.

Statistical techniques are usually easy to compute but difficult to assess. Most model users are not familiar enough with their values, and what these values represent, to use them effectively. However, they provide an effective measure when performing parameter sensitivity studies, code intra- and intercomparisons, and spatial and temporal resolution evaluations.

An important element of code testing is the evaluation of accuracy, stability and reproducibility for various ranges and combinations of parameter values. This issue is addressed through a carefully designed sensitivity analysis procedure, preferably using benchmark problems. A code's performance, according to the protocol, is determined not only by objective measures such as accuracy in terms of computed deviations, computational time used in deriving code-based results, and required computer memory and disk-space, but also by test problem setup and the familiarity of code testers with the particular code. Although, this report includes some measures and parameters for resource utilization requirements, they are often difficult to determine and rather subjective. Application of the performance evaluation measures, developed as part of the protocol, to the FTWORK code led to the conclusion that only the quantitative determination of computer resources utilization is recommended; the steepness of the learning curve for a particular code as well as the time required to understand the test problem and optimally implement it in an input data set can only be addressed in descriptive terms.

Assessment of a code's applicability to solving practical engineering problems and supporting regulatory decision-making focuses on those code selection criteria that have not yet been addressed during the functionality analysis part of the protocol. Applicability assessment guidance is based on the notion that well-documented example applications contribute significantly to the confidence one may have in a code's proper operation. Although originally conceived as an integral element of the protocol, it is concluded that applicability assessment is an optional aspect of the protocol, performed only when the results of illustrative field applications using comparable codes are available.

Applicability assessment of individual codes does not allow quantitative assessment of the results in the absence of an independent measure or benchmark. By standardization of applicability assessment test problems, code intercomparison may become a well-accepted alternative, especially for complex codes for which few benchmarks are available. The challenge in developing applicability assessment test cases is to describe and bound them well enough to avoid confusion during implementation for a particular code, while maintaining enough flexibility to allow optimal utilization of a particular code's features. If the problem is not described in enough detail, modeling assumptions, boundary condition assignments and parameter distribution may become incomparable between different codes. Restricting the geometry of test problems to linear and blocky features may limit the applicability assessment of codes specifically well suited to deal with curvilinear and highly irregular spatial features. Finally, the inclusion of applicability assessment tests in code documentation provides an excellent opportunity to illustrate the proper setup, parameter selection and input preparation for the particular code.

Well-designed applicability test problems are integrated functionality and performance test problems. If run with a well established and tested code in high spatial and temporal resolution, they may become a benchmark for testing other similarly featured codes. The challenge here is to determine what level of resolution is adequate. One way to approach this question is to use increasingly denser spatial and temporal discretization and compare differences between two levels of discretization at selected points in space and/or time. Theoretically, if the problem is unconditionally stable, the difference should become smaller for higher resolutions. However, determining actual discretization, especially in space, might provide a major challenge if the comparisons are to be made in a large number of fixed locations. Furthermore, such a relative accuracy versus resolution exercise requires significant computational resources. An alternative course of action is to provide various code designers with the basic problem description in terms of geometry and stresses, and have an independent group of experts evaluate the results to determine what is the "best" representation of the response surfaces.

The functionality analysis, performance evaluation and applicability assessment protocol, presented in this report, provide a comprehensive framework for systematic and in-depth evaluation of a variety of ground-water simulation codes. While allowing flexibility in implementation, it secures, if properly applied, addressing all potential coding problems. It should be noted that the protocol does not replace scientific review nor the use of sound programming principles. Most effectively, the code testing under the protocol should be performed as part of the code development process. Additional testing according to the protocol may be performed under direction of regulatory agencies, or by end-users. If properly documented, code testing according to the protocol supports effective independent review and assessment for application suitability. As such, the protocol contributes significantly to improved quality assurance in code development and use in ground-water modeling.

6. REFERENCES

Adrion, W.R., M.A. Branstad, and J.C. Cherniasky. 1986. Validation, Verification and Testing of Computer Software; In: Software Validation, Verification, Testing and Documentation, Petrocelli Books, Princeton, New Jersey.

Andersen. P.F. 1993. A Manual of Instructional problems for the U.S.G.S. MODFLOW Model. EPA/600/R-93/010, Office of Research and Development, U.S. Environmental Protection Agency, Washington, D.C.

Anderson, M.P., and W.W. Woessner. 1992. Applied Groundwater Modeling: Simulation of Flow and Advective Transport. Academic Press, Inc., San Diego, California.

Aral, M.M., and Y. Tang. 1988. A New Boundary Element Formulation for Time-Dependent Confined and Unconfined Aquifer Problems. Water Resources Research, Vol. 24(6), pp. 831-842.

ASTM. 1984. E 978: Standard Practices for Evaluating Environmental Fate Models of Chemicals. Am. Soc. for Testing and Materials, West Conshohocken, Pennsylvania.

ASTM. 1993. D 5447: Standard Guide for Application of a Ground-Water Flow Model to a Site-Specific Problem. Am. Soc. for Testing and Materials, West Conshohocken, Pennsylvania.

Bear, J. 1979. Hydraulics of Ground-Water. McGraw-Hill, New York, New York.

Beljin, M.S. 1988. Testing and Validation of Models for Simulating Solute Transport in Groundwater. GWMI 88-11. International Ground Water Modeling Center, Holcomb Research Inst., Indianapolis, Indiana.

Beljin, M.S. 1992. Review of Three Dimensional Analytical Models for Solute Transport in Ground-Water Systems. In: van der Heijde, P.K.M. and O.A. Elnawawy, Quality Assurance and Quality Control in the Development and Application of Ground-Water Models. EPA/600/R-93/011, U.S. Environm. Protection Agency, Washington, D.C.

Beljin, M.S. and L. Murdoch 1994. Analytical Models for Interceptor Trenches and Drains. GWMI 94-07. International Ground Water Modeling Center, Colorado School of Mines, Golden, Colorado.

Bredehoeft, J. D., and Konikow, L. F. 1993. Ground-water Models: Validate or Invalidate. Ground Water, Vol. 31(2), pp. 178-179.

Burnett, R.D., and E.O. Frind. 1987. Simulation of Contaminant Transport in Three Dimensions, 2, Dimensionality Effects. Water Resources Research, Vol. 23(4), pp.695-705.

Burns, L.A. 1983. Validation of Exposure Models: The Role of Conceptual Verification, Sensitivity Analysis, and Alternative Hypotheses. In: Sixth Symposium on Aquatic Toxicology and Hazard Assessment, STP 802, Am. Soc. For Testing and Materials, Philadelphia, Pennsylvania, pp. 255-281.

Chandler, R.L. and D.B. McWhorter. 1975. Upconing of Salt-Water - Fresh-Water Interface Beneath a Pumping Well. Ground Water, Vol. 13(4), pp. 354-394.

Chen, C-S. 1985. Analytical and Approximate Solutions to Radial Dispersion from an Injection Well to a Geological Unit with Simultaneous Diffusion into Adjacent Strata. Water Resources Research, Vol. 21(8), pp. 1069-1076.

Cleary, R.W., and M.J. Ungs. 1978. Analytical Models for Groundwater Pollution and Hydrology. Rept. 78-WR-15, Water Resources Program, Princeton Univ., Princeton, New Jersey.

Clifton, P.M., and S.P. Neuman. 1982. Effects of Kriging and Inverse Modeling on Conditional Simulation of the Avra Valley Aquifer in Southern Arizona. Water Resources Research, Vol. 18(4), pp. 1215-1234.

Cooley, R.L. 1972. Finite Element Solutions for Equations of Groundwater Flow. Hydrol. and Water Resourc. Public. 18, Desert Research Inst., Reno, Nevada.

Cooley, R.L., L.F. Konikow, and R.L. Naff. 1986. Non-Linear-Regression Ground-Water Flow Modeling of a Deep Regional Aquifer System. Water Resources Research, Vol. 22(13), pp. 1759-1778.

Contractor, D.N. 1981. A Two-Dimensional, Finite Element Model of Salt Water Intrusion in Groundwater Systems. Techn. Report No. 26, Water and Energy Res. Inst. Of the Western Pacific, University of Guam, Guam.

Davis, P.A., N.E. Olague, and M.T. Goodrich. 1991. Approaches for the validation of Models Used for Performance Assessment of High-Level Nuclear Waste Repositories. Report NUREG/VR-5537, SAND90-0575, Sandia Nat. Laboratories, Albuquerque, New Mexico.

Dendrou, S., and B. Dendrou. 1994. The EIS/GWM Integrated Computer Platform: Groundwater Flow Simulation Models Under the EIS/GWM Platform; Theoretical Framework and Numerical Implementation. Report Mi-94-M006. MicroEngineering, Inc., Annandale, Virginia.

DeWiest, R.J.M. 1966. Geohydrology. John Wiley & Sons, New York.

Donigian, Jr., A.S. and P.S.C. Rao. 1986. Example Model Testing Studies. In: Vadose Zone Modeling of Organic Pollutants, S.C. Hern and S.M. Melancon (eds.), pp. 103-131, Lewis Publishers, Chelsea, Michigan.

Edelman, J.H. 1972. Groundwater Hydraulics of Extensive Aquifers. Bulletin 13, Internat. Inst. for Land Reclamation and Improvement (ILRI), Wageningen, The Netherlands.

Essaid, H.I. 1990. The Computer Model SHARP, a Quasi-Three-Dimensional Finite Difference Model to Simulate Freshwater and Saltwater Flow in Layered Coastal Aquifer Systems. U.S. Geological Survey, Water-Resources Investigations Report 90-4130, Menlo Park, California.

Faust, C.R., P.N. Sims, C.P. Spalding, P.F. Andersen, and D.E. Stephenson. 1990. FTWORK: A Three-Dimensional Groundwater Flow and Solute Transport Code. WRSC-RP-89-1085, Westinghouse Savannah River Company, Aiken, South Carolina.

Faust, C.R., P.N. Sims, C.P. Spalding, P.F. Andersen, B.H. Lester, M.G. Shupe, and A. Harrover. 1993. FTWORK: Groundwater Flow and Solute Transport in Three Dimensions; Documentation Version 2.8. GeoTrans, Inc., Sterling, Virginia.

Fjeld, R.A., B.L. Sill, and B.B. Looney. 1987. The Sensitivity Index as a Screening Tool in the Analysis of Ground-Water Contaminant Transport. In: B.E. Buxton (ed.), Geostatistical, Sensitivity, and Uncertainty Methods for Ground-Water Flow and Radionuclide Transport Modeling, pp. 323-325. Battelle Press, Columbus, Ohio.

Franz, T., and N. Guiguer. 1990. FLOWPATH: A Two-Dimensional Horizontal Aquifer Simulation Model. Waterloo Hydrogeologic Software, Waterloo, Ontario, Canada.

Fried, J.J. 1975. Groundwater Pollution. Elsevier, Amsterdam, The Netherlands.

Frind, E.O. and G.E. Hokkanen. 1987. Simulation of the Borden Plume Using the Alternating Direction Galerkin Technique. Water Resources Research, Vol. 23(5), pp. 918-930.

Frind, E.O., and G.F. Pinder. 1973. Application of Galerkin's Procedure to Aquifer Analysis, Water Resources Research, Vol. 8(1), pp. 108-120.

Glover, R.E. 1960. Mathematical Derivations as Pertain to Groundwater Recharge. Agricultural Research Service, U.S Dept. of Agriculture, Ft. Collins, Colorado.

Golden Software, Inc. 1992. Grapher for Windows, User's Manual. Golden, Colorado.

Goode, D.J. 1990. Particle Velocity Interpolation in Block-Centered Finite Difference Ground-Water Flow Models. Water Resources Research, Vol. 26(5), pp. 925-940.

Gupta, S.K., C.R. Cole, C.T. Kincaid, and A.M. Monti. 1987. Coupled Fluid, Energy, and Solute Transport (CFEST) Model: Formulation and User's Manual. BMI/ONWI-660, Office of Nuclear Waste, Battelle Memorial Institute, Columbus, Ohio.

Hantush, M.S. 1960. Modification of Theory of Leaky Aquifers. Journ. Geoph. Research, 65, pp. 3713-3725.

Hantush, M.S. 1967. Growth and Decay of Groundwater Mounds in Response to Uniform Percolation. Water Resources Research, Vol. 3(1), pp. 227-234.

Haverkamp, R., M. Vauclin, J. Touma, P.J. Wierenga, and G. Vachaud,. 1977. A Comparison of Numerical Simulation Models for One-Dimensional Infiltration. Soil Science Society of America Journal, Vol. 41, pp. 285-294.

Hern, S.C., S.M. Melancon, and J.E. Pollard. 1985. Generic Steps in the Field Validation of Vadose Zone Fate and Transport Models. In: Hern, S.C., and S.M. Melancon (eds.), Vadose Zone Modeling of Organic Pollutants, pp.61-80. Lewis Publishers, Inc., Chelsea, Michigan.

Hills, R.G., K.A. Fisher, M.R. Kirkland, and P.J. Wieringa. 1994. Application of Flux-Corrected Transport to the Las Cruces Trench Site. Water Resources Research, Vol. 30(8), pp. 2377-2385.

Hoopes, J.A., and D.R.F. Harleman. 1967. Wastewater Recharge and Dispersion in Porous Media. ASCE Journ. of the Hydraulic Div., Vol. 93(HY5), pp. 51-71.

Huisman, L. 1972. Ground-Water Recovery. The McMillan Press Ltd., London, United Kingdom.

Hunt, B. 1983. Mathematical Analysis of Groundwater Resources. Butterworth & Co., London, United Kingdom.

Huyakorn, P.S., A.G. Kretschek, R.W. Broome, J.W. Mercer, and B. H. Lester. 1984a. Testing and Validation of Models for Simulating Solute Transport in Groundwater: Development, Evaluation, and Comparison of Benchmark Techniques. Report GWMI 84-13, International Ground Water Modeling Center, Holcomb Research Inst., Indianapolis, Indiana.

Huyakorn, P.S., and G.F. Pinder. 1983. Computational Methods in Subsurface Flow. Academic Press, New York.

Huyakorn, P.S., R.W. Broome, A.G. Kretschek, and J.W. Mercer. 1984b. SEFTRAN: A Simple and Efficient Flow and Transport Code. GeoTrans, Inc., Herndon, Virginia.

HYDROCOIN Work Group. 1986. HYDROCOIN Progress Report No. 4. Swedish Nuclear Power Inspectorate (SKI), Stockholm, Sweden.

HYDROCOIN Work Group. 1987. HYDROCOIN Progress Report No. 6. Swedish Nuclear Power Inspectorate (SKI), Stockholm, Sweden.

HYDROCOIN Work Group. 1988. The International HYDROCOIN Project, Level One: Code Verification. Swedish Nuclear Power Inspectorate (SKI), Stockholm, Sweden.

HYDROCOIN Work Group. 1990. The International HYDROCOIN Project, Level Two: Model Validation. Swedish Nuclear Power Inspectorate (SKI), Stockholm, Sweden.

Intera Environmental Consultants, Inc. 1983. A Proposed Approach to Uncertainty Analysis. Report ONWI-488, Battelle Memorial Inst., Columbus, Ohio.

INTRACOIN Work Group. 1984. International Nuclear Transport Code Intercomparison (INTRACOIN) Study: Final Report Level One - Code Verification. Swedish Nuclear Power Inspectorate, Stockholm, Sweden.

INTRACOIN Work Group. 1986. International Nuclear Transport Code Intercomparison (INTRACOIN) Study: Final Report Levels Two and Three - Model Validation and Uncertainty Analysis. Swedish Nuclear Power Inspectorate, Stockholm, Sweden.

Jain, M.K. 1984. Numerical Solution of Differential Equations. Halsted Press, John Wiley & Sons, New York.

Javandel, I., C. Doughty, and C.F. Tsang. 1984. Groundwater Transport: Handbook of Mathematical Models, American Geophysical Union, Water Resources Monograph 10, Washington, D.C.

Kaluarachchi, J.J., and J.C. Parker. 1989. An Efficient Finite Element Method for Modeling Multiphase Flow. Water Resources Research, Vol. 25(1), pp. 43-54.

Kinzelbach, W. 1987a. Methods for the Simulation of Pollutant Transport in Ground Water -- A Model Comparison. In: Proc. Conf. on Solving Ground Water Problems with Models, Denver, Colorado, February 10-12, 1987. pp. 656-674. Nat. Water Well Assoc, Dublin, Ohio.

Kinzelbach, W. 1987b. Numerische Methoden zur Modellierung des Schadstofftransports im Grundwasser. R.Oldenburg Verlag, München, Germany.

Konikow, L.F., and J.D. Bredehoeft. 1978. Computer Model of Two Dimensional Solute Transport and Dispersion in Ground Water. Techniques of Water-Resources Investig., Book 7, Chap. C2. U.S. Geological Survey, Reston, Virginia.

Konikow, L.F., and J.D. Bredehoeft. 1992. Ground-water Models Cannot Be Validated. Advances in Water Resources, Vol. 15, pp. 75-83.

Lahey Computer Systems, Inc.. 1992. FORTRAN User's Guide. Incline Village, Nevada.

Lapidus, L., and G.F.Pinder. 1982. Numerical Solutions of Partial Differential Equations in Science and Engineering. John Wiley & Sons, New York.

Larson, T., and S.P. Esling. 1993. Graphic Groundwater. Micro-innovations, Carbondale, Illinois.

Lenau, C.W. 1973. Contamination of Discharge Well from Recharge Well. Journ. Hydraulic Div. ASCE, Vol. 99(HY8).

Lobo Ferreira, J.P. 1988. A Comparative Analysis of Mathematical Mass Transport Codes for Ground-Water Pollution Studies. In: E. Custodio et al. (Eds.), Groundwater Flow and Quality Modelling, D. Reidel Publishing Comp., Dordrecht, The Netherlands, pp. 699-716.

Luckner, L., and W.M. Schestakow. 1991. Migration Processes in the Soil and Groundwater Zone. Lewis Publishers, Chelsea, Michigan.

Mackay, D.M., D.L. Freyberg, and P.V. Roberts. 1986. A Natural Gradient Experiment on Solute Transport in a Sand Aquifer, 1. Approach and Overview of Plume Movement. Water Resources Research, Vol. 22(13), pp. 2017-2029.

Marino, M.A., and J.N. Luthin. 1982. Seepage and Groundwater. Elsevier Scientific Publ. Comp., Amsterdam, The Netherlands.

Mathsoft, Inc. 1994. Mathcad 5.0 Plus User's Guide, Windows Version. Cambridge, Massachusetts.

McDonald, M.G., and A.W. Harbaugh. 1984. A Modular Three-Dimensional Finite Difference Ground-Water Flow Model. U.S. Geological Survey Open File Report 83-875, Reston, Virginia.

McDonald, M.G., and A.W. Harbaugh. 1988. A Modular Three-Dimensional Finite Difference Ground-Water Flow Model. Techniques of Water-Resources Investigations of the U.S. Geological Survey, Book 6, Chapter A1, Reston, Virginia.

Microsoft, Inc. 1994. Excel User's Guide. Redmond, Washington.

Milne, W.E. 1970 Numerical Solution of Differential Equations. Dover Publications, Inc., New York.

Moench, A.F., and T.A. Prickett. 1972. Radial Flow in an Infinite Aquifer Undergoing Conversion from Artesian to Water Table Conditions. Water Resources Research, v.8(4), pp. 494-499.

Molson, J.W., E.O. Frind, and C.D. Palmer. 1992. Thermal Storage in an Unconfined Aquifer: 2. Model Development, Validation, and Application. Water Resources Research, Vol. 28(10), pp. 2857-2867.

National Research Council (NRC), Committee on Ground Water Modeling Assessment, Water Science and Technology Board. 1990. Ground Water Models: Scientific and Regulatory Applications. National Academy Press, Washington, D.C.

Nicholson, T.J., T.J. McCartin, P.A. Davis, and W. Beyeler. 1987. NRC Experiences in HYDROCOIN: An International Project for Studying Ground-Water Flow Modeling Strategies. In: Proceedings, GEOVAL 87, April 4-9, 1987, Stockholm. Swedish Nuclear Power Inspectorate, Stockholm, Sweden.

Nofziger, D.L., J-S. Chen, and C.T. Haan. 1994. Evaluation of Unsaturated/Vadose Zone Models for Superfund Sites. EPA/600/R-93/184, Office of Research and Development, U.S. Environmental Protection Agency, Washington, D.C.

Papadopulos, I.S. 1965. Nonsteady Flow to a Well in an Infinite Anisotropic Aquifer. Intern. Assoc. Hydrol. Sci., Proc. Dubrovnik Symposium on the Hydrology of Fractured Rocks., pp. 21-31.

Park, N-S., and J.A. Liggett. 1991. Application of Taylor-Least Squares Finite Element to Three-Dimensional Advection-Diffusion Equation. Internat. Journ. for Num. Methods in Fluids, Vol. 13, pp. 759-773.

Peaceman, D.W. 1977. Fundamentals of Numerical Reservoir Simulation. Elsevier Scientific Publ. Comp., Amsterdam, The Netherlands.

Pinder, G.F., and J.D. Bredehoeft. 1968. Application of the Digital Computer for Aquifer Evaluation. Water Resources Research, Vol. 4(5), pp. 1069-1093.

Pollock, D.W. 1989. Documentation of Computer Programs to Compute and Display Pathlines Using Results from the U.S. Geological Survey Modular Three-Dimensional Finite-Difference Ground-Water Model. U.S.G.S. Open File report 89-381, Reston, Virginia.

Press, W.H.,m S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. 1992. Numerical Recipes in FORTRAN. Cambridge University Press, Cambridge, United Kingdom.

Prickett, T.A., and C.G. Lonnquist. 1971. Selected Digital Computer Techniques for Groundwater Resource Evaluation. Bulletin 55, Illinois State Water Survey, Champaign, Illinois.

Prickett, T.A., T.G. Naymik, and C.G. Lonnquist. 1981. A `Random-Walk' Solute Transport Model for Selected Ground Water Quality Evaluations. Bulletin 65, Illinois State Water Survey, Champaign, Illinois.

Rushton, K.R., and L.M. Tomlinson. 1977. Permissible Mesh Spacing in Aquifer Problems Solved by Finite Differences. Journ. Hydrology, Vol.34, pp. 63-76.

Sá Da Costa, A.A.G., and J.L. Wilson. 1979. A Numerical Model of Seawater Intrusion in Aquifers. Report 247, Ralph M. Parsons Lab., Massachusetts Inst. of technology, Cambridge, Massachusetts.

Segol, G. 1994. Classic Groundwater Simulations: Proving and Improving Numerical Models. PTR Prentice Hall, Englewood Cliffs, New Jersey.

Shafer, J.M. 1987. GWPATH: Interactive Ground-Water Flow Path Analysis. Bulletin 69, Illinois State Water Survey, Champaign, Illinois.

Simmons, C.S., and T.J. McKeon. 1984. INFIL1D: A Quasi-Analytical Model for Simulating One-Dimensional, Constant Flux Infiltration. Report PNL-4945, Battelle Pacific Northwest Laboratory, Richland, Washington.

Sims, P.N., P.F. Andersen, D.E. Stephenson, and C.R. Faust. 1989. Testing and Benchmarking of a Three-Dimensional Groundwater Flow and Solute Transport Model. Proceedings. Conf. Solving Ground Water Problems with Models. Indianapolis, Indiana, February 7-9, 1989. National Water Well Association, Dublin, Ohio.

Smith, L. and R.A. Freeze. 1979. Stochastic Analysis of Steady State Groundwater Flow in a Bounded Domain, 1, One Dimensional Simulations. Water Resources Research, Vol.15(3), pp. pp.521-528.

Sykes, J.F., J.L. Wilson, and R.W. Andrews. 1985. Sensitivity Analysis for Steady-State Groundwater Flow Using Adjoint Operators. Water Resources Research, Vol. 21(3), pp. 359-371.

Theis, C.V. 1935. The Relation Between Lowering of the Piezometric Surface and the Rate and Duration of Discharge of a Well Using Ground Water Storage. Trans. A. Geophys. Un., 16th Annual Meeting, Pt.2, pp. 519-524.

Thiem, G. 1906. Hydrologische Methoden (Hydrologic Methods). J.M. Gebhart, Leipzig, Germany.

Thompson, A.F.B., R. Ababou, and L.W. Gelhar. 1989. Implementation of the Three-Dimensional Turning Bands Random Field Generator. Water Resources Research, Vol. 25(10), pp. 2227-2243.

Trescott, P.C. 1975. Documentation of Finite Difference Model for Simulation of Three-Dimensional Ground-Water Flow. U.S. Geological Survey Open-File Report 75-438. Reston, Virginia.

Trescott, P.C., and S.P. Larson. 1976. Supplement to Open-File Report 75-438, Documentation of Finite Difference Model for Simulation of Three-Dimensional Ground-Water Flow. U.S. Geological Survey Open-File Report 76-591. Reston, Virginia.

Trescott, P.C., G.F. Pinder, and S.P. Larson. 1976. Finite Difference Model for Aquifer Simulation in Two Dimensions with Results of Numerical Experiments. Techniques of Water-Resources Investigations of the U.S. Geological Survey, Book 7, Chapter C1. Reston, Virginia.

Tsang, C-H. 1987. Comments on Model Validation. Transport in Porous Media, Vol. 2, pp. 623.

Tsang, C-H. 1991. The Modeling Process and Model Validation. Ground Water, Vol. 29, pp. 825.

van der Heijde, P. K. M., P.S. Huyakorn, J.W. Mercer. 1985. Testing and Validation of Ground Water Models. In: Proc. Conf. on Practical Applications of Ground Water Models, Nat. Water Well Assoc., Worthington, Ohio.

van der Heijde, P.K.M., and R.A. Park. 1986. U.S. EPA Groundwater Modeling Policy Study Group, Report of Findings and Discussion of Selected Groundwater Modeling Issues. Internat. Ground Water Modeling Center, Holcomb Research Inst., Indianapolis, Indiana.

van der Heijde, P.K.M., and O.A. Elnawawy. 1992. Quality Assurance and Quality Control in the Development and Application of Ground-Water Models. EPA 600/R-93/011, Office of Research and Development, U.S. Environmental Protection Agency, Washington, D.C.

van der Heijde, P.K.M., S.S. Paschke, and D.A. Kanzer. 1993. Ground-Water Flow and Solute Transport Model Functionality Testing and Performance Evaluation. In: H.J. Morel-Seytoux (ed.), Proc. Thirteenth AGU Hydrology Days, Fort Collins, Colorado, pp. 378-389.

van der Heijde, P.K.M. 1994. Design and Operation of a Ground-water Software Information Data Base: Model Annotation and Retrieval System 'MARS'; version 4.1. Report GWMI 94-06, Internat. Ground Water Modeling Center, Golden, Colorado.

van der Heijde, P.K.M. 1995. Analytical Solutions in Ground-Water Modeling: MathCad® Implementations. Report GWMI 95-03, Internat. Ground Water Modeling Center, Golden, Colorado.

van Genuchten, M. Th., and W.J. Alves. 1982. Analytical Solutions of the One-Dimensional Convective-Dispersive Solute Transport Equation. Tech. Bull. 1661. Agricultural Research Service, U.S. Dept. of Agriculture, Washington, D.C.

Voss, C.I. 1984. SUTRA Saturated-Unsaturated Transport; A Finite-Element Simulation Model for Saturated-Unsaturated, Fluid-Density-Dependent Ground-Water Flow with Energy Transport or Chemically Reactive Single-Species Solute Transport. U.S. Geological Survey, Reston, Virginia.

Walton, W.C. 1984. Handbook of Analytical Ground Water Models. Rept. GWMI 84-06/1. International Ground Water Modeling Center, Holcomb Research Inst., Indianapolis, Indiana..

Ward, D.S., M. Reeves, and L.E. Duda. 1984. Verification and Field Comparison of the Sandia Waste-Isolation Flow and Transport Model (SWIFT). NUREG/CR-3316 SAND83-1154, Sandia Nat. Lab., Albuquerque, New Mexico.

Warner, J.W., D. Molden, M. Chehata, and D.K. Sunada. 1989. Mathematical Analysis of Artificial Recharge from Basins. Water Resources Bull., Vol. 25(2), pp. 411.

Warrick, A.W., J.W. Biggar, and D.R. Nielsen. 1971. Simultaneous Solute and Water Transfer for an Unsaturated Soil. Water Resources Research, Vol. 7(5), pp. 1217-1218 and 1224-1225.

Weeks, J.B, and R.J. Sun. 1987. Regional Aquifer-System Analysis Program of the U.S. Geological Survey - Bibliography, 1978-1986. U.S. Geological Survey Water-Resources Investigations Report 87-4138.

Wexler, E.J. 1992. Analytical Solution for One-, two-, and Three-Dimensional Solute Transport in Ground-Water Systems with Uniform Flow. Techniques of Water-Resources Investigations of the U.S. Geological Survey, Book 3, Chapter B7, Denver, Colorado.

Witherspoon, P.A., I. Javandel, and S.P. Neuman. 1968. Use of Finite Element Method in Solving Transient Flow Problems in Aquifer Systems. Int. Assoc. of Hydrol. Sc., Symp. Publ. No. 81.

Wilson, J.L., L.R. Townley, and A. Sa Da Costa. 1979. Mathematical Development and Verification of a Finite Element Aquifer Flow Model AQUIFEM-1. TAP Report 79-2, Mass. Inst. Of Technology, Cambridge, Massachusetts.

Zheng, C. 1989. PATH3D: A Ground-Water Path and Travel-Time Simulator. S.S. Papadopulos and Assoc., Inc., Bethesda, Maryland.

Zheng, C., 1993. MT3D: A Modular Three-Dimensional Transport Model for Simulation of Advection, Dispersion and Chemical Reaction of Contaminants in Groundwater Systems. S.S. Papadopulos and Assoc., Inc., Bethesda, Maryland.

7. GROUND-WATER MODELING TERMINOLOGY

This list has been compiled by the International Ground Water Modeling Center (IGWMC) of the Colorado School of Mines, Golden, Colorado. It includes terms which have been approved by the American Society for Testing and Materials (ASTM).

Acceptance Criteria

- preset criteria to determine whether a (site- or problem-specific) model's predictive capability is acceptable for the intended use.

Analytic Element Method (AEM)

- a method for approximating the solution of the ground-water flow equation based on the superposition of suitable closed-form analytical functions.

Analytical Function Method (AFM)

- a method for approximating the solution of the ground-water flow equation using analytical functions with degrees of freedom so that a flow pattern is generated that satisfies the boundary conditions at all points of an approximate boundary.

Analytical Method (AM)

- a set of mathematical procedures used to obtain analytical solutions of the governing equations; examples of such procedures are: infinite series, integral transformations, and complex variables.

Analytical Model

- in subsurface fluid flow, a model that uses closed form solutions to the governing equations applicable to ground-water flow and transport processes.

Analytical Solution

- a closed form (explicit) solution of the governing equation, continuous in space and time, sometimes requiring tabular or numerical evaluation.

Analog Model

- a model based on a one-to-one correspondence between the hard-to-observe natural system (e.g., ground-water system) and another phenomenon that is easier to observe, and between the excitation and response functions of both systems (e.g., membrane analog, electric analog, Hele Shaw analog).

Application Verification

- using the set of parameter values and boundary conditions from a calibrated model to approximate acceptably a second set of field data measured under similar hydrologic conditions.

  Discussion- Application verification is to be distinguished from code verification, which refers to software testing, comparison with analytical solutions, and comparison with other similar codes to demonstrate that the code represents its mathematical foundation.

Aquifer

- a geologic formation, group of formations, or part of a formation that is saturated and is capable of providing a significant quantity of water.

- *aquifer, confined* - an aquifer bounded above and below by confining beds in which the static head is above the top of the aquifer.

- *aquifer, unconfined* - an aquifer that has a water-table.

Benchmark
- an independently derived reference solution for a stated problem against which the performance of a computer code is evaluated; often in the form of an analytical solution.

Benchmarking
- the process of using reference solutions against which the performance of a computer code is evaluated.

Block
- a three-dimensional model unit having a regular geometry and uniform properties representing a physical portion of a ground-water or vadose water system; used with the finite difference method (see also cell).

Block-Centered Grid
- discretization of the model domain for use with the finite-difference method in a manner that the nodes, where the dependent variable is calculated, are placed at the center of the block (or cell). System parameters are assumed to be uniform over the extent of the block. Specified-head boundaries are located at the nodes; flux boundaries are located at the edge of the block.

Boundary
- geometrical configuration of the surface enclosing the model domain.

Boundary Condition

- a mathematical expression of the state of the physical system that constrains the equations of the mathematical model.

Note: Boundary conditions are values for the dependent variable (Dirichlet or first kind), the derivatives of the dependent variable (Neumann or second kind), or a combination of both (Cauchy or third kind) representing the state of a physical system along its boundaries.

For saturated flow: values for head or pressure (specified head condition; Dirichlet or first kind), the head or pressure gradient (specified flux condition; Neumann or second kind), or a combination of both (head-dependent flux condition; Cauchy or third kind) representing the state of the flow system along its natural boundaries.

For unsaturated flow: values for head, pressure, suction or moisture content (specified head or moisture content condition; Dirichlet or first kind), the gradient of head, pressure, suction or moisture content (specified flux condition; Neumann or second kind), or a combination of both (head/water content dependent flux condition; Cauchy or third kind) representing the state of the flow system along its natural boundaries.

For solute transport: values for concentration (specified concentration condition; Dirichlet or first kind), the solute flux (specified solute or mass flux condition; Neumann or second kind), or a combination of both (concentration dependent mass flux condition; Cauchy or third kind) representing the state of the solute transport along the natural boundaries of the ground-water system.

Boundary Element

- a point or section of the model boundary representing a specific boundary condition.

Boundary Element Method (BEM)
- see Boundary Integral Equation Method.

Boundary Integral Equation Method (BIEM)
- a method in which the boundary value problem is expressed in terms of an integral equation; this equation is solved by approximating the boundary by a series of straight lines (elementary curves) or flat surfaces (elementary surfaces), and making simplifying assumptions regarding the behavior of the solution along boundary segments or elements.

Calibrated Model
- a model for which all residuals between calibration targets and corresponding model outputs, or statistics computed from residuals, are less than pre-set acceptable values.

Calibration
- the process of refining the model representation of the hydrogeologic framework, hydraulic properties, and boundary conditions to achieve a desired degree of correspondence between the model simulations and observations of the ground-water flow system.

Calibration Criteria
- qualitative and quantitative measures used in the calibration process to measure the progress in the calibration process.

Calibration Targets
- measured, observed, calculated, or estimated hydraulic heads or ground-water flow rates that a model must reproduce, at least approximately, to be considered calibrated.

  Discussion - The calibration target includes both the value of the head or flow rate and its associated error of measurement, so that undue effort is not expended attempting to get a model application to closely reproduce a value which is known only to within an order of magnitude

Calibration Value
- field-measured values of dependent or derived variables used in the calibration process to obtain calibration residuals (e.g., heads, concentrations, mass fluxes, and velocities).

Capillary Fringe
- the basal region of the vadose zone comprising sediments that are saturated, or nearly saturated, near the water table, gradually decreasing in water content with increasing elevation above the water table.

Cell
- also called *element*, a distinct one- two- or three-dimensional model unit representing a discrete portion of a physical system.

  Note: Although in most model formulations a cell has uniform properties assigned, some model formulations allow for the model properties to vary within a cell according to a linear or nonlinear function.

Censored Data
- knowledge that the value of a variable in the physical hydrogeologic system is less than or greater than a certain value, without knowing the exact value.

Discussion- for example, if a well is dry, than the potentiometric head at that place and time must be less than the elevation of the screened interval of the well although its specific value is unknown.

Code

- see computer code.

Code Selection

- the process of choosing the appropriate computer code, algorithm, or other analysis technique capable of simulating those characteristics of the physical system required to fulfill the modeling project's objective(s).

Code Testing

- execution of test problems to evaluate computer code performance.

Code Validation

- the process of determining how well a ground-water modeling code's theoretical foundation and computer implementation describe actual system behavior in terms of the degree of correlation between calculated and independently observed cause-and-effect responses of the reference ground-water system for which the code has been developed.

Note 1: The term “validation” in ground-water modeling means different things to different people. In software engineering, code validation is a well-established term, defined as "..... the determination of the correctness of the final software product with respect to user needs and requirements." Applying this definition to ground-water modeling software, ground-water modeling code validation is the process of determining how well the code's theoretical foundation and computer implementation describe actual system behavior in terms of the degree of correlation between code computations and independently derived observations of the cause-and-effect responses of reference ground-water system.

Note 2: Code validation in ground-water modeling, as defined above, is by nature a subjective and open-ended process; the result of the code validation process is a level of confidence in the code’s ability to simulate the reference system, or the determination of the code’s inability to simulate such a system. As there is no practical way to determine that a ground-water modeling code correctly simulates all variants of the reference system, the code can never be considered “validated.”

Code Verification

- the process of demonstrating the consistency, completeness, correctness and accuracy of a ground-water modeling code with respect to its design criteria by evaluating the functionality and operational characteristics of the code and testing embedded algorithms and internal data transfers through execution of problems for which independent benchmarks are available.

Note 1: In software engineering, verification is the process of demonstrating consistency, completeness, and correctness of the software. ASTM Standard E978 defines verification as "..... the examination of the numerical technique in the computer code to ascertain that it truly represents the conceptual model and that there are no inherent problems with obtaining a solution". Applying these definitions to ground-water modeling software, the objective of the code verification process is threefold: 1) to check the correctness of the program logic and the computational accuracy of the algorithms used to solve the governing equations; 2) to assure that the computer code is fully operational (no programming errors); and 3) to evaluate the performance of the code with respect to all its designed and inherent functions.

Note 2: A code can be considered "verified" when all its functions and operational characteristics have been tested and have met specific performance criteria, established at the beginning of the verification procedure. Considering a code verified does not imply that a ground-water model application constructed with the code is verified.

Compartmentalization
- division of the environment into discrete locations in time or space.

Computer Code (computer program)
- the assembly of numerical techniques, bookkeeping, and control language that represents the model from acceptance of input data and instructions to delivery of output.

Conceptual Error
- a modeling error where model formulation is based on incorrect or insufficient understanding of the modeled system.

Conceptual Model
- an interpretation or working description of the characteristics and dynamics of the physical system.
- a qualitative interpretation or working description of the geometry, characteristics and dynamics of a physical system in terms of system elements, operative processes, interlinkages and hierarchy of these elements and processes, and system stresses, bounds, and responses.

Confining Bed (Confining Unit)
- *confining bed* - a hydrogeologic unit of less permeable material bounding one or more aquifers.
- *confining unit* - a body of relatively low permeable material stratigraphically adjacent to one or more aquifers.

Constant-Head Boundary
- the conceptual representation of a natural feature such as a lake or river that effectively fully penetrates the aquifer and prevents water-level changes in the aquifer at that location.

Constant Head Node
- a location in the discretized ground-water flow model domain where the hydraulic head remains the same over the time period considered; see also specified head.

Constitutive Coefficients and Parameters
- type of model input that is not directly observable, but, rather, must be inferred from observations of other model variables; for example, the distribution of transmissivity, specific storage, porosity, recharge, and evapotranspiration.

Contaminant Fate
- chemical changes and reactions that change the chemical nature of the contaminant, effectively removing the contaminant from the subsurface hydrologic system.

Contaminant Transformation
- chemical reactions which change the chemical nature and properties of the contaminating compound.

Contaminant Transport Model
- a model describing the movement of contaminants in the environment.

Control Parameter or Variable
- an input parameter instructing the computer regarding the execution of code options.

Coupled Models (see also linked models)
- a model that contains two or more processes described by separate governing equations, the solutions of which are interdependent.

  Note: For example, models that are based on both a flow and a solute transport equation, the solution of which is coupled through concentration-dependent density effects on the flow, and flow-related advection and dispersion effects on the solute movement.

Deterministic Process
- a process in which there is an exact mathematical relationship between the independent and dependent variables in the system.

Deterministic System
- a system defined by definite cause-and-effect relations.

Deviations
- see residuals

Digital model
- (obsolete term) see computer model.

Direct problem
- computing outputs of a physical system from specified inputs and parameters.

Discretization
- division of the model and/or time domain into distinct subdomains accessible for numerical approximation of the governing equations.

Discretization Error
- modeling error due to incorrect or improper design of a grid or mesh; such errors may be related to the location of the nodes, the size of the grid elements or cells, or the geometry of the grid or individual cells.

Dispersivity
- a scale-dependent aquifer parameter that determines the degree to which a dissolved constituent will spread in flowing ground water.

Distributed-Parameter Model
- a model which takes into account the detailed spatial variations in properties, behavior, or response surface of the simulated system.

Element
- see cell.

Equipotential Line
- a line connecting points of equal hydraulic head. A set of such lines provides a contour map of a potentiometric surface.

Fidelity
- the degree to which a model application is designed to be realistic.

Field Characterization
- a review of historical, on- and off-site, as well as surface and sub-surface data, and the collection of new data to meet project objectives; field characterization is a necessary prerequisite to the development of a conceptual model.

Finite Difference Method (FDM)
- a discrete technique for solving the given partial differential equation (PDE) by 1) replacing the continuous domain of interest by a finite number of regular-spaced mesh- or grid-points (i.e., nodes) representing volume-averaged sub-domain properties, and 2) by approximating the derivatives of the PDE for each of these points using finite differences; the resulting set of linear or nonlinear algebraic equations is solved using direct or iterative matrix solving techniques.

Finite Difference Model
- a type of numerical model that uses a mathematical technique called finite-difference method to obtain an approximate solution to the governing partial differential equation (in space and time).

Finite Element Method (FEM)
- a discrete technique for solving the given partial differential equation (PDE) wherein the domain of interest is represented by a finite number of mesh- or grid-points (i.e., nodes), and information between these points is obtained by interpolation using piecewise continuous polynomials; the resulting set of linear or nonlinear algebraic equations is solved using direct or iterative matrix solving techniques.

Finite Element Model
- a type of numerical model that uses a technique called the finite-element method to obtain an approximate solution to the governing partial differential equation (in space and sometimes time).

Fixed Head, Concentration, or Temperature
- see specified head, concentration or temperature

Fixed Flux
- see specified flux

Flow Path
- represents the area between two flow lines along which ground water can flow.

Flux
- the volume of fluid crossing a unit cross-sectional surface area per unit time.

Forcing Terms
- see hydrologic stress

Forecasting
- predictive simulation of time-dependent system responses at some period in the future.

Functionality (of a ground-water modeling code)
- the set of functions and features the code offers the user in terms of model framework geometry, simulated processes, boundary conditions, and analytical capabilities and operational capabilities.

Functionality Testing
- testing a generalized computer code to establish that the code's functions (as represented by the mathematical model) and its design features are correctly implemented.

Generic Simulation Model
- the (generalized) computer code representing a (generalized) mathematical model usable for different site- or problem-specific simulations.

Grid
- see model grid

Grid Block
- see block

Ground Water
- that part of the subsurface water that is in the saturated zone. Note - Loosely, all subsurface water as distinct from surface water.

Ground-Water Barrier
- soil, rock, or artificial material which has a relatively low permeability and which occurs below the land surface where it impedes the movement of ground water and consequently causes a pronounced difference in the potentiometric level on opposite sides of the barrier.

Ground-Water Basin
- a ground-water system that has defined boundaries and may include more than one aquifer of permeable materials, which are capable of furnishing a significant water supply. Note - a basin is normally considered to include the surface area and the permeable materials beneath it. The surface-water divide need not coincide with a ground-water divide.

Ground-Water Discharge
- the water released from the zone of saturation; also the volume of water released.

Ground-Water Flow
- the movement of water in the zone of saturation.

Ground-Water Flow Model
- an application of a mathematical model to represent a regional or site-specific ground-water flow system.

Ground-Water Flow System
- a water-saturated aggregate of rock, in which water enters and moves, and which is bounded by rock that does not allow any water movement, and by zones of interaction with the earth's surface and with surface water systems; a ground-water flow system has two basic hydraulic functions: it is a reservoir for water storage, and it serves as a conduit by facilitating the transmission of water from recharge to discharge areas, integrating various inputs and dampening and delaying the propagation of responses to those inputs; a ground-water flow system may transport dissolved chemical constituents and heat.

Ground-Water Model
- see ground-water model application.

Ground-Water Model Application

- a non-unique, simplified mathematical description of one or more subsurface components of a local or regional hydrologic system, coded in a computer programming language, together with a quantification of the simulated system in the form of framework geometry, boundary conditions, system and process parameters, and system stresses.

  Discussion - As defined above, a ground-water model application is a representation of an actual hydrologic system; it should not be confused with the generic computer code used in formulating the ground-water model. This standard concerns only the development, testing and documentation of generic simulation computer codes, not ground-water model applications.

Ground-Water Modeling

- the process of developing ground-water models.

Ground-Water Modeling Code

- the non-parameterized computer code used in ground-water modeling to represent a non-unique, simplified mathematical description of the physical framework, geometry, active processes, and boundary conditions present in a reference subsurface hydrologic system.

Ground-Water Recharge

- the process of water addition to the saturated zone; also the volume of water added by this process.

Ground-Water System

- see ground-water flow system.

Head (Total; Hydraulic Head)

- the sum of three components at a point: (1) elevation head, h, which is equal to the elevation of the point above a datum; (2) pressure head, h_p, which is the height of a column of static water that can be supported by the static pressure at the point; and (3) velocity head, h_v, which is the height the kinetic energy of the liquid is capable of lifting the liquid.

Hindcasting

- predictive simulation of time-dependent system responses at some period back in the past.

History Matching

- is calibration using time series of the dependent variable or derivatives thereof at specific locations.

Hydraulic Conductivity

- the volume of water at the existing kinematic viscosity that will move in a unit time under unit hydraulic gradient through a unit area measured at right angles to the direction of flow.

Hydraulic Gradient

- the change in total hydraulic head of water per unit distance of flow.

Hydraulic Head

- see head, total

Hydraulic Properties
- *hydraulic properties* - intensive properties of soil and rock that govern the transmission (that is, hydraulic conductivity, transmissivity, and leakance) and storage (that is, specific storage, storativity, and specific yield) of water.

Hydrologic Boundaries
- physical boundaries of a hydrologic system

Hydrologic Condition
- *hydrologic condition* - a set of ground-water inflows or outflows, boundary conditions, and hydraulic properties that cause potentiometric heads to adopt a distinct pattern.

Hydrologic Properties
- properties of soil and rock that govern the entrance of water and the capacity to hold, transmit, and deliver water, e.g. porosity, effective porosity, specific retention, permeability, and direction of maximum and minimum permeability.

Hydrologic Stress
- natural or anthropogenic excitation of the hydrologic system.

Hydrologic System
- the general concepts of the hydrologic elements, active hydrologic processes, and the interlinkages and hierarchy of elements and processes.

Hydrologic Unit
- geologic strata that can be distinguished on the basis of capacity to yield and transmit fluids; aquifers and confining units are types of hydrologic units; boundaries of a hydrologic unit may not necessarily correspond either laterally or vertically to lithostratigraphic formations.

Hydrostratigraphic Unit
- see hydrologic unit

Image Well
- an imaginary well located opposite a control well such that a boundary is the perpendicular bisector of a straight line connecting the control and image wells; used to simulate the effect of a boundary on water-level changes.

Impermeable Boundary
- the conceptual representation of a natural feature such as a fault or depositional contact that places a boundary of significantly less-permeable material laterally adjacent to an aquifer.

Indirect Problem
- see inverse problem.

Initial Conditions
- the state of the physical system at the beginning of the time domain for which a solution of the governing equations is sought, expressed in terms of the dependent variable.

Input Estimation
- the process of selecting appropriate model input values (see also model construction).

Integral Finite Difference Method

- (sometimes called Integrated Finite Difference Method) a discrete technique for solving the given partial differential equation (PDE) by 1) explicit partitioning of the continuous domain of interest in a finite number of irregular-shaped sub-domains each containing a mesh- or grid-point (i.e., node) representing volume averaged sub-domain properties; and 2) by approximating the derivatives in the PDE for each of these points using finite differences; the resulting set of linear or nonlinear algebraic equations are solved using direct or iterative matrix solving techniques.

Inverse Method
- a method of calibrating a ground-water flow model using a computer code to systematically vary inputs or input parameters to minimize residuals or residual statistics.

- the procedure to estimate model parameters by minimizing the difference between measured and computed model outputs through systematic modification of model inputs.

Kriging
- a geostatistical interpolation procedure for estimating spatial distributions of model inputs from scattered observations.

Linked Models (see also coupled models)
- a model that contains two or more processes described by separate governing equations, the solution of one or more of which is dependent on the solution of another.

  Note: For example, models that are based on both a flow and a solute transport equation and where the solution of the transport equation is linked to the solution of the flow equation through flow-related advection and dispersion effects on the solute movement, without the solution of the flow equation being influenced by the solution of the transport equation.

Lumped-Parameter Model
- model in which spatial variations in the properties, behavior, or response surface of the simulated system are ignored.

Mathematical Model
- (a) mathematical equations expressing the physical system and including simplifying assumptions; (b) the representation of a physical system by mathematical expressions from which the behavior of the system can be deduced with known accuracy.

Matric Potential
- the energy required to extract water from a soil against the capillary and adsorptive forces of the soil matrix.

Matric Suction
- for isothermal soil systems, matric suction is the pressure difference across a membrane separating soil solution, in-place, from the same bulk (see soil-water pressure).

Mesh
- see model grid

Mesh-Centered Grid
- discretization of the model domain for use with the finite-difference method in a manner that the nodes, where the dependent variable is calculated, are placed at the intersections of blocks (or cells). System parameters are assumed to be uniform over the area or volume equating to half the distance between nodes. The boundary coincides with nodes;both specified-head and flux boundaries are always located directly at the nodes.

Method of Characteristics (MOC)

- a numerical method for solving hyperbolic partial differential equations as encountered in transient ground-water flow and subsurface solute transport, among others, by replacing them with an equivalent system of ordinary differential equations (characteristics).

Method of Images

- use of symmetry and superposition of solutions of linear governing partial differential equations to analyze effects of boundaries and internal discontinuities of simple geometric configuration on the distribution of heads and concentrations; allows application of solutions for an infinite space to be used in finite domains.

Model

- an assembly of concepts in the form of mathematical equations that portray understanding of a natural phenomenon.

- a representation of a system or process to facilitate observation of the system, formulation of hypotheses and theories regarding the structure and operation of the system, and analysis of the effects of manipulating the system.

Model Application

- see ground-water model.

Model Construction

- the process of transforming the conceptual model into a parameterized mathematical form; as parameterization requires assumptions regarding spatial and temporal discretization, model construction requires a-priori selection of a computer code.

Model Domain

- the volume of the physical system for which the computation of the state variable is desired.

Model Grid

- a system of connected nodal points superimposed over the problem domain to spatially discretize the problem domain into cells (finite difference method) or elements (finite element method) for the purpose of numerical modeling.

Modeling

- the process of formulating a model of a system or process.

Model Input

- the constitutive coefficients, system parameters, forcing terms, auxiliary conditions, and program control parameters required to apply a computer code to a particular problem.

Modeling Objectives

- the purpose(s) of a model application.

Model Output

- see output.

Model Representation

- a conceptual, mathematical or physical depiction of a field or laboratory system.

Model Testing

- see code testing.

Model Validation

- in code development (see also code validation): the process of determining how well a model's theoretical foundation and computer implementation describe actual system behavior in terms of the "degree of correlation" between calculated and independently observed cause-and-effect responses of the prototype real-world ground-water system (or research site or problem) for which the generic (or generalized) simulation model has been developed. Model validation represents the final step in determining the applicability of the quantitative relationships derived for the real-world prototype system the model is designed to simulate.

  Note: The results of model validation should not be expressed in terms of a generic simulation model's unconditional validity, but rather in terms of the model's applicability to specific type of systems, subject to specific conditions.

- in model application: evaluating the predictive accuracy of a model performed by comparing model predictions to field measurements collected after publication of the model study (see post audit).

Model Verification

- in model application: a) the procedure of determining if a (site-specific) model's accuracy and predictive capability lie within acceptable limits of error by tests independent of the calibration data; b) in model application: using the set of parameter values and boundary conditions from a calibrated model to acceptably approximate a second set of field data measured under similar hydrologic conditions.

- in code testing: see code verification.

Node (Nodal Point)

- in a numerical model, a location in the discretized model domain where a dependent variable is computed.

No-Flow Boundary

- boundary where specified flux condition applies with flux equal zero.

Numerical Methods

- a set of procedures used to solve the equations of a mathematical model in which the applicable partial differential equations are replaced by a set of algebraic equations written in terms of discrete values of state variables at discrete points in space and time.

  Discussion - There are many numerical methods. Those in common use in ground-water models are the finite-difference method, the finite-element method, the boundary element method, and the analytic element method.

Numerical Model

- in subsurface fluid flow modeling, a model that uses numerical methods to solve the governing equations of the applicable problem.

Numerical Solution

- an approximative solution of a governing (partial) differential equation derived by replacing the continuous governing equation with a set of equations in discrete points of the model's time and space domains.

Over-Calibration

- achieving artificially low residuals by inappropriately fine-tuning model input parameters and not performing application verification.

Output
- in subsurface fluid flow modeling, all information that is produced by the computer code.

Parameter
- any of a set of physical properties which determine the characteristics or behavior of a system.

Parameter Estimation
- see input estimation

Parameter Identification
- determining parameter distributions by analyzing the responses of a system to stresses.

Parameter Identification Model
- (sometimes called parameter estimation model or inverse model) a computer code for determination of selected unknown parameters and stresses in a ground-water system, given that the response of the system to all stresses is known and that information is available regarding certain parameters and stresses.

Perched Ground Water
- unconfined ground water separated from an underlying body of ground water by an unsaturated zone.

Percolation
- the movement of water through the vadose zone, in contrast to infiltration at the land surface and recharge across the water table.

Performance Criteria
- see acceptance criteria.

Performance Measures
- informative and efficient measures for use as in evaluation of a code's (generic) predictive capability; such measures characterize accuracy and stability of the solution derived with the code over total space and time domains appropriate for the code, and for the full range of parameter values that might be encountered in the systems for which the code has been developed.

Performance Target
- a measure of model accuracy; see also acceptance criteria.

Performance Testing
- (also performance evaluation) determining for the range of expected uses of the generic simulation code, its accuracy, efficiency, reliability, reproducibility, and parameter sensitivity by comparing code results with predetermined benchmarks.

Post Audit
- comparison of model predictions to field measurements collected after the predictions have been published, and subsequent analysis of differences in residuals.

Postprocessing
- using computer programs to analyze, display and store results of simulations.

Potentiometric Surface

- an imaginary surface representing the static head of ground water. The water table is a particular potentiometric surface.

  Discussion - where the head varies with depth in the aquifer, a potentiometric surface is meaningful only if it describes the static head along a particular specified surface or stratum in that aquifer. More than one potentiometric surface is required to describe the distribution of head in this case.

Predictive Simulation
- solution of the forward mathematical problem by specifying system parameters and calculating system responses (either steady-state or transient).

Preprocessing
- using computer programs to assist in preparing data sets for use with generic simulation codes; may include grid generation, parameter allocation, control parameter selection, and data file formatting.

Prescribed Head, Concentration or Temperature
- see specified head, concentration and temperature.

Prescribed Flux
- see specified flux.

Pressure Head
- the head of water at a point in a porous system; negative for unsaturated systems, positive for saturated systems. Quantitatively, it is the water pressure divided by the specific weight of water.

Probabilistic Model
- see stochastic model.

Program
- see computer code.

Quality Assurance in Code Development (QA)
- the procedural and operational framework put in place by the organization managing the code development project, to assure technically and scientifically adequate execution of all project tasks, and to assure that the resulting software product is functional and reliable.

Random Walk Method
- a method for solving the governing solute transport equation by tracking a large number of particles proportional to solute concentration, and each particle advected deterministically and dispersed probabilistically.

Reliability
- the probability that a model will satisfactorily perform its intended function under given circumstances; it is the amount of credence placed in the results of model application.

Residual
- the difference between the computed and observed values of a variable at a specific time and location.

Round-Off Error

- modeling error due to computer induced differences in the result between an exact calculation and a computer-based calculation due to limitations in the representation of numbers and functions in a computer and restrictions on accuracy programmed in the software.

Saturated Zone
- see zone of saturation

Saturated Zone Flow Model
- see ground-water model.

Seepage Face
- a physical boundary segment of a ground-water system along which ground-water discharges and which is present when a phreatic surface ends at the downstream external boundary of a flow domain; along this boundary segment, of which the location of the upper end is a-priori unknown, water pressure equals atmospheric pressure and hydraulic head equals elevation head.

Semi-Analytical Model
- a mathematical model in which complex analytical solutions are evaluated using approximative techniques, resulting in a solution discrete in either the space or time domain.

Sensitivity
- the variation in the value of one or more output variables (such as hydraulic heads) or quantities calculated from the output variables (such as ground-water flow rates) due to changes in the value of one or more inputs to a ground-water flow model (such as hydraulic properties or boundary conditions).

Sensitivity Analysis
- a quantitative evaluation of the impact of variability or uncertainty in model inputs on the degree of calibration of a model and on its results or conclusions.

  Discussion - Andersen and Woessner use “calibration sensitivity analysis” for assessing the effect of uncertainty on the calibrated model and “prediction sensitivity analysis” for assessing the effect of uncertainty on the prediction. The definition of sensitivity analysis for the purpose of this guide combines these concepts, because only by simultaneously evaluating the effects on the model’s calibration and predictions can any particular level of sensitivity be considered significant or insignificant.

- a procedure based on systematic variation of model input values 1) to identify those model input elements that cause the most significant variations in model output; and 2) to quantitatively evaluate the impact of uncertainty in model input on the degree of calibration and on the model's predictive capability.

Simulation
- one complete execution of a ground-water modeling computer program, including input and output.

  Discussion- for the purposes of this guide a simulation refers to an individual modeling run. However, simulation is sometimes also used broadly to refer to the process of modeling in general.

Simulation Log
- a log used to document (in terms of input data, code used, simulation purpose and results) of individual model simulations.

Sink
- a process whereby, or a feature from which, water, vapor, NAPL, solute or heat is extracted from the ground-water or vadose zone flow system.

Soil Gas
- vadose zone atmosphere.

Soil-Water Pressure
- the pressure of the water in a soil-water system, as measured by a piezometer for a saturated soil, or by a tensiometer for an unsaturated soil.

Solute Transport Model
- application of a model to represent the movement of chemical species dissolved in ground water.

Source
- a process whereby, or a feature from which, water, vapor, NAPL, solute or heat is added to the ground-water or vadose zone flow system.

Source of Contaminants
- the physical location (and spatial extent) of the source contaminating the aquifer; in order to model fate and transport of a contaminant, the characteristics of the contaminant source must be known or assumed.

Source Loading
- the rate at which a contaminant is entering the ground-water system at a specific source.

Source Strength
- see source loading

Specific Capacity
- the rate of discharge from a well divided by the drawdown of the water level within the well at a specific time since pumping started.

Specific Storage
- the volume of water released from or taken into storage per unit volume of the porous medium per unit change in head.

Specific Yield
- the ratio of the volume of water that the saturated rock or soil will yield by gravity to the volume of the rock or soil. In the field, specific yield is generally determined by tests of unconfined aquifers and represents the change that occurs in the volume of water in storage per unit area of unconfined aquifer as the result of a unit change in head. Such a change in storage is produced by draining or filling of pore space and is, therefore, mainly dependent on particle size, rate of change of the water table, and time of drainage.

Specified Flux
- boundary condition of the second kind; also called fixed or prescribed flux.

Specified Head, Concentration, Temperature
- boundary condition of the first kind; also called fixed or prescribed head, concentration or temperature.

Steady-State Flow

- a characteristic of a ground-water or vadose zone flow system where the magnitude and direction of specific discharge at any point in space are constant in time.

Stochastic

- consideration of subsurface media and fluid parameters as random variables.

  Discussion: A stochastic or random variable is a variable quantity with a definite range of values, each one of which, depending on chance, can be obtained with a definite probability.

Stochastic Model

- a model which incorporates stochastic description of the modeled system and/or processes to quantitatively establish the extent to which uncertainty in model input translates to uncertainty in model predictions.

  Discussion: A stochastic or random variable is a variable quantity with a definite range of values, each one of which, depending on chance, can be obtained with a definite probability.

Stochastic Process

- a process in which the dependent variable is random (so that prediction of its value depends on a set of underlying probabilities) and the outcome at any instant is not known with certainty.

  Discussion: A stochastic or random variable is a variable quantity with a definite range of values, each one of which, depending on chance, can be obtained with a definite probability.

Storage Coefficient

- the volume of water an aquifer releases from or takes into storage per unit surface area of the aquifer per unit change in head. For a confined aquifer, the storage coefficient is equal to the product of the specific storage and aquifer thickness. For an unconfined aquifer, the storage coefficient is approximately equal to specific yield.

Superposition Principle

- the addition or subtraction of two or more different solutions of a governing linear partial differential equation (PDE) to obtain a composite solution of the PDE.

Transient Flow

- a condition that occurs when at any location in a ground-water or vadose zone flow system the magnitude and/or direction of the specific discharge changes with time.

Transmissivity

- the volume of water at the existing kinematic viscosity that will move in a unit time under a unit hydraulic gradient through a unit width of the aquifer.

  Discussion - it is equal to an integration of the hydraulic conductivities across the saturated part of the aquifer perpendicular to the flow paths.

Uncertainty Analysis

- the quantification of uncertainty in the spatially distributed values of input properties of a ground-water or vadose zone flow or transport model, and its propagation into model results. [1, modified].

Unsaturated Zone

- see vadose zone

Unsaturated Zone Flow Model
- see vadose zone flow model.

Unsteady flow
- see transient flow.

Vadose Zone
- the hydrogeological region extending from the soil surface to the top of the principle water table; commonly referred to as the “unsaturated zone” or “zone of aeration”. These alternate names are inadequate as they do not take into account locally saturated regions above the principle water table (for example, perched water zones).

Vadose Zone Flow Model
- a non-unique, simplified, mathematical description of the flow of liquids, vapor or air in the subsurface zone above the water-table, coded in a computer programming language, together with a quantification of the simulated system in the form of boundary conditions, system and process parameters, and system stresses.

Vadose Zone Flow System
- an aggregate of rock, in which both water and air enters and moves, and which is bounded by rock that does not allow any water movement, and by zones of interaction with the earth's surface, atmosphere, and surface water systems. A vadose zone flow system has two basic hydraulic functions: it is a reservoir for water storage, and it serves as a conduit by facilitating the transmission of water from intake to outtake areas, integrating various inputs and dampening and delaying the propagation of responses to those inputs. A vadose zone flow system may transport dissolved chemical constituents and heat.

Validation
- see model validation and code validation.

Verification
- see model verification or code verification

Water Table (Ground-Water Table)
- the surface of a ground-water body at which the water pressure equals atmospheric pressure; earth material below the ground-water table is saturated with water.

Zone of Saturation
- a hydrologic zone in which all the interstices between particles of geologic material or all of the joints, fractures, or solution channels in a consolidated rock unit are filled with water under pressure greater than that of the atmosphere.

APPENDIX A .

FUNCTIONALITY DESCRIPTORS AND STANDARD TERMS

(After van der Heijde, 1994)

Tables A.1. Functionality Descriptors

Table A.1.1. General Software Information

Type of information	Comments
Software identification number	unique number, for example IGWMC data base key number
Software name	acronym; full name in brackets; if no name known, provide short description
Description date	Date when description was prepared
Functionality description analyst	name of person who prepared this description
Date of first release of software	
Version number of latest (current) release	
Date of latest release	official software release date by custodian, latest date or documentation, latest date stamp on program files
Name of authors of code	last name first followed by initials for first author (to allow sorting by last name of principal author); other authors start with initials followed by last name; no institution names in this field (see separate field)!
Development purpose/objective	see appendix A.2 for example terms
Software classification/type	see appendix A.3 for example terms
System(s) of supported units	units of measurement
Short description of model	abstract/summary; should include aspects of hydrogeology, dimensionality, transient/steady-state, flow and transport processes, boundary conditions. mathematical methods, calculated variables, user-interface, output options, etc.
Computer system requirements	list requirements per computer platform separated by commas; list different platforms separated by semi-colons; include hardware and software requirements
Program code information	language, compiler, etc.; reviewer's compilation information, if code is received in un-compiled form
Evaluation of documentation	use combination of standard terms to describe what is covered in documentation (see appendix A.4)
Evaluation of documented code testing	use standard terms to describe what kind of testing has been performed (see appendix A.5)

Table A.1.1. continued

Evaluation of level of external review	refers to description of theoretical framework, code performance and other issues in peer reviewed journals, reports or text books
Code input processing capabilities	code input preparation, data editing, type of user-interface (GUI, graphic site maps with direct spatial input and gridding options), file import capabilities (file formats)
Code output processing capabilities	form of screen output (for parameter/variable type see specific software types); file save and export options
Code operation	batch operation, operation from menu-based shell, user-interactive computational features
Code availability terms	see appendix A.6 for example terms; may be expanded upon
Availability of software support	type, level and conditions; identify source of support in terms of custodian, distributor or other parties
Development institution	name and address of institute, university/department, agency/department, company where code has been developed
Custodian institution	name and address of institute, university/department, agency/department, company where code has been developed

Table A.1.2. Hydro-/Soil-Stratigraphic System

Type of information	Comments
Model dimensionality	dimensions supported by code
Characteristics of numerical grid	fixed vs. flexible number of cell/element, size/shape of cells/elements, fixed vs. movable grids
Type of aquifers/aquifer-aquitard sequences supported	various options for hydrogeologic layering
Medium properties of saturated zone	saturated zone flow property distribution in time and space supported by model; see appendix A.7 for terms
Medium properties of unsaturated zone	unsaturated zone flow property distribution in time and space supported by model; see appendix A.8 for terms

Table A.1.3. Flow Simulation Capabilities

Type of information	Comments
Flow characteristics of saturated/unsaturated zone	*e.g.*, steady-state, transient, Darcian, turbulent, non-linear laminar
Flow processes in saturated/unsaturated zone	*e.g.*, evaporation, condensation, evapotranspiration, recharge from precipitation, induced recharge, delayed yield from storage, infiltration, plant uptake, hysteresis, capillary rise
Changing aquifer conditions	*e.g.*, soil layer/aquifer/aquitard pinch-out, storativity conversion in space/time (confined-unconfined),
Soil functions	soil characteristic function, etc.
Fluid conditions	see appendix A.9 for terms; expand if needed
Boundary/initial conditions for flow	see appendix A.10 for terms; expand if needed
Mathematical solution method(s) for flow part	analytical/approximate analytical/numerical solution; major numerical method, *e.g.*, analytic element, finite difference, integral finite difference, finite element; time discretization method; matrix solving technique(s)
Parameter identification for flow part of code	identified parameters, *e.g.*, recharge, hydraulic conductivity; identification method, *e.g.*, graphic curve matching, direct/indirect numerical method, linear/nonlinear regression, least squares
Output options for flow	*e.g.*, head/pressure, potential, drawdown, moisture content, intercell fluxes, velocities, stream function values, streamlines, pathlines, traveltimes, isochrones, interface position, capture zone delineation, position saltwater wedge, water budget components (global water balance), boundary fluxes

Table A.1.4. Solute Transport Simulation Capabilities

Type of information	Comments
Compounds model can handle	*e.g.*, any constituent, single constituent, two/more interacting constituents, TDS, heavy metals, nitrogen/phosphorus compounds, organics, radionuclides, bacteries, viruses
Transport and fate processes	*e.g.*, advection, mechanical dispersion, molecular diffusion, ion exchange, substitution, hydrolysis, dissolution, precipitation, redox reactions, acid/base reactions, complexation, radioactive decay, chain decay, first-order (bio-) chemical decay, aerobic/anaerobic biotransformation, plant solute uptake, vapor phase sorption, liquid phase sorption (linear isotherm/retardation, Langmuir/Freundlich isotherm, sorption hysteresis, non-equilibrium sorption), volatilization, condensation, (de)nitrification, nitrogen cycling, phosphorus cycling, die-off (bacteries, viruses), filtration

Table A.1.4. continued

Boundary/initial conditions for solute transport	*e.g.*, fixed concentration or specified time-varying concentration, zero solute flux, fixed or specified time-varying cross-boundary solute flux, solute flux from stream dependent on flow rate and concentration in stream, solute flux to stream dependent on flow rate and concentration in ground water, injection well with constant or specified time-varying concentration and flow rate, production well with solute flux dependent on concentration in ground water, solute flux dependent on intensity and concentration of natural recharge
Mathematical solution method(s) for solute transport part of code	coupling with fluid flow (concentration-influenced density and viscosity); analytical/approximate analytical/numerical solution; major numerical method, *e.g.*, analytic element, finite difference, integral finite difference, finite element, method of characteristics, random walk method; time discretization method; matrix solving technique(s)
Output options for solute transport	type of output, *e.g.*, concentration values, concentration in pumping wells, internal and cross-boundary solute fluxes, mass balance components (cell-by-cell, global), uncertainty in results (*i.e.*, statistical measures); form of output, *e.g.*, results in ASCII text format, spatial distribution and time series of concentration for postprocessing, direct screen display (text, graphics), and graphic vector file (HGL, DXF) or graphic bitmap/pixel/raster file (BMP, PCX, TIF); computational progress, *e.g.*, iteration progress and error, mass balance error, cpu use and memory allocation

Table A.1.5. Heat Transport Simulation Capabilities

Type of information	Comments
Heat transport processes	*e.g.*, convection, rock matrix conduction, fluid conduction, thermal dispersion, thermal diffusion (into aquifer matrix), thermal expansion of liquid, radiation, phase changes (water-steam, water-ice), evaporation, condensation, freezing/thawing
Boundary/initial conditions for heat transport	*e.g.*, fixed or specified time-varying temperature, zero heat flux, fixed or specified time-varying cross-boundary heat flux, injection well with constant or specified time-varying temperature and flow rate, production well with heat flux dependent on temperature of ground water, heat flux dependent on intensity and temperature of natural recharge
Mathematical solution method(s) for heat transport part of code	coupling with fluid flow; temperature-influenced density and viscosity; modification of hydraulic conductivity; analytical/approximate analytical/numerical solution; major numerical method, *e.g.*, analytic element, finite difference, integral finite difference, finite element, method of characteristics, random walk method; time discretization method; matrix solving technique(s)
Output options for heat transport	type of output, *e.g.*, temperature values, temperature in pumping wells, internal and cross-boundary heat fluxes, heat balance components (cell-by-cell, global), uncertainty in results (*i.e.*, statistical measures); form of output, *e.g.*, results in ASCII text format, spatial distribution and time series of temperature for postprocessing, direct screen display (text, graphics), and graphic vector file (HGL, DXF) or graphic bitmap/pixel/raster file (BMP, PCX, TIF); computational progress, *e.g.*, iteration progress and error, heat/energy balance error, cpu use and memory allocation

Table A.1.6. Capabilities with Respect to Simulation of Rock Matrix Deformation

Type of information	Comments
Deformation cause	*e.g.*, fluid withdrawal (increased internal rock stresses), overburden increase (increased system loading), man-made cavities and karst cave-in (reduced rock stresses)
Deformation model components	*e.g.*, displacements in aquifer, aquitard and/or overburden
Type of deformation model	*e.g.*, empirical relationship, depth/porosity model, aquitard drainage model, mechanistic model (process-based model)
Deformation processes	*e.g.*, subsidence (vertical movement of land surface), compaction/consolidation (vertical deformation, decrease of layer thickness), 2D/3D matrix deformation, matrix expansion (due to releases of skeletal stresses), coupling with fluid flow, parameter re-estimation (calculating effects of deformation on hydraulic conductivity and storage coefficient), elastic/plastic deformation; stress-dependent hydraulic conductivity compressibility of rock matrix
Boundary/initial conditions deformation	*e.g.*, prescribed constant or time-varying displacement, prescribed pore pressure, prescribed skeletal stress
Mathematical solution method(s) for deformation	analytical/approximate analytical/numerical solution; major numerical method, *e.g.*, finite difference, integral finite difference, finite element, method of characteristics; time discretization method; matrix solving technique(s)
Output options for deformation	type of output, *e.g.*, matrix displacements (internal skeletal displacements; 1D, 2D, 3D), surface displacements (subsidence; 1D), pore pressure, skeletal stress/strain, calculated parameters;, uncertainty in results (*i.e.*, statistical measures); form of output, *e.g.*, results in ASCII text format, spatial distribution and time series of displacements, pore pressure or stress/strain for postprocessing, direct screen display (text, graphics), and graphic vector file (HGL, DXF) or graphic bitmap/pixel/raster file (BMP, PCX, TIF); computational progress, *e.g.*, iteration progress and error, cpu use and memory allocation

Table A.1.7. Capabilities for Optimization of Management Decisions

Type of information	Comments
Type of management model	*e.g.*, lumped parameter, distributed parameter
Objective function	*e.g.*, hydraulic objective function (heads, pumping rates), water quality objective function (concentrations, removed mass), economic objective function (cost)
Optimization constraints	*e.g.*, drawdown, pumping/injection rates, concentration at compliance point, removed mass (because of treatment/disposal)

Decision variables	*e.g.*, pumping/injection rates, cost
Mathematical solution method(s) for management model	*e.g.*, embedding method, linked simulation-optimization, response matrix method, hierarchical approach, Lagrangian multipliers, linear/quadratic/stochastic/ mixed integer/dynamic programming
Output options for management model	*e.g.*, location of wells, pumping/injection rates

A.2. Standard Model Development Purpose/Objective Terms

Term	Description
research	model has been developed as part of a research project or in support of a research project
education	model has been developed primarily for educational purposes; e.g., to demonstrate a modeling technique or modeling method
general use	model has been developed or can be used for general applications; natural processes are described in generalized functions, requiring user-specified data for site-specific use
site-dedicated	model has been developed for a particular site or region; process functions may be site- or region-dependent and may not be transferable to other sites or regions without modifications
policy-setting	model has been developed specifically for policy setting; may not be applicable to site-specific conditions

A.3. Standard Model Type Terms

Term	Description
saturated flow	groundwater flow in the saturated zone; including pathline, streamline, and capture zone models based on flow equations
unsaturated flow	flow of water in the unsaturated zone; single phase or in conjunction with air flow
vapor flow/transport	movement of vapor in soils and chemical interaction between vapor phase and liquid and/or solid phase
solute transport	movement and (bio-)chemical transformation of water dissolved chemicals and their chemical interaction with the soil or rock matrix
heat transport	transport of heat in (partially) saturated rock or soil

Table A.3. continued

matrix deformation	deformation of soil or aquifer rock due to removal or injection of water or changes in overburden
geochemical	chemical reactions in the fluid phase and between the fluid phase and the solid phase
management/optimization	flow or transport models which includes mathematical optimization to develop a 'best' management strategy
ground-/surface-water hydraulics	interaction between groundwater and surface water described in terms of fluid mass exchanges; hydraulics of both groundwater and surface water are described
parameter ID unsaturated flow	calculation of the parameters of the soil hydraulic functions from laboratory measurements
inverse model	numerical models for distributed flow and/or transport parameter identification in the saturated zone
aquifer test analysis	analytical or numerical models for evaluation of aquifer flow parameters from pumping tests
tracer test analysis	analytical or numerical models for evaluation of aquifer transport parameters from tracer tests
water/steam flow	heat transport models in which both the liquid and steam phases are described and phase changes supported
fresh/salt water flow	sharp interface approach with either fresh water flow only, or flow in both the fresh- and salt-water zone
multi-phase flow	flow of water, NAPL and/or air/vapor
watershed runoff	watershed surface-, stream-, and groundwater runoff
surface water runoff	stream runoff routing
sediment transport	surface sediment transport
virus transport	transport of viruses
biochemical transformation	hydrochemical or solute transport models which include specific biochemical reactions and population growth/die-off equations
pre-/postprocessing	model input preparation and output reformatting or display
stochastic simulation	including Monte Carlo analysis
geostatistics	kriging

multimedia exposure	exposure assessment models for groundwater, surface water and atmospheric pathways
expert system	groundwater-oriented advisory system
data base	groundwater application oriented data base
ranking/screening	classification; no simulation
fracture network	no primary porosity, connected fractures only; discrete network of fractures connected at network nodes
porous medium	default medium type; primary porosity only
dual porosity medium	fractured porous medium with porous blocks intersected by connected or non-connected fractures; mass exchange between fractures and porous blocks
porous medium, fractures	porous medium with individual fractures
karst	models specifically designed for karst systems (pipe flow, non-Darcian flow, etc.)
water budget	lumped parameter approach for ground water flow
heat budget	lumped parameter approach for heat flow
chemical mass balance	lumped parameter approach for solute transport
water level conversion	converting water level observations to velocities using Darcy's law

A.4. Standard Code Documentation Terms

Term	Description
concepts and theory	documentation of underlying concepts and theory
test results	documentation of code testing results
model setup	instruction in model formulation, gridding, boundary selection and input parameter estimation
input instructions	formats and order of input data; required files
example problems	detailed examples of operation of code (with input data)
flow chart	charts illustrating program operation and data flow

Table A.5. continued

code/modules description	description of program elements and their functions
code structure	description of program elements and their functions
installation/ compilation	installation of software on specific computers; compilation setup

A.5. Standard Code Testing Terms

Term	Description
functionality testing	systematic testing of functionality of the code (processes, boundary conditions, etc.; IGWMC test procedure - part 1)
code intercomparison	evaluating code's functionality by comparing against another, well-established code (IGWMC test procedure part 1, level 2)
field testing	evaluating code's applicability by evaluating its performance in a field application for which a detailed, high-quality data set is available (IGWMC test procedure part 2, level 3)
laboratory data sets	evaluating model's physical basis and its functionality by using an independent data set obtained under highly controlled circumstances (IGWMC test procedure part 1, level 3)
benchmarking (analyt. solutions)	evaluating code's functionality by comparing against known analytical solutions (benchmarks; IGWMC test procedure part 1, level 1))
post-audits	evaluating code's applicability by comparing system predictions against observed system responses (IGWMC test procedure part 2, level 3)
performance testing	evaluating a code's applicability to or suitability for specific types of problems (IGWMC test procedure part 2)

A.6. Standard Code Availability Terms

Term	Description
public domain	developed with public funds; no restrictions in use, copying, redistribution; cannot be copyrighted
restricted public domain	developed with public funds; restrictions apply with respect to copying, redistribution and use

Table A.7. continued

proprietary	developed with private funds; restrictions apply with respect to single and/or multi-party use and copying; cannot be redistributed without permission of owner
license	use only after acceptance of license agreement restricting use and copying; cannot be redistributed; restrictions on network use
copyrighted	protected by copyright laws; restrictions on use and copying; cannot be redistributed without permission
non-proprietary	status not established; not proprietary or licensed
purchase	purchase fee applies
free	can be obtained for free

A.7. Standard Terms for Saturated Zone Medium

Term	Description
porous medium	continuous macroscopic model domain; primary porosity only
fracture system	complex representation of fracture geometry; secondary porosity only
individual fractures	representing a single or a limited number of well-defined fractures
fracture network	fractures represented as system of individual flow channels connected at discrete points; secondary porosity only
EFN	equivalent fracture network; stochastic approach; replace system with secondary porosity only
EPM	equivalent porous medium; deterministic or stochastic approach; replaces system consisting of primary and secondary porosity or secondary porosity only with single porous medium system
dual porosity model	fractured saturated porous rock with mass exchange between porous blocks and fractures; flow either in fractures or fractures and matrix blocks; storage primarily in matrix blocks
isotropic	hydraulic properties do not change with variations in flow direction
anisotropic	hydraulic properties may vary with variations in flow direction
homogeneous	hydraulic properties do not vary in space
heterogeneous	hydraulic properties may vary in space

A.8. Standard Terms for Unsaturated Zone Medium

Term	Description
porous medium	continuous macroscopic model domain; primary porosity only
layered soil	varying hydraulic soil properties in vertical direction
areally variable properties	areally varying hydraulic soil properties
fractured soil	fractured, slightly consolidated soils
macropores	cracked soils with flow regimes in macropores different from that in micropores
perched water table	saturated conditions in unsaturated soil above water table; not in direct contact with saturated zone
dual porosity model	fractured unsaturated porous rock with mass exchange between porous blocks and fractures; flow either in fractures or fractures and matrix blocks; storage primarily in matrix blocks

A.9. Standard Terms for Fluid Conditions

Term	Description
single fluid - water	water flow in saturated and unsaturated zone
single fluid -air/vapor	vapor flow in soils
single fluid - NAPL	NonAqueous-Phase Liquids
air and water	dual fluid system; flow of water and air in soils
steam and water	dual fluid system; flow of water and steam in geothermal reservoirs
salt/fresh water	dual fluid system; fresh and salt water separated by sharp interface
stagnant salt water	single moving fluid; flow of fresh water only in fresh/salt water system separated by sharp interface
moving salt/fresh water	dual fluid system; flow of both fresh and salt water separated by sharp interface
water and NAPL	dual fluid system; flow of both water and NAPL in saturated or unsaturated zone

Table A.10. continued

water, vapor and NAPL	multi-fluid system; flow of water, vapor and NAPL in unsaturated zone
compressible fluid	fluid(s) are considered compressible
incompressible fluid	fluid(s) are considered incompressible
variable density	fluid density may vary in time and space (dependent on temperature, concentration)
variable viscosity	fluid viscosity may vary in time and space (dependent on temperature, concentration)

A.10. Standard Terms for Flow Boundary Conditions

Term	Description
constant head/pressures	constant in time, variable in space (fixed head)
variable head/pressures	variable in time, variable in space
constant moisture content	constant in time, variable in space (unsaturated flow)
variable moisture content	variable in time, variable in space (unsaturated flow)
constant source/sink flux	constant in time, variable in space (e.g., wells)
variable source/sink flux	variable in time, variable in space (e.g., wells)
constant recharge	recharge from surface, constant in time, variable in space (saturated zone)
variable recharge	recharge from surface; variable in time, variable in space (saturated zone)
no flow	impermeable boundary
subsurface flux	underflow
infiltration	downward flux at soil surface (unsaturated flow)
ponding	constant head at soil surface (unsaturated flow)
steady free surface	water table
movable free surface	water table; e.g., FEM for cross-sectional flow through dam
seepage face	water table intersects with soil surface; e.g., in dam face

Table A.10. continued

springs	flux depends on water table/head and elevation of spring/discharge point
induced infiltration/exfiltration	leakage from/to surface water or sourcebed aquifer

APPENDIX B.

GROUND-WATER SIMULATION CODE FUNCTIONALITY CHECKLISTS

GROUND WATER SIMULATION CODE FUNCTIONALITY CHECKLIST

MODEL NAME:
VERSION:
RELEASE DATE:

AUTHOR(S):
INSTITUTE OF DEVELOPMENT:

CONTACT ADDRESS:
PHONE:
FAX:

PROGRAM LANGUAGE:
COMPUTER PLATFORM(S):

LEGAL STATUS:
PREPROCESSING OPTIONS:

POSTPROCESSING FACILITIES:

MODEL TYPE

- ☐ single phase saturated flow
- ☐ single phase unsaturated flow
- ☐ vapor flow/transport
- ☐ solute transport
- ☐ virus transport
- ☐ heat transport
- ☐ matrix deformation
- ☐ geochemical
- ☐ optimization
- ☐ groundwater and surface water hydraulics
- ☐ parameter ID saturated flow (inverse numerical)
- ☐ parameter ID unsaturated flow (analytical/ numerical)
- ☐ parameter ID solute transport (numerical)
- ☐ aquifer test analysis
- ☐ tracer test analysis
- ☐ flow of water and steam
- ☐ fresh/salt water interface
- ☐ two-phase flow
- ☐ three-phase flow
- ☐ phase transfers
- ☐ chemical transformations
- ☐ biochemical transformations
- ☐ watershed runoff
- ☐ sediment transport
- ☐ surface water runoff
- ☐ stochastic simulation
- ☐ geostatistics
- ☐ multimedia exposure
- ☐ pre-/postprocessing
- ☐ expert system
- ☐ data base
- ☐ ranking/screening
- ☐ water budget
- ☐ heat budget
- ☐ chemical species mass balance

UNITS

- ☐ SI system
- ☐ metric units
- ☐ US customary units
- ☐ any consistent system
- ☐ user-defined

PRIMARY USE

- ☐ research
- ☐ education
- ☐ general use
- ☐ site-dedicated
- ☐ policy-setting

Parameter discretization

- ☐ lumped
  - ☐ mass balance approach
  - ☐ transfer function(s)
- ☐ distributed
- ☐ deterministic
- ☐ stochastic

Spatial orientation

saturated flow
- ☐ 1D horizontal
- ☐ 1D vertical
- ☐ 2D horizontal (areal)
- ☐ 2D vertical (cross-sectional or profile)
- ☐ 2D axi-symmetric (horizontal flow only)
- ☐ fully 3D
- ☐ quasi-3D (layered; Dupuit approx.)
- ☐ 3D cylindrical or radial (flow defined in horizontal and vertical directions)

unsaturated flow
- ☐ 1D horizontal
- ☐ 1D vertical
- ☐ 2D horizontal
- ☐ 2D vertical
- ☐ 2D axi-symmetric
- ☐ fully 3D
- ☐ 3D cylindrical or radial

Restart capability - types of updates possible

- ☐ dependent variables (e.g., head, concentration, temperature)
- ☐ fluxes
- ☐ velocities
- ☐ parameter values
- ☐ stress rates (pumping, recharge)
- ☐ boundary conditions

Discretization in space

- ☐ no discretization
- ☐ uniform grid spacing
- ☐ variable grid spacing
- ☐ movable grid (relocation of nodes during run)
- ☐ maximum number of nodes/cells/elements
  - ☐ modifiable in source code (requires compilation)
  - ☐ modifiable through input
- ☐ maximum number of nodes (standard version):
- ☐ maximum number of cells/elements (standard version):

Possible cell shapes
- ☐ 1D linear
- ☐ 1D curvilinear
- ☐ 2D triangular
- ☐ 2D curved triangular
- ☐ 2D square
- ☐ 2D rectangular
- ☐ 2D quadrilateral
- ☐ 2D curved quadrilateral
- ☐ 2D polygon
- ☐ 2D cylindrical
- ☐ 3D cubic
- ☐ 3D rectangular block
- ☐ 3D hexahedral (6 sides)
- ☐ 3D tetrahedral (4 sides)
- ☐ 3D spherical

Saturated zone

Hydrogeologic zoning

- ☐ confined
- ☐ semi-confined (leaky-confined)
- ☐ unconfined (phreatic)
- ☐ hydrodynamic approach
- ☐ hydraulic approach (Dupuit-Forcheimer assumption for horizontal flow)
- ☐ single aquifer
- ☐ single aquifer/aquitard system
- ☐ multiple aquifer/aquitard systems
- ☐ max. number of aquifers:
  - ☐ discontinuous aquifers (aquifer pinchout)
  - ☐ discontinuous aquitards (aquitard pinchout)
  - ☐ storativity conversion in space (confined-unconfined)
  - ☐ storativity conversion in time
  - ☐ aquitard storativity

Hydrogeologic medium

- ☐ porous medium
- ☐ fractured impermeable rock (fracture system, fracture network)
- ☐ discrete individual fractures
- ☐ equivalent fracture network approach
- ☐ equivalent porous medium approach
- ☐ dual porosity system (flow in fractures and optional in porous matrix, storage in porous matrix and exchange between fractures and porous matrix)
- ☐ uniform hydraulic properties (hydraulic conductivity, storativity)
- ☐ anisotropic hydraulic conductivity
- ☐ nonuniform hydraulic properties (heterogeneous)

Flow characteristics

- ☐ single fluid, water
- ☐ single fluid, vapor
- ☐ single fluid, NAPL
- ☐ air and water flow
- ☐ water and steam flow
- ☐ moving fresh water and stagnant salt water
- ☐ moving fresh water and salt water
- ☐ water and NAPL
- ☐ water, vapor and NAPL
- ☐ incompressible fluid
- ☐ compressible fluid
- ☐ variable density
- ☐ variable viscosity
- ☐ linear laminar flow (Darcian flow)
- ☐ non-Darcian flow
- ☐ steady-state flow
- ☐ transient (non-steady state) flow
- ☐ dewatering (desaturation of cells)
- ☐ dewatering (variable transmissivity)
- ☐ rewatering (resaturation of dry cells)
- ☐ delayed yield from storage

Boundary conditions

- ☐ infinite domain
- ☐ semi-infinite domain
- ☐ regular bounded domain
- ☐ irregular bounded domain
- ☐ fixed head
- ☐ prescribed time-varying head
- ☐ zero flow (impermeable barrier)
- ☐ fixed cross-boundary flux
- ☐ prescribed time-varying cross-boundary flux
- ☐ areal recharge:
  - ☐ constant in space
  - ☐ variable in space
  - ☐ constant in time
  - ☐ variable in time

Boundary conditions - continued

- ☐ induced recharge from or discharge to a source bed aquifer or a stream in direct contact with ground water
  - ☐ surface water stage constant in time
  - ☐ surface water stage variable in time
  - ☐ stream penetrating more than one aquifer
- ☐ induced recharge from a stream not in direct contact with groundwater
- ☐ evapotranspiration dependent on distance surface to water table
- ☐ drains (gaining only)
- ☐ free surface
- ☐ seepage face
- ☐ springs

Sources/Sinks

- ☐ point sources/sinks (recharging/pumping wells)
  - ☐ constant flow rate
  - ☐ variable flow rate
  - ☐ head-specified
  - ☐ partially penetrating
  - ☐ well loss
  - ☐ block-to-radius correction
  - ☐ well-bore storage
  - ☐ multi-layer well
- ☐ line source/sinks (internal drains)
  - ☐ constant flow rate
  - ☐ variable flow rate
  - ☐ head-specified
- ☐ collector well (horizontal, radially extending screens)
- ☐ mine shafts (vertical)
  - ☐ water-filled
  - ☐ partially filled
- ☐ mine drifts, tunnel (horizontal)
  - ☐ water-filled
  - ☐ partially filled

Unsaturated Zone

Soil medium
- ☐ porous medium
- ☐ fractured impermeable rock
- ☐ discrete individual fractures
- ☐ dual porosity system
- ☐ equivalent fracture network approach
- ☐ equivalent porous medium approach
- ☐ micropore/macropore system
- ☐ uniform hydraulic properties
- ☐ nonuniform hydraulic properties
- ☐ anisotropic hydraulic properties
- ☐ areal homogeneous (single soil type)
- ☐ areal heterogeneous (multi soil types)
- ☐ swelling/shrinking soil matrix
- ☐ dipping soil layers
- ☐ number of soil layers:

Flow characteristics

- ☐ single fluid, water
- ☐ single fluid, vapor
- ☐ single fluid, NAPL
- ☐ air and water flow
- ☐ water and NAPL
- ☐ water, vapor and NAPL
- ☐ variable density
- ☐ variable viscosity
- ☐ linear laminar flow (Darcian flow)
- ☐ non-Darcian flow
- ☐ steady-state flow
- ☐ transient (non-steady state) flow

Parameter representation

Parameter definition
- ☐ suction vs.saturation (see next section)
- ☐ porosity
- ☐ residual saturation
- ☐ hydraulic conductivity vs.saturation (see next section)
- ☐ number of soil materials:

Soil moisture saturation - matric potential relationship (NRC 1990)
- ☐ Brutsaert (1966)
- ☐ van Genuchten (1980)
- ☐ Haverkamp et al. (1977)
- ☐ tabular

Soil hydraulic conductivity-saturation/hydraulic potential relationship (NRC 1990)
- ☐ Wind (1955)
- ☐ Brooks and Corey (1966)
- ☐ van Genuchten (1980)
- ☐ Gardner (1958)
- ☐ Haverkamp et al. (1977)
- ☐ Averjanov (1950)
- ☐ Rijtema (1965)
- ☐ tabular

Intercell conductance representation (K_r-determination)
- ☐ arithmetic
- ☐ harmonic
- ☐ geometric

Boundary conditions

- ☐ fixed head
- ☐ prescribed time-varying head
- ☐ fixed moisture content
- ☐ prescribed time-varying moisture content
- ☐ zero flow (impermeable barrier)
- ☐ fixed boundary flux
- ☐ prescribed time-varying boundary flux
- ☐ areal recharge: ☐ constant in space
  - ☐ variable in space
  - ☐ constant in time
  - ☐ variable in time
- ☐ ponding
- ☐ automatic conversion between prescribed head and flux condition

Flow related processes

- ☐ evaporation
- ☐ evapotranspiration
- ☐ plant uptake of water (transpiration)
- ☐ capillary rise
- ☐ hysteresis
- ☐ interflow
- ☐ perched water

Dependent variable(s)

- ☐ head
- ☐ drawdown
- ☐ pressure
- ☐ suction
- ☐ potential
- ☐ moisture content
- ☐ stream function
- ☐ velocity

Solution methods - Flow

- ☐ analytical
  - ☐ single solution
  - ☐ superposition
  - ☐ method of images

- ☐ analytic element method
  - ☐ point sources/sinks
  - ☐ line sinks
  - ☐ ponds
  - ☐ uniform flow
  - ☐ rainfall
  - ☐ layering
  - ☐ inhomogeneities
  - ☐ doublets
  - ☐ leakage through confining beds

- ☐ Semi-analytical
  - ☐ continuous in time, discrete in space
  - ☐ continuous in space, discrete in time
  - ☐ approximate analytical solution

- ☐ Solving stochastic PDEs
  - ☐ Monte Carlo simulations
  - ☐ spectral methods
  - ☐ small perturbation expansion
  - ☐ self-consistent or renormalization technique

- ☐ Numerical

Spatial approximation
- ☐ finite difference method
  - ☐ block-centered
  - ☐ node-centered
- ☐ integrated finite difference method
- ☐ boundary elements method
- ☐ particle tracking
- ☐ pathline integration
- ☐ finite element method

Time-stepping scheme
- ☐ fully implicit
- ☐ fully explicit
- ☐ Crank-Nicholson

Matrix-solving technique
- ☐ Iterative
  - ☐ SIP
  - ☐ Gauss-Seidel (PSOR)
  - ☐ LSOR
  - ☐ SSOR
  - ☐ BSOR
  - ☐ ADIP
  - ☐ Iterative ADIP (IADI)
  - ☐ Predictor-corrector
- ☐ Direct
  - ☐ Gauss elimination
  - ☐ Cholesky decomposition
  - ☐ Frontal method
  - ☐ Doolittle
  - ☐ Thomas algorithm
- ☐ Point Jacobi

- ☐ Iterative methods for nonlinear equations
  - ☐ Picard method
  - ☐ Newton-Raphson method
  - ☐ Chord slope method

- ☐ Semi-iterative
  - ☐ conjugate-gradient

Inverse Modeling/Parameter Identification for Flow

Parameters to be identified

- ☐ hydraulic conductivity
- ☐ transmissivity
- ☐ storativity/storage coefficient
- ☐ leakance/leakage factor
- ☐ areal recharge
- ☐ cross-boundary fluxes
- ☐ boundary heads
- ☐ pumping rates
- ☐ soil parameters/coefficients
- ☐ streambed resistance

User input

- ☐ prior information on parameter(s) to be identified
- ☐ constraints on parameters to be identified
- ☐ instability conditions
- ☐ non-uniqueness criteria
- ☐ regularity conditions

Parameter identification method

- ☐ aquifer tests (based on analytical solutions)
- ☐ numerical inverse approach

Direct method (model parameters treated as dependent variable)

- ☐ energy dissipation method
- ☐ algebraic approach
- ☐ inductive method (direct integration of PDE)
- ☐ minimizing norm of error flow (flatness criterion)
- ☐ linear programming (single- or multi-objective)
- ☐ quadratic programming
- ☐ matrix inversion

- ☐ Marquardt

Indirect method (iterative improvement of parameter estimates)

- ☐ linear least-squares
- ☐ non-linear least-squares
- ☐ quasi-linearization
- ☐ linear programming
- ☐ quadratic programming
- ☐ steepest descent
- ☐ conjugate gradient
- ☐ non-linear regression (Gauss-Newton)
- ☐ Newton-Raphson
- ☐ influence coefficient
- ☐ maximum likelihood
- ☐ (co-)kriging
- ☐ gradient search
- ☐ decomposition and multi-level optimization
- ☐ graphic curve matching

Output Characteristics - Flow

Echo of input (in ASCII text format)
- ☐ grid (nodal coordinates, cell size, element connectivity
- ☐ initial heads/pressures/potentials
- ☐ initial moisture content/saturation
- ☐ soil parameters/function coefficients
- ☐ aquifer parameters
- ☐ flow boundary conditions
- ☐ flow stresses (*e.g.*, recharge, pumping)

Simulation results - form of output
- ☐ dependent variables in binary format
- ☐ complete results in ASCII text format
- ☐ spatial distribution of dependent variable for postprocessing
- ☐ time series of dependent variable for postprocessing
- ☐ direct screen display - text
- ☐ direct screen display - graphics
- ☐ direct hardcopy (printer)
- ☐ direct plot (pen-plotter)
- ☐ graphic vector file
- ☐ graphic bitmap/pixel/raster file

Simulation results - type of output
- ☐ head/pressure/potential
  - ☐ areal values (table, contours)
  - ☐ temporal series (table, x-t graphs)
- ☐ saturation/moisture content
  - ☐ areal values (table, contours)
  - ☐ temporal series (table, x-t graphs)
- ☐ head differential/drawdown
  - ☐ areal values (table, contours)
  - ☐ temporal series (table, x-t graphs)
- ☐ moisture content/saturation
  - ☐ areal values (table, contours)
  - ☐ temporal series (table, x-t graphs)

Type of output - continued
- ☐ internal (cross-cell) fluxes
  - ☐ areal values (table, vector plots)
  - ☐ temporal series (table, x-t graphs)
- ☐ infiltration fluxes
  - ☐ areal values (table, vector plots)
  - ☐ temporal series (table, x-t graphs)
- ☐ evapo(transpi)ration fluxes
  - ☐ areal values (table, vector plots)
  - ☐ temporal series (table, x-t graphs)
- ☐ cross boundary fluxes
  - ☐ areal values (table, vector plots)
  - ☐ temporal series (table, x-t graphs)
- ☐ velocities
  - ☐ areal values (table, vector plots)
  - ☐ temporal series (table, x-t graphs)
- ☐ stream function values
- ☐ streamlines/pathlines (graphics)
- ☐ capture zone delineation (graphics)
- ☐ traveltimes (table of arrival times; tics on pathlines)
- ☐ isochrones (*i.e.*, lines of equal travel times; graphics)
- ☐ position of interface (table, graphics)
- ☐ location of seepage faces
- ☐ water budget components
  - ☐ cell-by-cell
  - ☐ global (main components for total model area)
- ☐ calculated flow parameters
- ☐ uncertainty in results (*i.e.*, statistical measures)

Computational information
- ☐ iteration progress
- ☐ iteration error
- ☐ mass balance error
- ☐ cpu time use
- ☐ memory allocation

Water Quality Constituents

- ☐ any constituent(s)
- ☐ single constituent
- ☐ two interacting constituents
- ☐ multiple interacting constituents
- ☐ radionuclides
- ☐ total dissolved solids (TDS)
- ☐ anorganics - general
- ☐ anorganics - specific
  - ☐ heavy metals
  - ☐ nitrogen compounds
  - ☐ phosphorus compounds
  - ☐ sulphur compounds
- ☐ organics
- ☐ micro-organisms
  - ☐ bacteria, coliforms
  - ☐ viruses

Transport and Fate Processes

(Conservative) transport
- ☐ advection
  - ☐ steady-state
    - ☐ uniform-parallel to transport coordinate system
    - ☐ uniform-may be under an angle with transport coordinate system
    - ☐ non-uniform
  - ☐ transient
  - ☐ velocities generated within code
    - ☐ from internal flow simulation
    - ☐ from external flow simulation or measured heads
  - ☐ velocities required as input
- ☐ mechanical dispersion
  - ☐ longitudinal
  - ☐ transverse
- ☐ molecular diffusion

Phase transfers
- ☐ solid<->gas; (vapor) sorption
- ☐ solid<->liquid; sorption
  - ☐ equilibrium isotherm
    - ☐ linear (retardation)
    - ☐ Langmuir
    - ☐ Freundlich
  - ☐ non-equilibrium isotherm
  - ☐ desorption (hysteresis)
- ☐ liquid->gas; volatilization
- ☐ liquid->solids; filtration

Fate - Type of reactions:
- ☐ ion exchange
- ☐ substitution/hydrolysis
- ☐ dissolution/precipitation
- ☐ reduction/oxidation

Fate - Type of reactions - continued)
- ☐ acid/base reactions
- ☐ complexation
- ☐ biodegradation
  - ☐ aerobic
  - ☐ anaerobic

Fate - Form of reactions:
- ☐ zero order production/decay
- ☐ first order production/decay
- ☐ radioactive decay
  - ☐ single mother/daughter decay
  - ☐ chain decay
- ☐ microbial production/decay
  - ☐ aerobic biodegradation
  - ☐ anaerobic biodegradation

Parameter representation

dispersivity
- ☐ isotropic (longitudinal=transverse)
- ☐ 2D anisotropic - allows longitudinal/transverse ratio
- ☐ 3D anisotropic - allows different longitudinal/transverse and horizontal transverse/vertical transverse ratios
- ☐ homogeneous (constant in space)
- ☐ heterogeneous (variable in space)
- ☐ scale-dependent
- ☐ internal cross terms

diffusion coefficient
- ☐ homogeneous (constant in space)
- ☐ heterogeneous (variable in space)

retardation factor
- ☐ homogeneous (constant in space)
- ☐ heterogeneous (variable in space)

- ☐ Chemical processes embedded in transport equation
- ☐ Chemical processes described by equations separate from the transport

Boundary Conditions for Solute Transport

General boundary conditions
- ☐ fixed concentration (constant in time)
- ☐ specified time-varying concentration
- ☐ zero solute flux
- ☐ fixed boundary solute flux
- ☐ specified time-varying boundary solute flux
- ☐ springs with solute flux dependent on head-dependent flow rate and concentration in ground water
- ☐ solute flux from stream dependent on flow rate and concentration in stream
- ☐ solute flux to stream dependent on flow rate and concentration in ground water

Sources and sinks
- ☐ injection well with constant concentration and flow rate
- ☐ injection well with time-varying concentration and flow rate
- ☐ production well with solute flux dependent on concentration in ground water
- ☐ point sources (e.g., injection wells)
- ☐ line sources (e.g., infiltration ditches)
- ☐ horizontal areal (patch) sources (e.g., feedlots, landfills)
- ☐ vertical patch sources
- ☐ non-point (diffuse) sources
- ☐ plant solute uptake

Solution methods - Solute transport

- ☐ flow and solute transport equations are uncoupled
- ☐ flow and solute transport equations are coupled
  - ☐ through concentration-dependent density
  - ☐ through concentration-dependent viscosity

- ☐ Analytical
  - ☐ single solution
  - ☐ superposition
  - ☐ method of images

- ☐ Semi-analytical
  - ☐ continuous in time, discrete in space
  - ☐ continuous in space, discrete in time
  - ☐ approximate analytical solution

- ☐ Solving stochastic PDEs
  - ☐ Monte Carlo simulations
  - ☐ spectral methods
  - ☐ small perturbation expansion
  - ☐ self-consistent or renormalization technique

- ☐ Numerical

Spatial approximation
- ☐ finite difference
  - ☐ block-centered
  - ☐ node-centered
- ☐ integrated finite difference
- ☐ particle-tracking
- ☐ method of characteristics
- ☐ random walk
- ☐ boundary element method
- ☐ finite element method

Time-stepping scheme
- ☐ fully implicit
- ☐ fully explicit
- ☐ Crank-Nicholson

Matrix-solving technique
- ☐ Iterative
  - ☐ SIP
  - ☐ Gauss-Seidel (PSOR)
  - ☐ LSOR
  - ☐ SSOR
  - ☐ BSOR
  - ☐ ADI
  - ☐ Iterative ADIP (IADI)
- ☐ Direct
  - ☐ Gauss elimination
  - ☐ Cholesky decomposition.
  - ☐ Frontal method
  - ☐ Doolittle
  - ☐ Thomas algorithm
  - ☐ Point Jacobi

- ☐ Iterative methods for nonlinear equations
  - ☐ Picard method
  - ☐ Newton-Raphson method
  - ☐ Chord slope method

- ☐ Semi-iterative
  - ☐ conjugate-gradient

Inverse/parameter Identification for Solute Transport

Parameters to be identified
- ☐ velocity
- ☐ dispersivity
- ☐ diffusion coefficient
- ☐ retardation factor
- ☐ source strength
- ☐ initial conditions (concentrations)

User input
- ☐ prior information on parameters to be identified
- ☐ constraints on parameters to be identified
- ☐ instability conditions
- ☐ non-uniqueness criteria
- ☐ regularity conditions

Parameter identification method

- ☐ tracer tests (based on analytical solutions)
- ☐ numerical inverse approach

Direct method (model parameters treated as dependent variable)
- ☐ energy dissipitation method
- ☐ algebraic approach
- ☐ inductive method (direct integration of PDE)
- ☐ minimizing norm of error flow (flatness criterion)
- ☐ linear programming (single- or multi-objective)
- ☐ quadratic programming
- ☐ matrix inversion

Indirect method (iterative improvement of parameter estimates)
- ☐ linear least-squares
- ☐ nonlinear least-squares
- ☐ quasi-linearization
- ☐ linear programming
- ☐ quadratic programming
- ☐ steepest descent
- ☐ conjugate gradient
- ☐ nonlinear regression (Gauss-Newton)
- ☐ Newton-Raphson
- ☐ maximum likelihood
- ☐ (co-)kriging

Output Characteristics - Solute Transport

Echo of input (in ASCII text format)
- ☐ grid (nodal coordinates, cell size, element connectivity)
- ☐ initial concentrations
- ☐ transport parameter values
- ☐ transport boundary conditions
- ☐ transport stresses (source/sink fluxes)

Simulation results - Type of output
- ☐ concentration values
- ☐ concentration in pumping wells
- ☐ internal and cross-boundary solute fluxes
- ☐ velocities (from given heads)
  - ☐ areal values (table, vector plots)
  - ☐ temporal series (table, x-t graphs)
- ☐ mass balance components
  - ☐ cell-by-cell
  - ☐ global (total model area)
- ☐ calculated transport parameters
- ☐ uncertainty in results (*i.e.*, statistical measures)

Simulation results - Form of output
- ☐ binary files of concentrations
- ☐ complete results in ASCII text format
- ☐ spatial distribution of concentration for postprocessing
- ☐ time series of concentration for postprocessing
- ☐ direct screen display -text
- ☐ direct screen display - graphics
- ☐ direct hardcopy (printer)
- ☐ direct plot (pen-plotter)
- ☐ graphic vector file
- ☐ graphic bitmap/pixel/raster file

Computational progress
- ☐ iteration progress
- ☐ iteration error
- ☐ mass balance error
- ☐ cpu use
- ☐ memory allocation

Transport Processes

- ☐ convection
  - ☐ steady-state
    - ☐ uniform flow
    - ☐ non-uniform flow
  - ☐ transient
- ☐ conduction
  - ☐ through rock-matrix
  - ☐ through liquid
- ☐ thermal dispersion
- ☐ thermal diffusion between rock matrix and liquid
- ☐ radiation
- ☐ phase change
  - ☐ evaporation/condensation
    - ☐ water/vapors
    - ☐ water/steam
  - ☐ freezing/thawing
- ☐ heat exchange between phases
- ☐ internal heat generation (heat source)

Parameter representation (parameters not checked are considered homogeneous)

thermal conductivity of rock matrix
- ☐ homogeneous (constant in space)
- ☐ heterogeneous (variable in space)

thermal dispersion coefficient
- ☐ isotropic (longitudinal=transverse)
- ☐ anisotropic
- ☐ homogeneous (constant in space)
- ☐ heterogeneous (variable in space)

Boundary Conditions for Heat Transport

General boundary conditions
- ☐ fixed temperature (constant in time)
- ☐ specified time-varying temperature
- ☐ zero heat flux/temperature gradient
- ☐ fixed heat flux/temperature gradient
- ☐ specified time-varying heat flux/temperature gradient
- ☐ heat flux from stream dependent on flow rate and stream temperature
- ☐ heat flux to stream dependent on flow rate and ground-water temperature
- ☐ heat flux through overburden dependent on flow rate and recharge temperature
- ☐ heat flux through overburden dependent on temperature difference between aquifer and atmosphere

Sources and sinks
- ☐ injection well with given constant temperature and flow rate
- ☐ injection well with given time-varying temperature and flow rate
- ☐ production well with given flow rate and heat flux dependent on ground-water temperature
- ☐ point sources
- ☐ line sources
- ☐ areal sources
- ☐ non-point (diffuse) sources

Solution Methods - Heat Transport

- ☐ flow and heat transport equations are uncoupled
- ☐ flow and heat transport equations are coupled
  - ☐ through temperature-dependent density
  - ☐ through temperature-dependent viscosity

- ☐ Analytical
  - ☐ single solution
  - ☐ superposition
  - ☐ method of images
- ☐ Semi-analytical
  - ☐ continuous in time, discrete in space
  - ☐ continuous in space, discrete in time
  - ☐ approximate analytical solution

- ☐ Solving stochastic PDEs
  - ☐ Monte Carlo simulations
  - ☐ spectral methods
  - ☐ small perturbation expansion
  - ☐ self-consistent or renormalization technique

- ☐ Numerical

Spatial approximation
- ☐ finite difference
  - ☐ block-centered
  - ☐ node-centered
- ☐ integrated finite difference
- ☐ particle-tracking
- ☐ method of characteristics
- ☐ random walk
- ☐ boundary element method
- ☐ finite element method

Time-stepping scheme
- ☐ fully implicit
- ☐ fully explicit
- ☐ Crank-Nicholson

Matrix-solving technique
- ☐ Iterative
  - ☐ SIP
  - ☐ Gauss-Seidel (PSOR)
  - ☐ LSOR
  - ☐ SSOR
  - ☐ BSOR
  - ☐ ADI
  - ☐ Iterative ADIP (IADI)
- ☐ Direct
  - ☐ Gauss elimination
  - ☐ Cholesky decomposition.
  - ☐ Frontal method
  - ☐ Doolittle
  - ☐ Thomas algorithm
  - ☐ Point Jacobi

- ☐ Iterative methods for nonlinear equations
  - ☐ Picard method
  - ☐ Newton-Raphson method
  - ☐ Chord slope method

- ☐ Semi-iterative
  - ☐ conjugate-gradient

<u>Output Characteristics - Solute Transport</u>

Echo of input (in ASCII text format)
- ☐ grid (nodal coordinates, cell size, element connectivity
- ☐ initial temperatures
- ☐ transport parameter values
- ☐ transport boundary conditions
- ☐ transport stresses (source/sink fluxes)

Simulation results - Type of output
- ☐ temperature values
- ☐ temperature in pumping wells
- ☐ internal and cross-boundary heat fluxes
- ☐ velocities (from given heads)
  - ☐ areal values (table, vector plots)
  - ☐ temporal series (table, x-t graphs)
- ☐ heat balance components
  - ☐ cell-by-cell
  - ☐ global (total model area)
- ☐ calculated transport parameters
- ☐ uncertainty in results (*i.e.*, statistical measures)

Simulation results - Form of output
- ☐ binary files of temperatures
- ☐ complete results in ASCII text format
- ☐ spatial distribution of temperature for postprocessing
- ☐ time series of temperature for postprocessing
- ☐ direct screen display -text
- ☐ direct screen display - graphics
- ☐ direct hardcopy (printer)
- ☐ direct plot (pen-plotter)
- ☐ graphic vector file
- ☐ graphic bitmap/pixel/raster file

Computational progress
- ☐ iteration progress
- ☐ iteration error
- ☐ heat balance error
- ☐ cpu use
- ☐ memory allocation

Modeled System

Deformation cause
- ☐ fluid withdrawal (increased internal rock matrix stresses)
- ☐ overburden increase (increased system loading)
- ☐ man-made cavities (reduced rock-matrix stresses)

Model components
- ☐ aquifer only
- ☐ aquifer/overburden
- ☐ aquifer(s)/aquitard(s)
- ☐ aquifer(s)/aquitard(s)/overburden

Model Types

- ☐ Empirical model
  - ☐ depth/porosity model
- ☐ Semi-empirical model
  - ☐ aquitard drainage model
- ☐ Mechanistic process-based model (see processes)
  - ☐ Terzaghi (1925)
  - ☐ Biot (1941)

Processes

- ☐ one-dimensional deformation
  - ☐ subsidence (vertical movement of land surface
  - ☐ compaction (vertical deformation; decrease of thickness of sediments due to increase of effective stress; also consolidation)
  - ☐ matrix expansion (due to reduced skeletal stress)
- ☐ two-dimensional deformation
  - ☐ vertical (cross-sectional)
  - ☐ horizontal (areal)
- ☐ three-dimensional deformation
- ☐ coupling fluid flow and deformation
  - ☐ single equation
  - ☐ two coupled equations
- ☐ coupling temperature change with fluid flow and deformation (e.g., geothermal reservoirs)
- ☐ elastic deformation
- ☐ inelastic (plastic) deformation

Parameter Representation

Note that parameters not mentioned are considered homogeneous in space. (Refer to Flow System Characterization beginning on B-3.)

- ☐ stress-dependent hydraulic conductivity

compressibility of rock matrix
- ☐ homogeneous (constant in space)
- ☐ heterogeneous

coefficient of consolidation (isotropic)
- ☐ homogeneous
- ☐ heterogeneous

Boundary Conditions for Deformation

- ☐ prescribed displacement
  - ☐ constant in time
  - ☐ varying in time
- ☐ prescribed pore pressure
  - ☐ constant in time
  - ☐ varying in time
- ☐ prescribed skeletal stress
  - ☐ constant in time
  - ☐ varying in time

Solution Methods - Deformation

Flow and deformation equations are:
- ☐ uncoupled
- ☐ coupled

- ☐ Analytical
  - ☐ single solution
  - ☐ superposition

- ☐ Semi-analytical
  - ☐ continuous in time, discrete in space
  - ☐ continuous in space, discrete in time
  - ☐ approximate analytical solution

- ☐ Numerical

Spatial approximation
- ☐ finite difference
  - ☐ block-centered
  - ☐ node-centered
- ☐ integrated finite difference
- ☐ finite element method

Time-stepping scheme
- ☐ fully implicit
- ☐ fully explicit
- ☐ Crank-Nicholson

Matrix-solving technique
- ☐ Iterative
  - ☐ SIP
  - ☐ Gauss-Seidel (PSOR)
  - ☐ LSOR
  - ☐ SSOR
  - ☐ BSOR
  - ☐ ADI
  - ☐ Iterative ADIP (IADI)

- ☐ Semi-iterative
  - ☐ conjugate-gradient

- ☐ Direct
  - ☐ Gauss elimination
  - ☐ Cholesky decomposition
  - ☐ Frontal method
  - ☐ Doolittle
  - ☐ Thomas algorithm
  - ☐ Point Jacobi

- ☐ Iterative methods for nonlinear equations
  - ☐ Picard method
  - ☐ Newton-Raphson method
  - ☐ Chord slope method

Output Characteristics - Deformation

Echo of input (in ASCII text format)
- ☐ grid (nodal coordinates, cell size, element connectivity)
- ☐ initial stresses
- ☐ deformation parameter values
- ☐ deformation boundary conditions

Simulation results - Form of output
- ☐ binary files
- ☐ complete results in ASCII text format
- ☐ spatial distribution for postprocessing
- ☐ time series for postprocessing
- ☐ direct screen display -text
- ☐ direct screen display - graphics
- ☐ direct hardcopy (printer,pen-plotter)
- ☐ graphic vector file/display
- ☐ graphic bitmap/pixel/raster file

Simulation results - Type of output
- ☐ matrix displacements (internal skeletal displacements;1D, 2D, 3D)
- ☐ surface displacements (subsidence; 1D)
- ☐ pore pressure
- ☐ skeletal stress/strain
- ☐ calculated parameters

Computational progress
- ☐ iteration progress
- ☐ iteration error
- ☐ cpu use
- ☐ memory allocation

REFERENCES FOR APPENDIX B

Averjanov, S.F. 1950. About Permeability of Subsurface Sites in Case of Incomplete Saturation. Engineering Collection, Vol. VII, as quoted by P. Ya., Polibarinova-Kuchina, Theory of Groundwater Movement, 1962, Princeton University Press, Princeton, New Jersey.

Biot, M.A. 1941. General Theory of Three-Dimensional Consolidation. J. Applied Physics, Vol. 12, pp. 155-164.

Brooks, R.H., and A.T. Corey. 1966. Properties of Porous Media Affecting Fluid Flow. Journ. Irrigation and Drainage Div. ASCE, Vol. 92(IR2), pp. 61-68.

Brutseart, W. 1966. Probability Laws for Pore-Size Distributions. Soil Science Vol. 101, pp. 85-92.

Gardner, W.R. 1958. Some Steady-State Solutions to the Unsaturated Flow Equation with Application to Evaporation from a Water-Table. Soil Science Vol. 85, pp. 228-232.

Haverkamp, R., M. Vauclin, J. Bouma, P.J. Wierenga, and G. Vachaud. 1977. A Comparison of Numerical Simulation Models for One-Dimensional Infiltration. Soil Sci. Soc. of Am. Journ., Vol. 41, pp. 285-294.

National Research Council (NRC), Committee on Ground Water Modeling Assessment, Water Science and Technology Board. 1990. Ground Water Models: Scientific and Regulatory Applications. National Academy Press, Washington, D.C.

Rijtema, P.E. 1965. An Analysis of Actual Evapotranspiration. Agric. Res. Report No. 659. Centre for Agric. Public. and Docum., Wageningen, The Netherlands.

Terzaghi, K. 1925. Erdbaumechanik auf Bodenphysikalische Grundlage. Franz Deuticke, Leipzig, Germany.

van Genuchten, M.T. 1980. A Closed-Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. Soil Sci. Soc. of Am. Journ.. Vol. 44, pp. 892-898.

Wind, G.P. 1955. Flow of Water through Plant Roots. Netherlands Journ. of Agric. Sc., Vol. 3, pp. 259-264.

APPENDIX C

GENERIC FUNCTIONALITY TABLES
FOR SATURATED FLOW AND SOLUTE TRANSPORT

This appendix includes a series of generic functionality tables. These functionality tables may be modified and used to help design a functionality testing program for a typical ground-water flow and contaminant transport simulation code. Each functionality table lists the questions and issues that may be of concern for the code function being assessed, the objectives that a functionality test should address, and the type of benchmark that could be used to accomplish this. The functionality tables presented in this Appendix represent only a sample of code function issues that should be examined to fully evaluate the functionality of a ground-water simulation code. Furthermore, issues as code sensitivity for spatial and temporal discretization, choice of solver, and selection of iteration/solver parameters are not addressed. It might be necessary to explore those issues through sensitivity analysis.

Table C-1. Functionality Issues for Confined/Unconfined Conditions

Functionality Issue	*Test Objective*	*Type of Test*
In unconfined aquifers, transmissivity is dependent on the computed heads.	To determine if the code correctly represents the water table under steady-state conditions. How sensitive are the results for the difference between initial conditions and final heads, or boundary conditions? Does the number of model layers make a difference?	steady-state benchmark Level 1B
In unconfined aquifers, a rising water table might arise above the initial model layer, invading dry cells (saturation/wetting).	To determine if the code functions properly when water invades dry model cells, both under steady-state conditions (initial condition set below final water bearing model cells) and transient conditions.	steady-state, transient benchmark Level 1B
In unconfined aquifers, a falling water table might drop below the bottom of the initial (partially) water-filled cells (desaturation).	To determine if the code functions properly when water evacuates wet model cells and fully water-filled cells become partially water-filled, both under steady-state conditions (initial condition set above final water bearing model cells) and transient conditions.	steady-state, transient benchmark Level 1B
Cyclic variations of the water table position over more than one model layer require repeated desaturation and resaturation of model layers.	To determine if accuracy (in terms of heads and mass balance) is maintained over multiple desaturation and rewetting cycles, and if no stability problems occur.	transient benchmark Level 1B
For unconfined conditions transmissibility is a function of saturated thickness. Various schemes exist to treat the resulting nonlinear terms, including (damped) corrections at each iteration and/or time step.	To determine the accuracy for watertable conditions for various steady-state and transient conditions (e.g., poor initial conditions, and small hydraulic conductivity or storativity).	steady-state transient conceptual test intercomparison Level 1A
When the head in a confined layer drops below the top of that layer, conditions reverse to unconfined. This phenomenon typically occurs in areas of the model domain where discharge is significant. If the discharge diminishes or is reversed, conditions may become confined again.	To determine proper assignment of storativity and other code settings when conditions change between confined and unconfined (in both quasi and fully 3-D mode), and to determine stability under these conditions.	transient benchmark intracomparison Level 1B and 2A
Most 2-D and 3-D codes include an option to simulate ground-water flow in a quasi three-dimensional mode.	To determine if quasi three-dimensional mode works properly for unconfined and semi-confined multi-layer systems.	transient benchmark Level 1A

Table C-2. Functionality Issues for Sources and Sinks

Functionality Issue	*Test Objective*	*Type of Test*
In 3-D models, wells might be screened over a large vertical distance within an aquifer, or even in more than one aquifer, drawing water from (or injecting in) different layers at different rates.	To evaluate if a multi-layer implementation of the screened portion of a well works correctly, both within a single aquifer and within a multi-layer aquifer system.	steady-state benchmark Level 1B
In 3-D models, the screened part of a well is typically represented by one or more cells. When more cells are used and the top cell becomes empty from pumping, the discharge needs to be redistributed over the active pumping cells (and vice versa).	To evaluate if a multi-cell pumping well maintains the correct discharge rate during the growth of the cone of depression.	transient conceptual test Level 1A
If the cone of depression due to pumping nears the bottom of the lowest pumping cell instability may occur, and the representation of the physics becomes inaccurate if pumping continues.	To determine if stability problems occur during the development of a deep cone of depression, and to evaluate code options to signalize and handle local dewatering due to pumping.	transient conceptual test Level 1A
Simulating recharging and discharging wells is one of the most common features of modeling and accurate results are expected. Furthermore, some aspects of wells may be represented by other code functions, providing identical results.	To determine accuracy of the code in simulating well discharge and recharge for various conditions, including for a fully-penetrating well in an unconfined, leaky confined, and fully confined aquifer, a partially-penetrating well in such aquifers, and a multi-aquifer well (drawdowns and mass balance).	steady-state transient benchmark intra-comparison Level 1B
When a well is active in the same cell as another stress (areal recharge, ET, etc.) or boundary condition, the resulting terms are numerically joined in the code in one or other fashion to form approximative equations for the cell (or node).	To evaluate if a code correctly adds stresses on a cell-by-cell basis, especially for combinations of time stepping and stress periods.	steady-state transient conceptual test Level 1A
Sinks remove solute mass from the system. A discharging well is a sink with a prescribed flow flux. Outbound solute flux is dependent on the flow flux and the intrinsic concentration.	To evaluate if a code correctly computes outbound solute flux in a well and the concentration distribution resulting from this mass removal.	steady-state transient conceptual test (hand calculations) benchmark Level 1A, 1B
Sources introduce solute mass to the system. A recharging well is a source with a prescribed flow flux. Inbound solute flux is the product of the flow flux and a specified concentration for the injected water.	To evaluate if a code correctly computes inbound solute flux in a well and the concentration distribution resulting from this mass accumulation.	steady-state transient conceptual test (hand calculations) benchmark Level 1A, 1B

Table C-3. Functionality Issues for Areal Recharge

Functionality Issue	*Test Objective*	*Type of Test*
The code is expected to accurately simulate the effects of domain-wide and locally applied areal recharge.	To determine if the areal recharge function operates correctly and accurately on a cell-by-cell basis.	steady-state benchmark Level 1B
Many codes support both steady-state and transient simulations; some codes distinguish between stress-periods and time-stepping.	To determine if the code properly and accurately handles areal recharge under transient conditions.	transient benchmark superposition Level 1B
Many codes combine areal recharge internally with other source/sink terms. This may inadvertently lead to coding errors, especially when a distinction is made between stress periods and time-stepping.	To determine, conceptually, if the areal recharge function operates correctly in conjunction with other cell-by-cell stresses.	transient conceptual test Level 1A
Some 3-D codes allow desaturation (and sometimes resaturation) of cells. Areal recharge is supposed to be introduced in the topmost active cell.	To determine, conceptually, if areal recharge is always added to the topmost active cell.	transient conceptual test Level 1A
Some codes display stability and accuracy problems when areal recharge is large and aquifer hydraulic conductivity is small (in general, this is grid-discretization and time-stepping dependent).	To determine if numerical algorithms are adequate to handle typical real-world situations.	steady-state conceptual test sensitivity analysis Level 1A
Typically, areal recharge is attributed to the nodal equations on a cell-by-cell basis. Often, there is no distinction between the effects of a recharge well in the top active cell and the effects of areal recharge in that cell.	To determine if errors exist in either the areal recharge or the injection well function.	transient benchmark intracomparison Level 1B, 2A
Inbound solute flux due to areal recharge is computed as the product of the recharge flux and given concentration of the recharging water.	To determine accuracy of the code in simulating concentrations and model mass balance due to the solute accumulation from areal recharge for various conditions.	steady-state transient conceptual test (hand calculations) benchmark Level 1A, 1B

Table C-4. Functionality Issues for Heterogeneity and Anisotropy

Functionality Issue	*Test Objective*	*Type of Test*
Heterogeneity with respect to hydraulic parameters is a key element for selection of numerical simulation codes. Sharp contrast in parameter values for neighboring cells may cause stability problems, excessive computation time, inaccurate results, or nonconvergence. Representing aquifers and aquitards in a fully three-dimensional model requires assigning values to successive model layers which may differ many orders of magnitude.	To determine to what level the code supports heterogeneity, both in horizontal and vertical direction through sensitivity analysis for hydraulic and numerical parameters, and for spatial and temporal discretization.	steady-state transient conceptual test benchmark intercomparison Level 1A, 1B, 2B
Anisotropy in hydraulic conductivity may be present. Permeability in the vertical direction is typically less than the horizontal permeability due to macro- and meso-scale layering within the hydrogeologic units. Furthermore, anisotropy may also occur in horizontal direction, especially in cemented, unconsolidated rock and in consolidated rock. Effects of simulating strong anisotropy include instabilities; inaccuracies, especially near no-flow boundaries; and excessive computational time.	To determine to what level the code supports anisotropy, both in horizontal and vertical direction through sensitivity analysis for hydraulic and numerical parameters, for spatial and temporal discretization, and for grid orientation.	steady-state transient conceptual test benchmark intracomparison intercomparison Level 1A, 1B, 2A, 2B

Table C-5. Functionality Issues for Type I (Prescribed Head/Concentration) and Type II (Prescribed Flux) Boundary Conditions

Functionality Issue	*Test Objective*	*Type of Test*
First type boundary condition cells are cells where the head or concentration is fixed; the model should respond accordingly. Note that for outflow boundaries, the concentration is dependent on the concentration of the boundary-crossing fluid and cannot be specified as boundary condition.	To determine if the code correctly assigns first-type boundary conditions and correctly responds (heads and mass balance) to them, both in steady-state and transient simulations.	steady-state transient benchmark Level 1B
	To determine if code correctly switches between intrinsic concentration for outbound solute transport and fixed concentration for inbound transport.	transient conceptual test (hand calculations) Level 1A
Second type boundary condition flow cells are cells where the water mass flux is fixed or zero; the model should respond accordingly.	To determine if the code responds (heads and mass balance) correctly to second-type boundary conditions for flow, both in steady-state and transient simulations.	steady-state transient benchmark intracomparison Level 1B

Functionality Issue	*Test Objective*	*Type of Test*
The most common second-type boundary condition for solute transport is zero-flux. Solute transport at outflow boundaries is dependent on intrinsic concentration and is not specified as boundary condition. Solute transport at inflow boundaries is flow-flux-dependent and commonly specified as concentration. All other types of specified inbound/outbound solute fluxes are commonly taken care of by the source/sink term of the governing equation.	To determine if the code correctly responds (concentrations and mass balance) to zero-flux boundary conditions for solute transport, both in steady-state and transient simulations.	steady-state transient conceptual test benchmark Level 1A, 1B
	To determine if the code correctly computes outbound boundary mass fluxes.	steady-state transient conceptual test (hand calculations) Level 1A
	To determine if code correctly responds to inbound boundary mass fluxes.	steady-state transient conceptual test (hand calculations) Level 1A
Often modeling a field site involves irregular boundaries and inactive model areas within the model domain. Many codes allow the user to switch off inactive cells, which should not contribute to error and mass balance calculations. Because the solvers typically march through the cells in a strict order and direction, it may encounter a sequence of active and inactive cells in a single-direction sweep.	To determine if the code correctly incorporates active cells in the solution and excludes the effects of inactive cells in the results.	steady-state transient conceptual test (hand calculations) intercomparison Level 1A, 2B

Table C-6. Functionality Issues for Type III Boundary Conditions (Head-Dependent Flux)

Functionality Issue	*Test Objective*	*Type of Test*
General head boundary (GHB) can apply leakage through several idealized boundaries including aquitards (with a source bed aquifer above), stream beds, and other boundaries with an external source or sink. The flow is proportional to the difference between the external head and the head in an active model cell, and dependent on the leakance.	To determine accuracy of the code in simulating GHB discharge/recharge and head distribution for various conditions (heads and mass balance).	steady-state transient benchmark intercomparison Level 1B, 2B
Leakage from or to a stream/river boundary is a modification of the general head boundary. In addition to GHB, the stream boundary allows the head in the model to decline below the bottom of the streambed, generating a constant inbound flux.	To evaluate if the stream boundary function properly switches when water table rises above bottom of streambed or when water table declines below this level.	transient conceptual test Level 1A
	To evaluate if results are comparable with other forms of the 3rd-type boundary condition (e.g., GHB).	transient intracomparison Level 2A
	To determine accuracy of the code in simulating stream discharge increase/decrease and head distribution for various conditions (heads and mass balance).	steady-state transient benchmark intercomparison Level 1B, 2B
Drain functions allow water to flow toward a sink as long as the head in the aquifer is higher than the bottom of the drain. This function is a form of the general 3rd-type boundary condition. .	To evaluate if the drain shuts down when the head in the aquifer declines below the bottom of the drain, and as the drain is reactivated if the aquifer head rises (again) above the drain level.	transient conceptual test Level 1A
	To evaluate if results are comparable with other forms of the 3rd-type boundary condition (e.g., partially penetrating stream).	transient intracomparison Level 2A
	To determine accuracy of the code in simulating drain discharge and head distribution for various conditions (heads and mass balance).	steady-state transient benchmark intercomparison Level 1B, 2B
Evapotranspiration is considered a 3rd-type boundary condition. For details see Table C-7.	see Table C-7.	see Table C-7.
Inbound solute transport is dependent on concentration of the external source and the flux calculated with the GHB, stream boundary or drain boundary. Outbound flux is dependent on flux calculated with GHB, stream boundary or drain boundary and intrinsic concentration.	To determine accuracy of the code in simulating concentrations and model mass balance due to the solute gain from the solute mass source (inbound transport) or solute loss (outbound transport) for various conditions.	steady-state transient benchmark intercomparison Level 1B, 2B

Table C-7. Functionality Issues for Evapotranspiration

Functionality Issue	*Test Objective*	*Type of Test*
Evapotranspiration (ET) is often implemented as dependent on a water-table elevation in the soil above which ET is maximum.	To determine if this code function behaves correctly under transient conditions.	transient conceptual test Level 1A
When the water-table lies below the extinction elevation, ET should be zero.	To determine if this code function behaves correctly under transient conditions.	transient conceptual test Level 1A
The evapotranspiration flux between the maximum ET elevation and the extinction elevation follows a code-specific mathematical relationship.	To evaluate if the fluxes generated by the ET function are accurate for various water-table elevations.	steady-state conceptual test (hand calculations) Level 1A
	To determine if the effects of the ET fluxes on flow (and thus head distribution) are accurate.	steady-state, transient intracomparison intercomparison Level 2A, 2B
Many codes combine evapotranspiration fluxes internally with other source/sink terms. This may inadvertently lead to coding errors, especially when a distinction is made between stress periods and time-stepping.	To determine, conceptually, if the areal recharge function operates correctly in conjunction with other cell-by-cell stresses.	transient conceptual test Level 1A
Some 3-D codes allow desaturation (and sometimes resaturation) of cells. ET is supposed to be introduced in the topmost active cell only.	To determine, conceptually, if ET is always added to the topmost active cell.	transient conceptual test Level 1A
Outbound solute flux due to ET is computed as the product of the ET flux and the intrinsic concentration. Some codes include a multiplication factor between 0 and 1 to fine tune the amount of solute uptake by plants.	To determine accuracy of the code in simulating concentrations and model mass balance due to the solute loss from evapotranspiration for various conditions.	steady-state transient conceptual test (hand calculations) intercomparison Level 1A, 2B

Table C-8. Functionality Issues for Advective and Dispersive Solute Transport

Functionality Issue	*Test Objective*	*Type of Test*
Advection-dominated transport often creates numerical problems in the vicinity of the solute front.	To determine accuracy in terms of concentrations and mass balance, to evaluate stability and the occurrence of oscillations and numerical dispersion, and to perform sensitivity analysis with respect to transport parameter values, and spatial and temporal discretization.	steady-state uniform flow transient transport benchmark Level 1B
Accuracy of simulation of dispersive transport is dependent on grid orientation. Inclusion of cross-terms of the dispersion coefficient may improve accuracy.	To determine sensitivity of concentration distribution and mass balance for grid orientation.	steady-state uniform flow transient transport benchmark Level 1B
Accuracy of dispersive transport may be influenced by the contrast in the main directional components of the dispersivity, especially when using non-optimal grid orientation.	To determine accuracy of concentration distribution and mass balance for different ratios for the dispersivity components.	steady-state uniform flow transient transport benchmark Level 1B
Sometimes, advective-dispersive transport is negligible and molecular diffusion is prominent.	To determine accuracy in terms of concentrations and mass balance when molecular diffusion is important.	transient benchmark Level 1A

Table C-9. Functionality Issues for Solute Fate (Retardation and Decay)

Functionality Issue	*Test Objective*	*Type of Test*
Sorption is often represented as a linear or nonlinear reversible equilibrium reaction, represented by a retardation coefficient. Some codes implicitly maintain mass balance in both the dissolved and solid phases, other codes display mass balance problems under certain scenarios.	To evaluate correctness of reversible sorption function and to determine accuracy in terms of concentrations and mass balance for various sorption rates (check for reversibility).	steady-state uniform flow transient transport hand calculations (mass balance) benchmark (concentrations) Level 1A, 1B
Some codes include zero-order production or removal in the source/sink term of the governing equation.	To evaluate correctness and accuracy of this function in terms of concentrations and mass balance.	steady-state uniform flow transient transport hand calculations (mass balance) benchmark (concentrations) Level 1A, 1B

Functionality Issue	*Test Objective*	*Type of Test*
Many codes include first-order production or decay in the source/sink term of the governing equation. Some codes display instabilities or inaccuracies when half-life times are about the same order of magnitude or smaller as the time steps.	To evaluate correctness and accuracy of this function in terms of concentrations and mass balance. for both large and small values of the decay coefficient (including zero).	steady-state uniform flow transient transport hand calculations (mass balance) benchmark (concentrations) Level 1A, 1B

APPENDIX D.

COMPLETED FUNCTIONALITY CHECKLISTS FOR FTWORK VERSION 2.8

GROUND WATER MODEL FUNCTIONALITY DESCRIPTION

MODEL NAME: FTWORK
VERSION: 2.8b
RELEASE DATE: March 1993

AUTHOR(S): Faust, C.R. et al.
INSTITUTION OF DEVELOPMENT: GeoTrans, Inc. for Savannah River Lab.

CONTACT ADDRESS: GeoTrans, Inc., Sterling, VA
PHONE: 703/444-7000
FAX: 703/444-1685

PROGRAM LANGUAGE: FORTRAN 77
COMPUTER PLATFORM(S); DOS 5.0, UNIX, others

LEGAL STATUS: Public domain
PREPROCESSING OPTIONS: not included

POSTPROCESSING FACILITIES: not included; produces exportable files

---

MODEL TYPE

- ■ single phase saturated flow
- □ single phase unsaturated flow
- □ vapor flow/transport
- ■ solute transport
- □ virus transport
- □ heat transport
- □ matrix deformation
- □ geochemical
- □ optimization
- □ groundwater and surface water hydraulics
- ■ parameter ID saturated flow (inverse numerical)
- □ parameter ID unsaturated flow (analytical/ numerical)
- □ parameter ID solute transport (numerical)
- □ aquifer test analysis
- □ tracer test analysis
- □ flow of water and steam
- □ fresh/salt water interface
- □ twophase flow threephase flow
- □ phase transfers
- □ chemical transformations
- □ biochemical transformations
- □ watershed runoff
- □ sediment transport
- □ surface water runoff
- □ stochastic simulation
- □ geostatistics
- □ multimedia exposure
- □ pre-/postprocessing
- □ expert system
- □ data base
- □ ranking/screening
- □ water budget
- □ heat budget
- □ chemical species mass balance

UNITS

- □ SI system
- □ metric units
- □ US customary units
- ■ any consistent system
- □ user-defined

PRIMARY USE

- □ research
- □ education
- ■ general use
- □ site-dedicated
- □ policy-setting

Parameter discretization

- □ lumped
  - □ mass balance approach
  - □ transfer function(s)
- ■ distributed
- ■ deterministic
- □ stochastic

Spatial orientation

saturated flow
- ■ 1D horizontal
- ■ 1D vertical
- ■ 2D horizontal (areal)
- ■ 2D vertical (cross-sectional or profile)
- □ 2D axi-symmetric (horizontal flow only)
- ■ fully 3D
- ■ quasi-3D (layered; Dupuit approx.)
- □ 3D cylindrical or radial (flow defined in horizontal and vertical directions)

unsaturated flow
- □ 1D horizontal
- □ 1D vertical
- □ 2D horizontal
- □ 2D vertical
- □ 2D axi-symmetric
- □ fully 3D
- □ 3D cylindrical or radial

Restart capability - types of updates possible

- ■ dependent variables (e.g., head, concentration, temperature)
- □ fluxes
- ■ velocities
- ■ parameter values
- ■ stress rates (pumping, recharge)
- ■ boundary conditions

Discretization in space

- □ no discretization
- ■ uniform grid spacing
- ■ variable grid spacing
- □ movable grid (relocation of nodes during run)
- ■ maximum number of nodes/cells/elements
  - ■ modifiable in source code (requires compilation)
  - □ modifiable through input
- □ maximum number of nodes (standard version):
- □ maximum number of cells/elements (standard version):

Possible cell shapes
- ■ 1D linear
- □ 1D curvilinear
- □ 2D triangular
- □ 2D curved triangular
- ■ 2D square
- ■ 2D rectangular
- □ 2D quadrilateral
- □ 2D curved quadrilateral
- □ 2D polygon
- □ 2D cylindrical
- ■ 3D cubic
- ■ 3D rectangular block
- □ 3D hexahedral (6 sides)
- □ 3D tetrahedral (4 sides)
- □ 3D spherical

Saturated zone

Hydrogeologic zoning

- ■ confined
- ■ semi-confined (leaky-confined)
- ■ unconfined (phreatic)
- ■ hydrodynamic approach
- ■ hydraulic approach (Dupuit-Forcheimer assumption for horizontal flow)
- ■ single aquifer
- ■ single aquifer/aquitard system
- ■ multiple aquifer/aquitard systems
  max. number of aquifers:
- ■ discontinuous aquifers (aquifer pinchout)
- ■ discontinuous aquitards (aquitard pinchout)
- ■ storativity conversion in space (confined-unconfined)
- ■ storativity conversion in time
- ■ aquitard storativity

Hydrogeologic medium

- ■ porous medium
- □ fractured impermeable rock (fracture system, fracture network)
- □ discrete individual fractures
- □ equivalent fracture network approach
- □ equivalent porous medium approach
- □ dual porosity system (flow in fractures and optional in porous matrix, storage in porous matrix and exchange between fractures and porous matrix)
- ■ uniform hydraulic properties (hydraulic conductivity, storativity)
- ■ anisotropic hydraulic conductivity
- ■ nonuniform hydraulic properties (heterogeneous)

Flow characteristics

- ■ single fluid, water
- □ single fluid, vapor
- □ single fluid, NAPL
- □ air and water flow
- □ water and steam flow
- □ moving fresh water and stagnant salt water
- □ moving fresh water and salt water
- □ water and NAPL
- □ water, vapor and NAPL
- ■ incompressible fluid
- □ compressible fluid
- □ variable density
- □ variable viscosity
- ■ linear laminar flow (Darcian flow)
- □ non-Darcian flow
- ■ steady-state flow
- ■ transient (non-steady state) flow
- ■ dewatering (desaturation of cells)
- □ dewatering (variable transmissivity)
- □ rewatering (resaturation of dry cells)
- □ delayed yield from storage

Boundary conditions

- □ infinite domain
- □ semi-infinite domain
- ■ regular bounded domain
- ■ irregular bounded domain
- ■ fixed head
- ■ prescribed time-varying head
- ■ zero flow (impermeable barrier)
- ■ fixed cross-boundary flux
- ■ prescribed time-varying cross-boundary flux
- ■ areal recharge:
  - ■ constant in space
  - ■ variable in space
  - ■ constant in time
  - ■ variable in time

Boundary conditions - continued

- ■ induced recharge from or discharge to a source bed aquifer or a stream in direct contact with ground water
  - ■ surface water stage constant in time
  - ■ surface water stage variable in time
  - □ stream penetrating more than one aquifer
- ■ induced recharge from a stream not in direct contact with groundwater
- ■ evapotranspiration dependent on distance surface to water table
- ■ drains (gaining only)
- ■ free surface
- ■ seepage face
- ■ springs

Sources/Sinks

- ■ point sources/sinks (recharging/pumping wells)
  - ■ constant flow rate
  - ■ variable flow rate
  - ■ head-specified
  - ■ partially penetrating
  - □ well loss
  - □ block-to-radius correction
  - □ well-bore storage
  - ■ multi-layer well
- ■ line source/sinks (internal drains)
  - ■ constant flow rate
  - ■ variable flow rate
  - ■ head-specified
- □ collector well (horizontal, radially extending screens)
- □ mine shafts (vertical)
  - □ water-filled
  - □ partially filled
- □ mine drifts, tunnel (horizontal)
  - □ water-filled
  - □ partially filled

Dependent variable(s)

- ■ head
- □ drawdown
- □ pressure
- □ suction
- □ potential
- □ moisture content
- □ stream function
- □ velocity

Solution methods - Flow

- □ analytical
  - □ single solution
  - □ superposition
  - □ method of images

- □ analytic element method
  - □ point sources/sinks
  - □ line sinks
  - □ ponds
  - □ uniform flow
  - □ rainfall
  - □ layering
  - □ inhomogeneities
  - □ doublets
  - □ leakage through confining beds

- □ Semi-analytical
  - □ continuous in time, discrete in space
  - □ continuous in space, discrete in time
  - □ approximate analytical solution

- □ Solving stochastic PDEs
  - □ Monte Carlo simulations
  - □ spectral methods
  - □ small perturbation expansion
  - □ self-consistent or renormalization technique

- ■ Numerical

Spatial approximation
- ■ finite difference method
  - ■ block-centered
  - □ node-centered
- □ integrated finite difference method
- □ boundary elements method
- □ particle tracking
- □ pathline integration
- □ finite element method

Time-stepping scheme
- □ fully implicit
- □ fully explicit
- ■ Crank-Nicholson

Matrix-solving technique
- ■ Iterative
  - □ SIP
  - □ Gauss-Seidel (PSOR)
  - □ LSOR
  - ■ SSOR
  - □ BSOR
  - □ ADIP
  - □ Iterative ADIP (IADI)
  - □ Predictor-corrector
- ■ Direct
  - ■ Gauss elimination
  - □ Cholesky decomposition
  - □ Frontal method
  - ■ Doolittle
  - □ Thomas algorithm
- □ Point Jacobi

- □ Iterative methods for nonlinear equations
  - □ Picard method
  - □ Newton-Raphson method
  - □ Chord slope method

- □ Semi-iterative
  - □ conjugate-gradient

Output Characteristics - Flow

Echo of input (in ASCII text format)

- ■ grid (nodal coordinates, cell size, element connectivity
- ■ initial heads/pressures/potentials
- □ initial moisture content/saturation
- □ soil parameters/function coefficients
- ■ aquifer parameters
- ■ boundary conditions
- ■ stresses (recharge, pumping)

Simulation results - form of output

- ■ dependent variables in binary format
- ■ complete results in ASCII text format
- ■ spatial distribution of dependent variable for postprocessing
- □ time series of dependent variable for postprocessing
- □ direct screen display - text
- □ direct screen display - graphics
- □ direct hardcopy (printer)
- □ direct plot (pen-plotter)
- □ graphic vector file
- □ graphic bit map/pixel/raster file

Simulation results - type of output

- ■ head/pressure/potential
  - ■ areal values (table, contours)
  - ■ temporal series (table, x-t graphs)
- □ saturation/moisture content
  - □ areal values (table, contours)
  - □ temporal series (table, x-t graphs)
- □ head differential/drawdown
  - □ areal values (table, contours)
  - □ temporal series (table, x-t graphs)
- □ moisture content/saturation
  - □ areal values (table, contours)
  - □ temporal series (table, x-t graphs)

Type of output - continued

- ■ internal (cross-cell) fluxes
  - ■ areal values (table, vector plots)
  - □ temporal series (table, x-t graphs)
- ■ infiltration/recharge fluxes
  - ■ areal values (table, vector plots)
  - □ temporal series (table, x-t graphs)
- ■ evapo(transpi)ration fluxes
  - ■ areal values (table, vector plots)
  - □ temporal series (table, x-t graphs)
- ■ cross boundary fluxes
  - ■ areal values (table, vector plots)
  - □ temporal series (table, x-t graphs)
- ■ velocities
  - ■ areal values (table, vector plots)
  - □ temporal series (table, x-t graphs)
- □ stream function values
- □ streamlines/pathlines (graphics)
- □ traveltimes (tables)
- □ isochrones (graphics)
- □ position of interface (table, graphics)
- □ location of seepage faces
- ■ water budget components
  - ■ cell-by-cell
  - ■ global (total model area)
- ■ calculated parameters

Computational information

- ■ iteration progress
- ■ iteration error
- ■ mass balance error
- □ cpu time use
- ■ memory allocation

Parameters to be identified

- ■ hydraulic conductivity
- ☐ transmissivity
- ☐ storativity/storage coefficient
- ☐ leakance/leakage factor
- ■ areal recharge
- ☐ cross-boundary fluxes
- ☐ boundary heads
- ☐ pumping rates
- ☐ soil parameters/coefficients
- ☐ streambed resistance

User input

- ☐ prior information on parameter(s) to be identified
- ☐ constraints on parameters to be identified
- ☐ instability conditions
- ☐ non-uniqueness criteria
- ☐ regularity conditions

Parameter identification method

- ☐ aquifer tests (based on analytical solutions)
- ☐ numerical inverse approach

Direct method (model parameters treated as dependent variable)

- ☐ energy dissipitation method
- ☐ algebraic approach
- ☐ inductive method (direct integration of PDE)
- ☐ minimizing norm of error flow (flatness criterion)
- ☐ linear programming (single- or multi-objective)
- ☐ quadratic programming
- ☐ matrix inversion
- ☐ Marquardt

Indirect method (iterative improvement of parameter estimates)

- ☐ linear least-squares
- ☐ non-linear least-squares
- ☐ quasi-linearization
- ☐ linear programming
- ☐ quadratic programming
- ☐ steepest descent
- ☐ conjugate gradient
- ■ non-linear regression (Gauss-Newton)
- ☐ Newton-Raphson
- ☐ influence coefficient
- ☐ maximum likelihood
- ☐ (co-)kriging
- ☐ gradient search
- ☐ decomposition and multi-level optimization
- ☐ graphic curve matching
- ■ Marquardt algorithm

Water Quality Constituents

- ■ any constituent(s)
- ■ single constituent
- □ two interacting constituents
- □ multiple interacting constituents
- □ radionuclides
- □ total dissolved solids (TDS)

- □ inorganics - general
- □ inorganics - specific
  - □ heavy metals
  - □ other metals
  - □ nitrogen compounds
  - □ phosphorus compounds
  - □ sulphur compounds
  - □ chlorides

- □ organics - general
- □ organics - specific
  - □ aromatic
  - □ oxygenated
  - □ halogenated
- □ micro-organisms
  - □ bacteria, coliforms
  - □ viruses

Transport and Fate Processes

(Conservative) transport

- ■ advection
  - ■ steady-state
    - ■ uniform-parallel to transport coordinate system
    - ■ uniform-may be under an angle with transport coordinate system
    - ■ non-uniform
  - ■ transient
  - ■ velocities generated within code
    - ■ from internal flow simulation
    - □ from external flow simulation or measured heads
  - □ velocities required as input
- ■ dispersion
  - ■ longitudinal
  - ■ transverse
  - ■ molecular diffusion

Phase transfers

- □ solid<->gas; (vapor) sorption
- ■ solid<->liquid; sorption
  - ■ equilibrium isotherm (retardation)
    - ■ linear
    - □ Langmuir
    - ■ Freundlich
  - □ non-equilibrium isotherm
  - □ desorption (hysteresis)
- □ liquid->gas; volatilization
- □ liquid->solids; filtration

Fate - Type of reactions:

- □ ion exchange
- □ substitution/hydrolysis
- □ dissolution/precipitation
- □ reduction/oxidation
- □ acid/base reactions
- □ complexation

Fate - Type of reactions - (continued)

- □ biodegradation
  - □ aerobic
  - □ anaerobic

Fate - Form of reactions:

- ■ zero order production/decay
- ■ first order production/decay
- ■ radioactive decay
  - ■ single mother/daughter decay
  - □ chain decay
- □ microbial production//decay
  - □ Monod functions (aerobic biodegradation)
  - □ Michaelis-Menten function (anaerobic biodegradation)

Parameter representation

- dispersivity
  - ■ isotropic (longitudinal=transverse)
  - ■ 2D anisotropic - allows longitudinal/transverse ratio
  - ■ 3D anisotropic - allows different longitudinal/transverse and horizontal transverse/vertical transverse ratios
  - ■ homogeneous (constant in space)
  - ■ heterogeneous (variable in space)
  - □ scale-dependent
  - ■ internal cross terms
- diffusion coefficient
  - ■ homogeneous (constant in space)
  - ■ heterogeneous (variable in space)
- retardation factor
  - ■ homogeneous (constant in space)
  - ■ heterogeneous (variable in space)
- decay factor
  - ■ homogeneous (constant in space)
  - ■ heterogeneous (variable in space)

- ■ Chem. processes embedded in transport equation
- □ Chem. processes described by equations separate from the transport

Boundary Conditions for Solute Transport

General boundary conditions

- ■ fixed concentration (constant in time)
- □ specified time-varying concentration
- ■ zero solute flux
- ■ fixed boundary solute flux
- □ specified time-varying boundary solute flux
- □ springs with solute flux dependent on head-dependent flow rate and concentration in ground water
- □ solute flux from stream dependent on flow rate and concentration in stream
- □ solute flux to stream dependent on flow rate and concentration in ground water

Sources and sinks

- ■ injection well with constant concentration and flow rate
- ■ injection well with time-varying concentration and flow rate
- ■ production well with solute flux dependent on concentration in ground water
- ■ point sources (e.g., injection wells)
- ■ line sources (e.g., infiltration ditches or canals)
- ■ horizontal areal (patch) sources (e.g., feedlots, landfills)
- ■ vertical patch sources (e.g., infiltrated spill)
- ■ non-point (diffuse) sources
- ■ plant solute uptake

Solution methods - Solute transport

- ■ flow and solute transport equations are uncoupled
- □ flow and solute transport equations are coupled (density/viscosity)

□ Analytical

- □ single solution
- □ superposition
- □ method of images

□ Semi-analytical

- □ continuous in time, discrete in space
- □ continuous in space, discrete in time
- □ approximate analytical solution

□ Solving stochastic PDEs

- □ Monte Carlo simulations
- □ spectral methods
- □ small perturbation expansion
- □ self-consistent or renormalization technique

■ Numerical

Spatial approximation

- ■ finite difference
  - ■ block-centered
  - □ node-centered
- □ integrated finite difference
- □ particle-tracking
- □ method of characteristics
- □ random walk
- □ boundary element method
- □ finite element method

Time-stepping scheme

- □ fully implicit
- □ fully explicit
- ■ Crank-Nicholson

Matrix-solving technique

- □ Iterative
  - □ SIP
  - □ Gauss-Seidel (PSOR)
  - □ LSOR
  - ■ SSOR
  - □ BSOR
  - □ ADI
  - □ Iterative ADIP (IADI)
- □ Direct
  - □ Gauss elimination
  - □ Cholesky decomposition.
  - □ Frontal method
  - □ Doolittle
  - □ Thomas algorithm
  - □ Point Jacobi

□ Iterative methods for nonlinear equations

- □ Picard method
- □ Newton-Raphson method
- □ Chord slope method

□ Semi-iterative

- □ conjugate-gradient

Output Characteristics - Solute Transport

Echo of input (in ASCII text format)
- ■ grid (nodal coordinates, cell size, element connectivity)
- ■ initial concentrations
- ■ parameter values
- ■ boundary conditions
- ■ stresses (source fluxes)

Simulation results - Type of output
- ■ concentration values
- □ concentration in pumping wells
- □ internal and cross-boundary solute fluxes
- ■ velocities (from given heads)
  - ■ areal values (table, vector plots)
  - □ temporal series (table, x-t graphs)
- ■ mass balance components
  - ■ cell-by-cell
  - ■ global (total model area)
- □ calculated parameters

Simulation results - Form of output
- ■ binary files of concentrations
- ■ complete results in ASCII text format
- ■ spatial distribution of concentration for postprocessing
- ■ time series of concentration for postprocessing
- □ direct screen display -text
- □ direct screen display - graphics
- □ direct hardcopy (printer)
- □ direct plot (pen-plotter)
- □ graphic vector file
- □ graphic bit map/pixel/raster file

Computational progress
- ■ iteration progress
- ■ iteration error
- ■ mass balance error
- □ cpu time use
- ■ memory allocation

APPENDIX E.

CODE TESTING -- FTWORK VERSION 2.8: EVALUATION OF DOCUMENTED TESTS

APPENDIX E.

CODE TESTING -- FTWORK VERSION 2.8: EVALUATION OF DOCUMENTED TESTS

The ground-water modeling code FTWORK (version 2.8B, March 1993; Faust *et al.*, 1993), developed by GeoTrans, Inc., Sterling, Virginia, has been used in a pilot study for IGWMC's functionality analysis, performance evaluation, and applicability assessment protocol. As part of this study, IGWMC has rerun and evaluated the tests documented by the authors. The following overview summarizes the IGWMC analysis of the performed tests. The presentation of results is divided in three sections: 1) forwards flow simulation; 2) forwards solute transport simulation; and 3) inverse flow simulation. For each test, an IGWMC test number is listed as well as the names of the author-provided data files and the IGWMC-generated output files. Problem setup and test objectives are presented, as well as a summary of the control parameters used and the test results. Where possible, results have been compared with analytical solutions, programmed in MathCad 5.0 Plus for Windows (van der Heijde, 1995).

GROUND-WATER FLOW PROBLEMS

IGWMC test #:	FTW-TST-1.1
input file name:	DRAIN-WT.DAT
IGWMC output file:	DRAIN-WT.OUT
code reference:	manual, section 4.1.1, p. 59
description	steady-state flow to two parallel drains in an unconfined aquifer subjected to vertical recharge from precipitation.
tested functions:	ground-water recharge and unconfined flow option
assumptions:	horizontal flow; isotropic, homogeneous material properties; constant, uniform rate of recharge; horizontal impermeable base; fully penetrating drains
model domain:	half strip between drains (symmetry)
grid:	single slice in x-direction (21 cells in x-direction, 1 cell in y-direction); single layer (1 cell in z-direction); Δx=80 ft, Δy=100 ft, Δz=300 ft
boundary conditions:	constant head at drain for x=0 ft (h_o=164 ft); no flow boundary at x=1640 ft (default boundary condition; edge of model); by default boundaries in y-direction and lower boundary in z-direction are no-flow boundaries
initial conditions:	164 ft at all nodes
parameters:	hydraulic conductivity = 3.28 ft/day porosity = 0.2 recharge rate = 0.0328 ft/day
time-stepping:	n.a.
benchmark:	analytical solution (Bear, J., Hydraulics of Groundwater, McGraw-Hill, New York, 1979, p. 180; Huisman,L. Ground-water Recovery, MacMillan Press, London, p. 29, 1972)
IGWMC implementation:	MATHCAD 5.0 file: drainu2.mcd
test performed by:	problem set up for numerical code by code developers; code run and benchmark comparison by IGWMC
type of comparison:	graphic plot of heads (see Fig. 1.1.1); tabular listing of heads (see Table 1.1.1); statistical measures
statistics:	see Table 1.1.1
control parameters:	SSOR relaxation factor=1.63; error criterion=1.0E-5; weighting factor=1.0; tolerance for nonlinear iterations =5.0E-5
iteration performance:	# iterations=7; percent water balance error=-2.43991E-10

Table 1.1.1. Comparison of head changes with distance from a fixed head boundary for steady flow to parallel drains in an unconfined aquifer subject to vertical recharge

test 1.1	distance	benchmark	FTWORK	residual	pos. res.	neg. res.	abs. res	squared res.	relative res.
	[ft]	[ft]	[ft]	[ft]	[ft]	[ft]	[ft]		
	80	171.6	171.6	0.0	0.0	0.0	0.0	0.0	0.0
	160	178.6	178.6	0.0	0.0	0.0	0.0	0.0	0.0
	240	184.9	184.9	0.0	0.0	0.0	0.0	0.0	0.0
	320	190.7	190.7	0.0	0.0	0.0	0.0	0.0	0.0
	400	196.0	196.0	0.0	0.0	0.0	0.0	0.0	0.0
	480	200.8	200.8	0.0	0.0	0.0	0.0	0.0	0.0
	560	205.3	205.3	0.0	0.0	0.0	0.0	0.0	0.0
	640	209.3	209.3	0.0	0.0	0.0	0.0	0.0	0.0
	720	212.9	212.9	0.0	0.0	0.0	0.0	0.0	0.0
	800	216.2	216.2	0.0	0.0	0.0	0.0	0.0	0.0
	880	219.1	219.1	0.0	0.0	0.0	0.0	0.0	0.0
	960	221.7	221.7	0.0	0.0	0.0	0.0	0.0	0.0
	1040	224.0	224.0	0.0	0.0	0.0	0.0	0.0	0.0
	1120	226.0	226.0	0.0	0.0	0.0	0.0	0.0	0.0
	1200	227.7	227.7	0.0	0.0	0.0	0.0	0.0	0.0
	1280	229.1	229.1	0.0	0.0	0.0	0.0	0.0	0.0
	1360	230.2	230.2	0.0	0.0	0.0	0.0	0.0	0.0
	1440	231.1	231.1	0.0	0.0	0.0	0.0	0.0	0.0
	1520	231.6	231.6	0.0	0.0	0.0	0.0	0.0	0.0
	1600	231.9	231.9	0.0	0.0	0.0	0.0	0.0	0.0
sum =		3775.2	4238.7	0.0	0.0	0.0	0.0	0.0	0.0
total n =		18	18	18	18	18	18	18	18
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
measure =					0.000	0.000	#DIV/0!		0.000
					MPE	MNE	MER		MRE1

MAE = mean absolute error	MRE1 (mean rel.error) = ME/MP
MB = mean benchmark	MRE2 (mean rel.error) = sum(RE)/N
ME = mean error	NME = negative mean error
MNE = maximum negative error	PME = positive mean error
MP = mean prediction	RE (relative error) = residual/num.code value
MPE = maximum positive error	RMSE = root mean square error

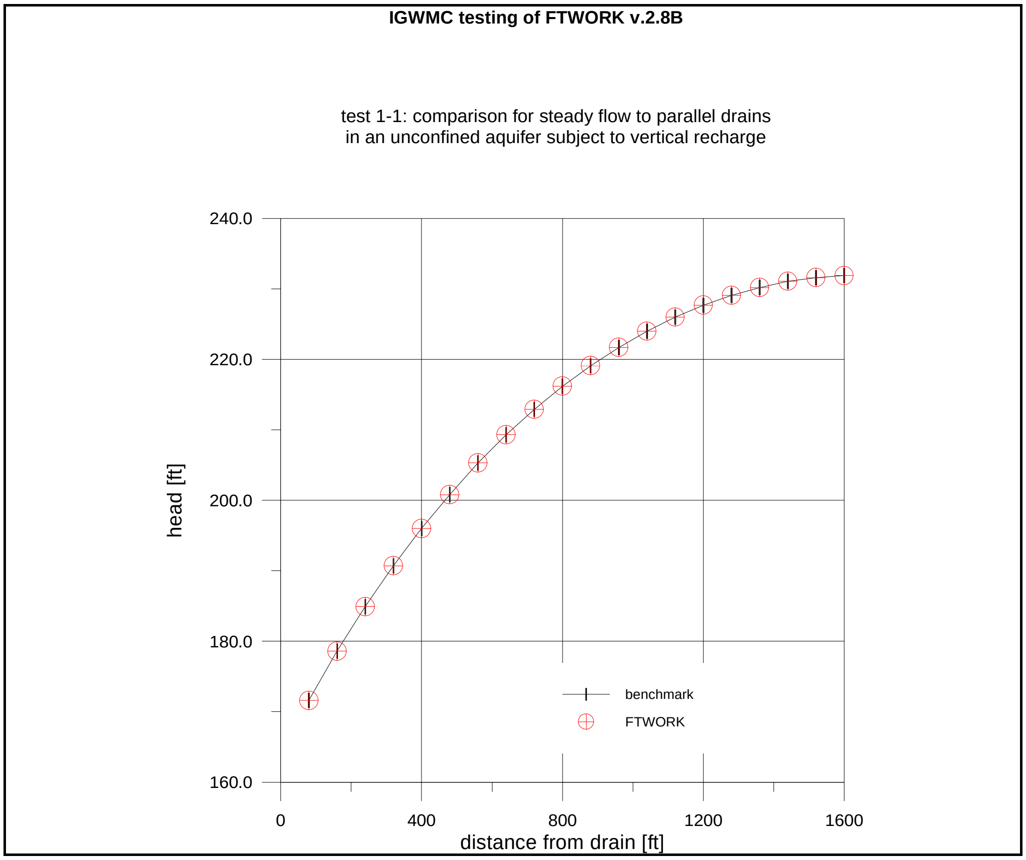
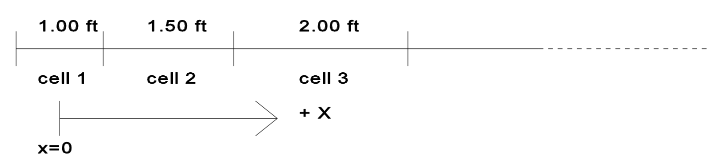


Figure 1.1.1. Comparison of heads with distance from a fixed head boundary for steady flow to parallel drains in an unconfined aquifer subject to vertical recharge.

IGWMC test #:	FTW-TST-1.2
input file name:	DRAIN-TR.DAT
IGWMC output file:	DRAIN-TR.OUT
code reference:	manual, section 4.1.2, p.61
description:	transient flow to a drain in a semi-infinite aquifer due to a step change in head
tested functions:	transient response of heads to specified head b.c. different from initial head distribution (recharge boundary)
assumptions:	horizontal flow; isotropic, homogeneous material properties, no recharge from precipitation; horizontal impermeable base; constant storage and transmissive properties (confined aquifer); instantaneous change in head in fully penetrating drain at x=0
model domain:	bounded strip replacing semi-infinite aquifer
grid:	single slice in x-direction (31 cells in x-direction, 1 cell in y-direction); single layer (1 cell in z-direction); varying grid block length in x-direction from 1ft near step-change head boundary to 300 ft at opposite boundary (see table); Δy=100 ft, Δz=200 ft. (note: center of first cell is at x=0 ft, center of second cell at x=1.25 ft, etc.)

cell spacing in x-direction (ft)							
1.00	1.50	2.00	3.00	4.00	5.00	7.00	10.00
15.00	20.00	20.00	20.00	20.00	20.00	30.00	30.00
40.00	40.00	50.00	50.00	80.00	80.00	100.0	100.0
120.0	120.0	160.0	160.0	160.0	160.0	200.0	



boundary conditions:	prescribed head at node 1 (x=0); all other boundaries are no-flow by default
initial conditions:	270 ft at node 1(at x=0); 300 ft at all other nodes.

parameters used:

parameter	benchmark	numerical code equivalent
hydraulic conductivity [ft/day]	2.19	3.28
porosity or specific storage [1/ft]	.20	.001
aquifer thickness [ft]	300.0	200.0 (to ensure that aquifer does not become unconfined)
resulting transmissivity [ft/day^2]	657.0	656.00
initial head before changeT<0 [ft]	300.0	300.0
step change [ft]	30.00	30.00
head directly after change at T=0 [ft]	270.0	270.0

time-stepping: $\Delta t_k = 1.4142\ \Delta t_{k-1}$; $\Delta t_1 = 0.01$ days; k=1....25

benchmark: analytical solution (Venetis, C., On the impulse response of an aquifer. IAHS Bulletin, v.13, p. 136, 1968); data used as given in code manual

test performed by: code developers; code rerun, output checked by IGWMC using existing test data set

type of comparison: graphic plots (fig. 1.2.1 - 1.2.4) and tabular listings (Table 1.2.1 and 1.2.2) of heads and head residuals vs. distance from head-change boundary and heads and head residuals vs. time at given location.

statistics: series 1.2a - MPE = 0.8; MNE = -0.2; ME = 0.176; MAE = 0.208; RMSE = 0.332; PME = 0.369; NME = -0.133; MER = 2.77

series 1.2b - MPE = 0.8; MNE = -0.3; ME = 0.317; MAE = 0.367; RMSE = 0.447; PME = 0.372; NME = -0.3; MER = 1.24

control parameters: SSOR relaxation factor=1.63; error criterion=1.0E-5; weighting factor=1.0; tolerance for non-linear iterations=5.0E-5

iteration performance for selected time steps (# of iterations set at 1 per time step; w.b.=water balance):

time step #	% w.b. error	cumulative % w.b. error	time step #	% w.b. error	cumulative % w.b. error
1	9.687E-13	9.687E-13	19	-2.636E-11	-7.030E-12
2	-2.479E-13	4.805E-13	20	-9.936E-13	-6.063E-12
3	1.283E-13	3.796E-13	21	2.150E-11	-1.677E-12
4	1.424E-12	6.193E-13	22	3.940E-11	4.874E-12
5	-7.748E-13	3.276E-13	23	-5.894E-11	-5.291E-12
6	-9.320E-13	8.435E-14	24	6.270E-11	5.545E-12
7	3.493E-12	7.107E-13	25	5.071E-11	1.272E-11

Table 1.2.1. Series 1.2a: comparison of heads with distance from step change boundary for t=1.52 days

distance [ft]	benchmark [ft]	code run [ft]	residual [ft]
1.25	270.3	270.3	0
3.00	270.7	270.8	0.1
5.50	271.3	271.4	0.1
9.00	272.2	272.3	0.1
13.50	273.2	273.4	0.2
19.50	274.6	275.0	0.4
28.00	276.6	277.1	0.5
40.50	279.4	280.0	0.6
58.00	283.2	283.9	0.7
78.00	286.9	287.7	0.8
98.00	290.2	290.8	0.6
118.00	292.9	293.3	0.4
138.00	295.0	295.2	0.2
163.00	296.9	297.0	0.1
193.00	298.4	298.3	-0.1
228.00	299.3	299.2	-0.1
268.00	299.8	299.6	-0.2
313.00	299.9	299.9	0
363.00	300.0	300.0	0
428.00	300.0	300.0	0
508.00	300.0	300.0	0
598.00	300.0	300.0	0
698.00	300.0	300.0	0
808.00	300.0	300.0	0
928.00	300.0	300.0	0

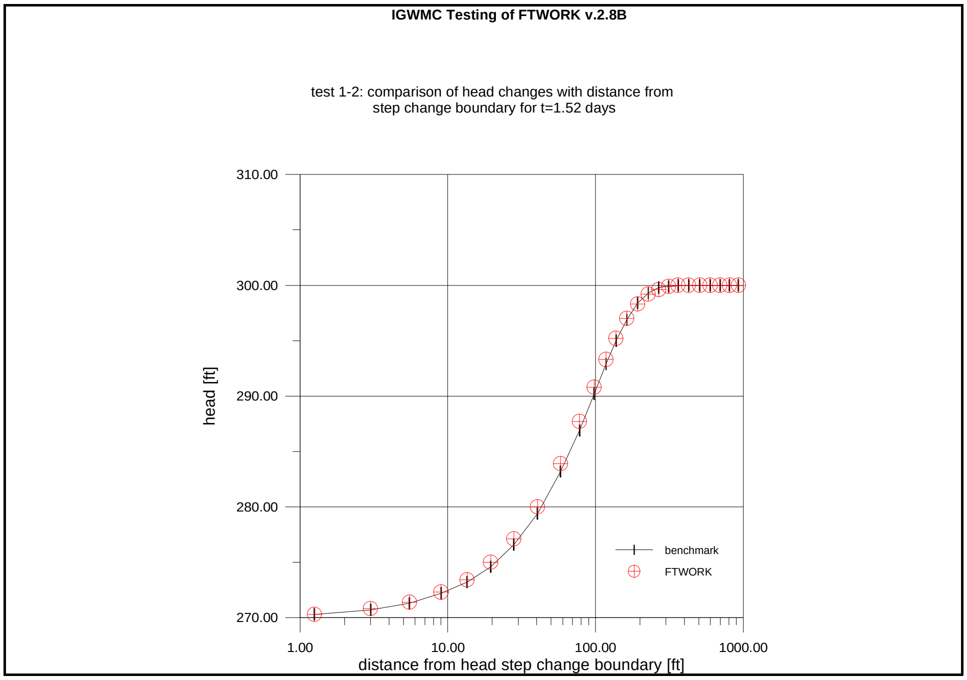


Figure 1.2.1.Series 1.2a: comparison of heads with distance from step change boundary for t=1.52d.

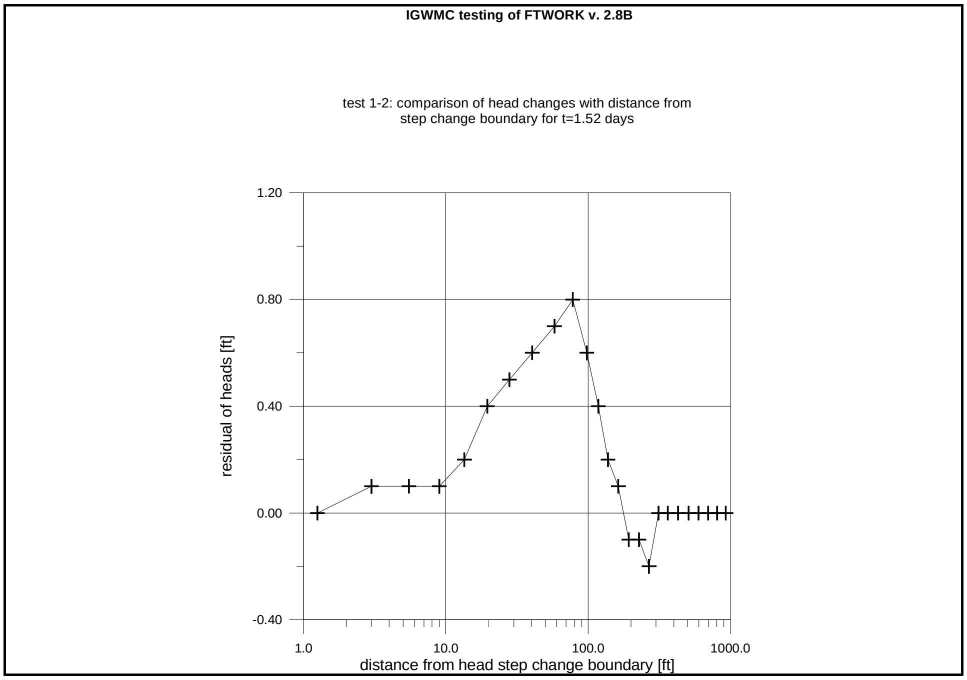


Figure 1.2.2.Series 1.2a: head residuals with distance from step change boundary for t=1.52d.

Table 1.2.2. Series 1.2b: comparison, over time, of heads in location x=28ft (node 8) due to a step change in head at t=0 days

time [ft]	benchmark [ft]	code run [ft]	residual [ft]
0.024	299.2	298.9	-0.3
0.072	294	294.5	0.5
0.169	288.0	288.8	0.8
0.362	283.0	283.8	0.8
0.748	279.3	279.9	0.6
1.520	276.6	277.1	0.5
3.070	274.7	275.0	0.3
6.160	273.3	273.5	0.2
12.300	272.4	272.5	0.1
24.700	271.7	271.8	0.1
49.400	271.2	271.3	0.1
98.800	270.8	270.9	0.1

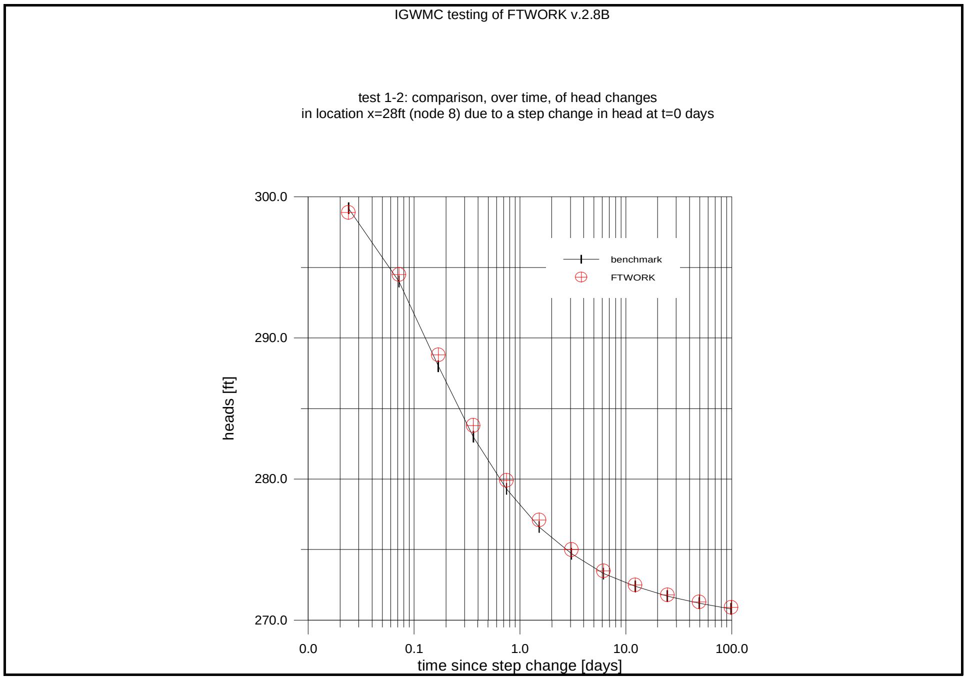


Figure 1.2.3. Series 1.2b: comparison, over time, of heads in location x=28ft (node 8) due to a step change in head at t=0 days

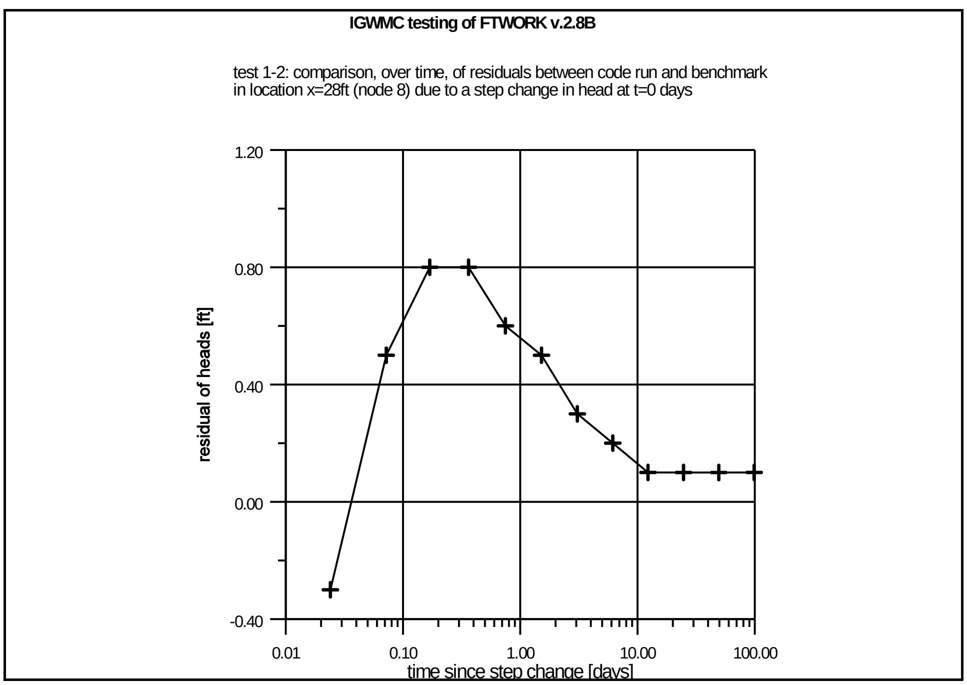


Figure 1.2.4. Series 1.2b: head residuals over time in location x=28ft (node 8) due to a step change in head at t=0 days

IGWMC test #: FTW-TST-1.3

input file name: F3.DAT

IGWMC output file: F3.OUT

code reference: manual, section 4.1.3, p.70

description: unsteady flow to a well near a straight line, fully penetrating recharge boundary in a confined aquifer

tested functions: transient response to a fixed head b.c. identical to initial head distribution (recharge boundary), and transient response to pumping a well with constant discharge

assumptions: horizontal flow; isotropic, homogeneous material properties, no recharge from precipitation; horizontal impermeable base; constant storage and transmissive properties (confined aquifer); fully penetrating well

model domain: bounded area replacing semi-infinite aquifer; for dimensions see Figure 1.3.1.

grid: rectangular single layer area of 30 cells in x-direction, 15 cells in y-direction and 1 cell in z-direction; varying grid block length in x- and y-direction ranging from 50 to 2,000 ft (see table); Δz=50ft (see Table 1.3.1.)

Table 1.3.1. Grid design for test FTW-TST-1.3

cell spacing in x-direction (ft)							
50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
50.00	50.00	50.00	50.00	70.00	100.00	150.00	200.00
300.00	500.00	700.00	1000.00	1500.00	2000.00		

cell spacing in y-direction (ft)							
50.00	50.00	50.00	50.00	50.00	70.00	100.00	150.00
200.00	300.00	500.00	700.00	1000.00	1500.00	2000.00	

boundary conditions: prescribed head at nodes where x=0 (first line of cells parallel to y-axis); all other boundaries are no-flow by default

initial conditions: 200 ft at all nodes

parameters: $Q = 0.1\ ft^3/sec$
$T = 0.001\ ft^2/sec$
$S_s = 0.00001\ ft^{-1}$

time-stepping: $\Delta t_k = 1.4142\ \Delta t_{k-1} \leq 864{,}000$ sec; $\Delta t_1 = 1{,}800$ sec; k=1....20

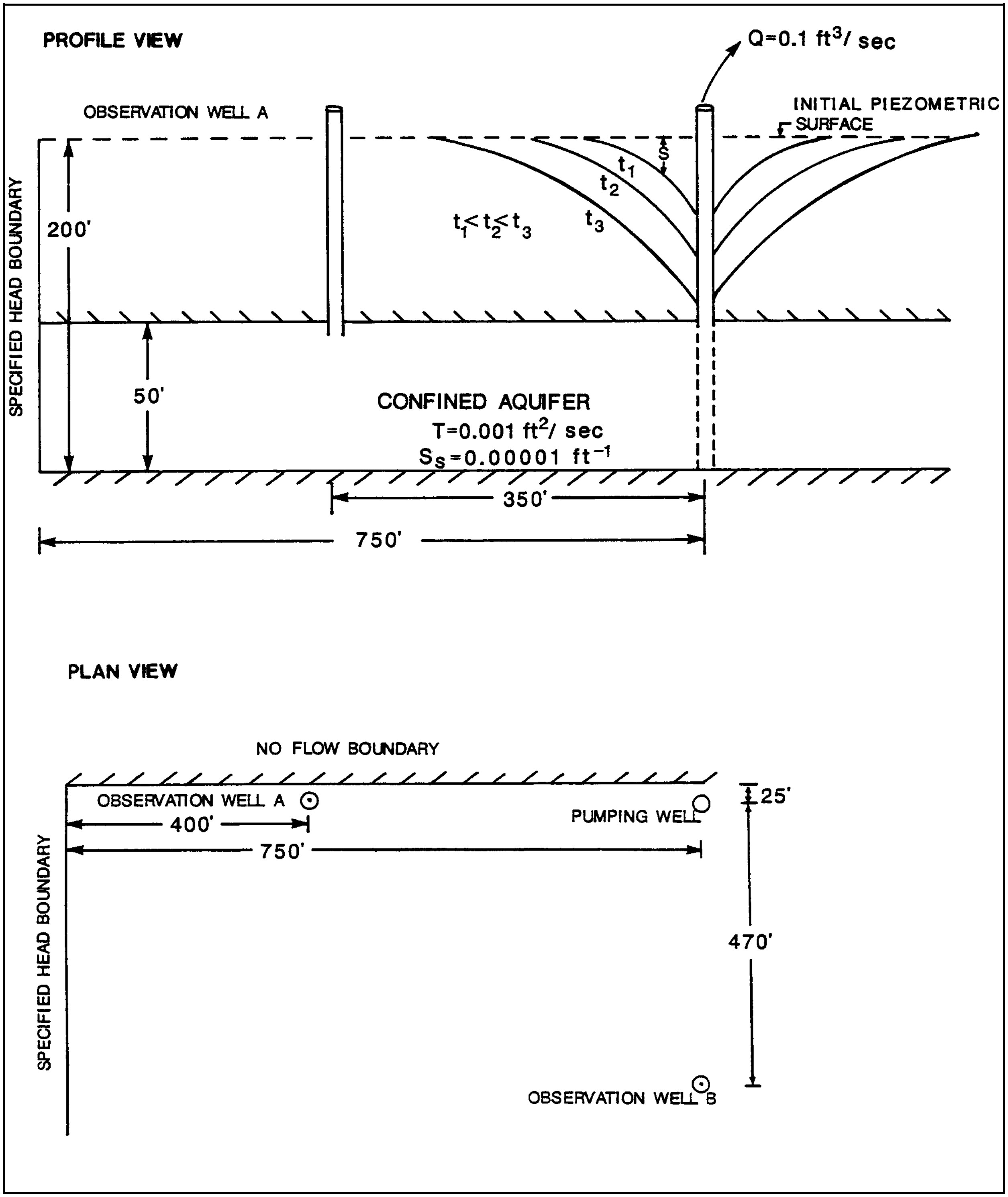


Figure 1.3.1. Schematic diagram of problem geometry for test FTW-TST-1.3 (from Faust *et al.*, 1993).

benchmark: analytical solution (Theis, 1935; superposition); data used as given in code manual

IGWMC implementation: MATHCAD 5.0 file: theis1-2.mcd

test performed by: problem setup for numerical code by code developers; code run and benchmark comparison by IGWMC using IGWMC generated benchmark data set

type of comparison: tabular listing of head (see Table 1.3.2); statistical measures

statistics: MPE = 3.4; MNE = -0.2; ME = 0.608; MAE = 0.692; RMSE = 1.158; PME = 0.975; NME = -0.167: MER = 5.838

control parameters: SSOR relaxation factor=1.63; error criterion=1.0E-4; weighting factor=1.0; tolerance for non-linear iteration=5.0E-4; max.# SSOR Iterations=30

iteration performance for various time steps

step	# iterations	% w.b.error	max.head change	step	# iterations	% w.b.error	max.head change
1	24	7.421E-4	37.4	11	30(=max)	-1.059E-2	4.94
2	25	-6.137E-4	17.4	12	30(=max)	-0.104	4.45
3	25	1.309E-3	11.0	13	30(=max)	-0.117	3.22
4	25	-4.231E-4	8.57	14	30(=max)	-0.100	2.35
5	24	-2.001E-3	7.40	15	30(=max)	-8.723E-2	1.80
6	24	1.889E-3	6.76	16	30(=max)	-7.727E-2	1.43
7	25	-7.506E-4	6.35	17	30(=max)	-6.934E-2	1.17
8	25	6.887E-4	6.04	18	30(=max)	-6.288E-2	0.98
9	25	-7.816E-4	5.73	19	30(=max)	-2.077E-2	0.71
10	25	1.029E-3	5.37				

comments: FTWORK documentation lists time maximum as 86,400 seconds instead of 864,000 seconds (p. 70 text; Fig. 4-7 and 4-8 time axis; Fig. 4-9 legend, TABLE 4.7 title)

Table 1.3.2. Comparison of head changes with distance from well or t=864,000 sec.

distance [ft]	benchmark [ft]	code run [ft]	residual [ft]
75	90.6	94.0	3.4
125	74.4	76.0	1.6
175	63.8	64.8	1.0
225	56.0	56.7	0.7
285	48.7	49.2	0.5
370	40.8	41.1	0.3
495	32.2	32.4	0.2
670	23.9	24.0	0.1
920	16.1	16.1	0.0
1320	9.0	8.8	-0.2
1920	3.9	3.7	-0.2
2770	1.2	1.1	-0.1

IGWMC test #: FTW-TST-1.4

input file name: THEIS.DAT

IGWMC output file: THEIS.OUT

code reference: manual, section 4.1.4, p.70.

description: transient response of head distribution in a non-leaky confined aquifer due to a well with a constant discharge rate

tested functions: transient response to pumping with a constant rate in a confined aquifer; serves as comparison with testing of leaky-confined conditions

assumptions: horizontal flow; isotropic, homogeneous material properties, no recharge from precipitation; horizontal impermeable base; constant storage and transmissive properties; fully penetrating well

model domain: because of symmetry considerations only one quarter of the aquifer domain is considered; bounded area replaces infinite aquifer

grid: variably spaced grid of 15 columns by 15 rows by 1 layer with grid size increasing away from well located in origin of grid; discretization in x- and y-direction identical (see table 1-4a).

boundary conditions: all boundaries are no flow by default

initial conditions: 0 ft

parameters: Q=0.4 ft^3/sec; T=0.005 ft^2/sec; S_s=0.0001ft^{-1}.

time-stepping: geometrically: $\Delta t_k = 1.4142\ \Delta t_{k-1}$; Δt_1 = 6 sec; k=1....12

benchmark: analytical solution (Theis, 1935); data used as given in code manual.

IGWMC implementation: MATHCAD 5.0 files: leaky1.mcd (compare with theis3.mcd) for distance vs. drawdown and leaky 2.mcd (compare with theis5.mcd) for time vs. drawdown (compare p. 84 of documentation).

test performed by: problem setup for numerical code by code developers; code run and benchmark comparison by IGWMC using IGWMC generated benchmark.

type of comparison: tabular listing of heads (table 1.4.1 and 1.4.2); statistical measures

statistics: series 1.4a: MPE = 0.08; MNE = -3.72; ME = -0.373; MAE =0.399; RMSE = 1.081; PME = 0.053; NME = -0.772; MER = -14.560 (n=12; n+=3; n-=6)

series 1.4b: MPE = 0.22; MNE = -0.45; ME = -0.121; MAE = 0.184; RMSE = 0.217; PME = 0.190; NME = -0.183; MER = -1.038 (n=12; n+=2; n-=10)

control parameters: SSOR relaxation factor=1.90; error criterion=1.0E-3; weighting factor=1.0; tolerance for nonlinear iteration=1.0E-3; max. # SSOR iterations=60

iteration performance: most time steps needed maximum # of iterations; water balance accuracy comparable with FTW-TST-1.3.

Table 1.4.1: Series 1.4a: comparison of head changes with distance from well for t=217 sec.

distance [ft]	benchmark [ft]	code run [ft]	residual [ft]
5.0	43.82	40.10	-3.72
17.5	27.91	27.65	-0.26
35.0	19.22	19.11	-0.11
60.0	12.69	12.50	-0.19
97.5	7.34	7.12	-0.22
152.5	3.30	3.23	-0.13
235.0	0.90	0.97	0.07
360.0	0.08	0.16	0.08
535.0	0	0.01	0.01
650.0	0	0	0
890.0	0	0	0
1440.0	0	0	0

Table 1.4.2: Series 1.4b: comparison of head changes with time at distance from well r=60ft.

time [sec]	benchmark [ft]	code run [ft]	residual [ft]
6.0	0.08	0.30	0.22
14.5	0.94	1.10	0.16
26.5	2.48	2.42	-0.06
43.5	4.33	4.13	-0.20
67.5	6.34	6.07	-0.26
101.0	8.46	8.16	-0.30
149.0	10.76	10.31	-0.45
217.0	12.69	12.50	-0.19
313.0	14.87	14.71	-0.16
449.0	17.06	16.94	-0.12
641.0	19.25	19.18	-0.07

912.0	21.44	21.42	-0.02

IGWMC test #: FTW-TST-1.5

input file name: LEAKY.DAT

IGWMC output file: LEAKY.OUT

code reference: manual, section 4.1.4, p.70

description: transient response of head distribution in a leaky confined aquifer due to a well with a constant discharge rate

tested functions: transient response to pumping with a constant rate in a leaky confined aquifer

assumptions: horizontal flow; isotropic, homogeneous material properties, no recharge from precipitation; horizontal impermeable base; constant storage, transmissive and leakage properties; fully penetrating well

model domain: because of symmetry considerations only one quarter of the aquifer domain is considered; bounded area replaces infinite aquifer

grid: identical to test FTW-TST-1.4

boundary conditions: identical to test FTW-TST-1.4

initial conditions: 0 ft

parameters: Q=0.4 ft^3/sec; T=0.005 ft^2/sec; S_s=0.0001ft^{-1}; K`/b`=1*$10^{-6}$$sec^{-1}$.

time-stepping: identical to test FTW-TST-1.4

benchmark: analytical solution (Hantush and Jacob, 1955); comparison with test 1.4 (Theis)

IGWMC implementation: MATHCAD files: leaky3.mcd for distance vs. drawdown and leaky4.mcd for time vs. drawdown (compare p. 84 of documentation); series approximation in leaky1.mcd and leaky2.mcd is less accurate

test performed by: problem setup for numerical code by code developers; code run and benchmark comparison by IGWMC using IGWMC generated benchmark

type of comparison: tabular listing of heads (table 1.5.1 and 1.5.2); statistical measures

statistics: series 1.5a: MPE = 0.06; MNE = -3.49; ME = -.315; MAE = 0.325; RMSE = 1.008; PME = 0.060; NME = -0.640; MER = -10.667 (n=12; n+=1; n-=6)

statistics: series 1.5b: MPE = 0.19; MNE = -0.28; ME = -0.028; MAE = 0.154; RMSE = 0.170; PME = 0.127; NME = -0.167; MER = -1.315 (n=12; n+=6; n-=6)

control parameters: SSOR relaxation factor=1.80; error criterion=1.0E-3; weighting factor=1.0; tolerance for nonlinear iteration=1.0E-3; max. # SSOR iterations=31

# of iterations for each time step: 31, 31, 31, 31, 31, 31, 28, 27, 24, 21, 17, 10, 6,1,1, 1, 1, 1, 1, 1

water balance accuracy: in range 2.0E-2 -- 5.0E-3 percent

Table 1.5.1. Series 1.5a: comparison of head changes with distance from well for t=217 sec.

distance [ft]	benchmark [ft]	code run [ft]	residual [ft]
5.0	35.02	31.53	-3.49
17.5	19.50	19.43	-0.07
35.0	11.64	11.70	0.06
60.0	6.46	6.41	-0.05
97.5	2.98	2.88	-0.10
152.5	1.03	0.97	-0.06
235.0	0.21	0.20	-0.01
360.0	0.02	0.02	0
535.0	0	0	0
650.0	0	0	0
890.0	0	0	0
1440.0	0	0	0

Table 1.5.2. Series 1.5b: comparison of head changes with time at distance from well r=60ft.

time [sec]	benchmark [ft]	code run [ft]	residual [ft]
6.0	0.08	0.27	0.19
14.5	0.85	0.95	0.10
26.5	2.10	1.98	-0.12
43.5	3.41	3.16	-0.25
67.5	4.57	4.29	-0.28
101.0	5.47	5.24	-0.23
149.0	6.10	5.94	-0.16
217.0	6.46	6.41	-0.05
313.0	6.62	6.67	0.05
449.0	6.68	6.79	0.11
641.0	6.69	6.84	0.15
912.0	6.69	6.85	0.16

IGWMC test #:	FTW-TST-1.6
input file name:	FTWORK_F.DAT
IGWMC output file:	FTWORK_F.OUT
code reference:	manual, section 5.1, p.133
description:	simulation of transient response of a regional two-aquifer flow system with constant head in the upper aquifer to increased pumping (additional wells) in lower aquifer in center of model domain; the real-world problem is taken from Andersen et al. (1984)
tested functions:	functionality: representation of three-dimensional flow in systems with high vertical contrast in hydraulic conductivity ; applicability: effects of a pumping well screened in multi model layers
model domain:	surficial aquifer of 30 ft thickness and a bedrock aquifer of 100 ft thickness separated by an aquitard of 40 ft thickness; the model area is part of a regional ground-water system and has no natural boundaries
grid:	rectangular block grid with variable grid in horizontal plane (see figure 1.6.1) and in variable layer thickness (see Figure 1.6.2); for details see FTWORK data file
boundary conditions:	in the rectangular model area all boundaries are taken as no-flow boundaries
initial conditions:	0 ft
parameters:	see FTWORK data file
time-stepping:	$\Delta t_k = 1.4\ \Delta t_{k-1}$ for k=1,....,8.
benchmark:	code intercomparison (MODFLOW, McDonald & Harbaugh, 1984); IGWMC Level 2.
test performed by:	problem setup for both codes by code developers; FTWORK code run by IGWMC to check output
type of comparison:	tabular listing (see Table 1.6.1); statistical measures
statistics:	see Table 1.6.1
control parameters:	SSOR relaxation factor=1.60; error criterion=1.0E-3; weighting factor=1.0; tolerance for non-linear iteration=1.0; max. # SSOR iterations=50
water balance accuracy:	in range 2.5E-3 -- 1.1E-5 percent
comments:	As the authors indicated, MODFLOW and FTWORK use the same block-centered finite difference formulations; the differences occuring in this test are due to the use of different solvers (SOR for FTWORK and SIP for MODFLOW)

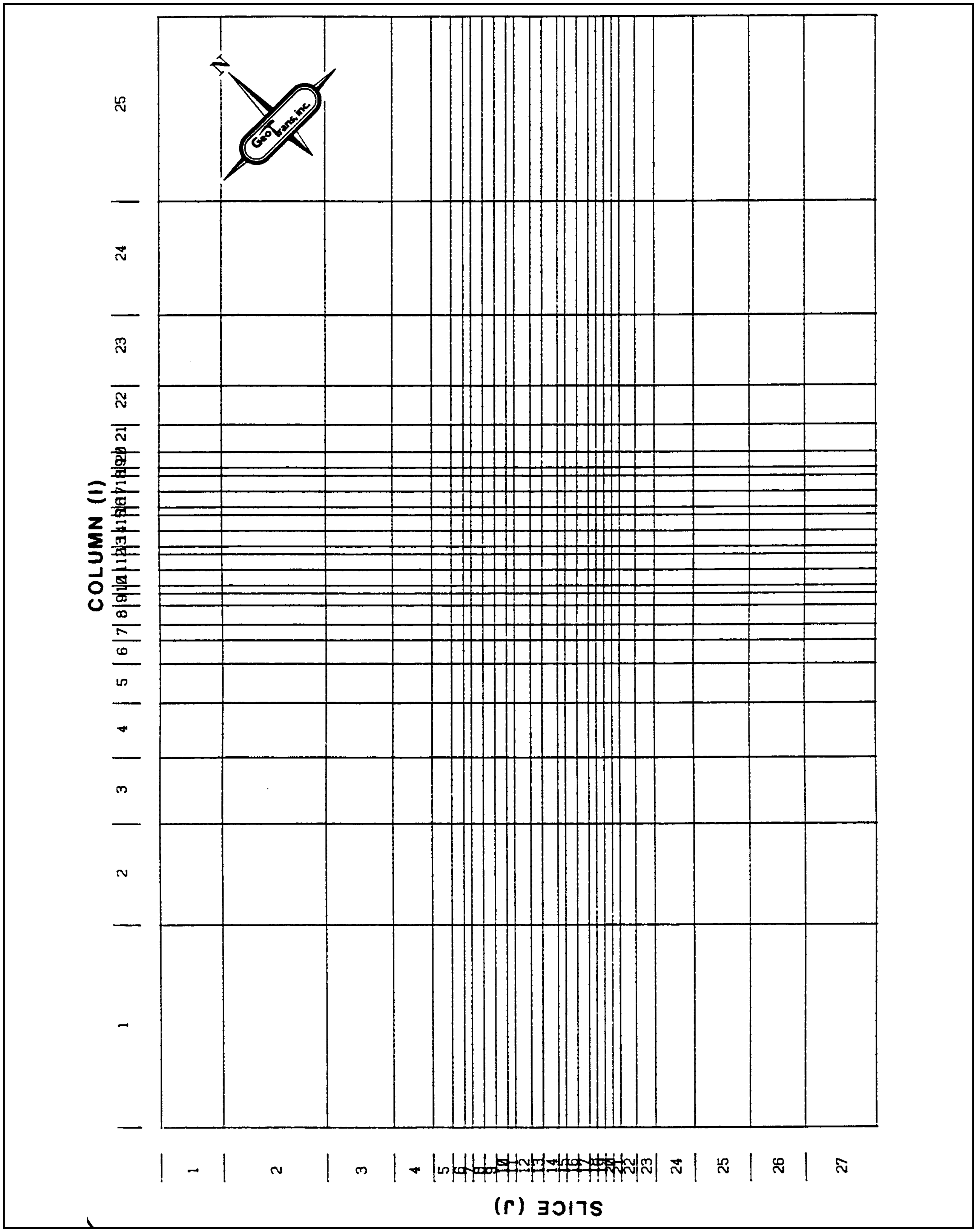


Figure 1.6.1. Horizontal discretization for test FTW-TST-1.6 (from Faust *et al.*, 1993).

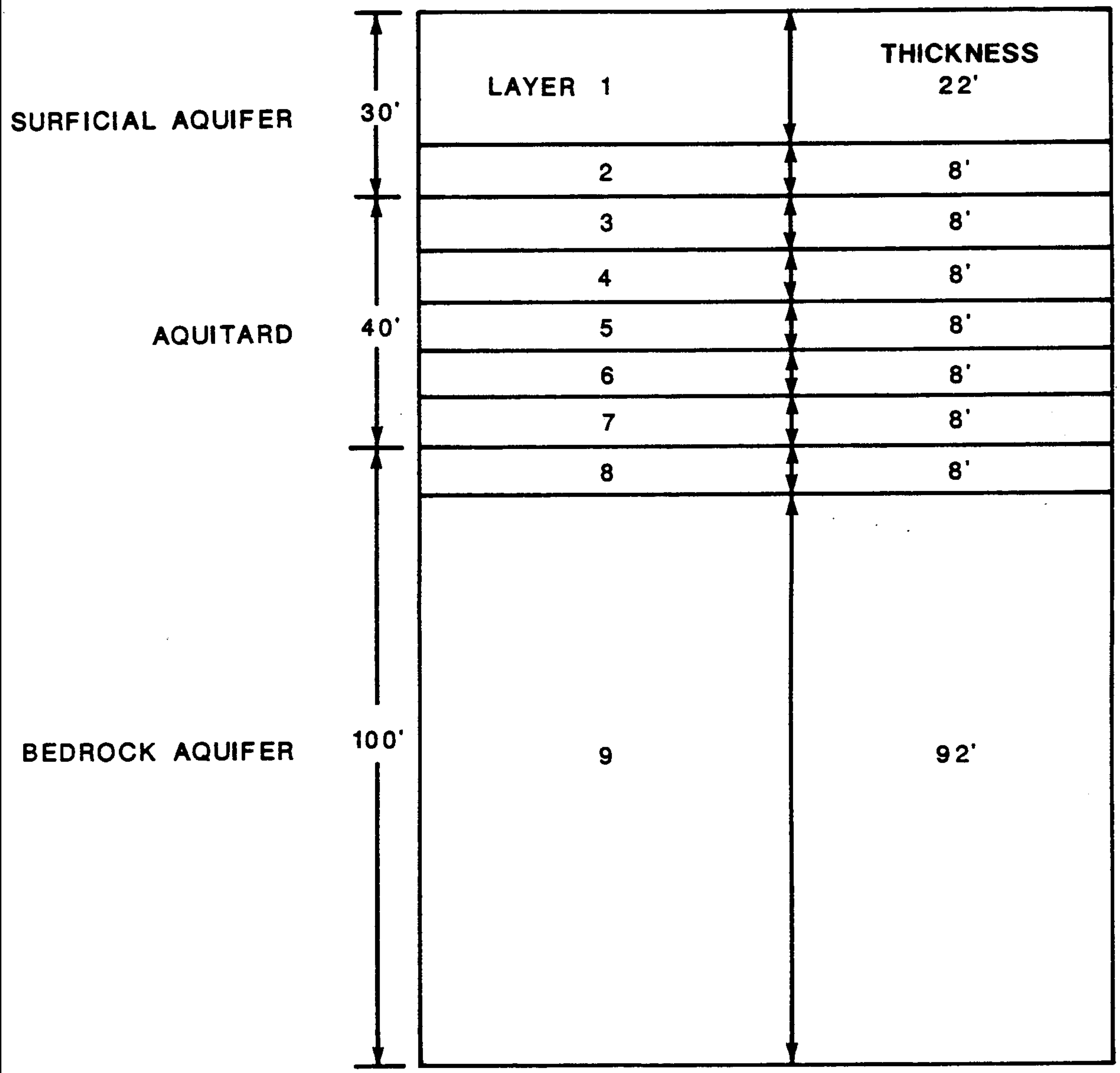


Figure 1.6.2. Vertical discretization for test FTW-TST-1.6 (from Faust *et al.*, 1993).

Table 1.6.1 Comparison of drawdowns calculated with MODFLOW and FTWORK at selected nodes for time=32 days

test 1.6	node	MODFLOW	FTWORK	residual	pos. res.	neg. res.	abs. res	squared res.	relative res.
		[ft]	[ft]	[ft]	[ft]	[ft]	[ft]		
	15,9,1	0.002	0.002	0.000	0.000	0.000	0.000	0.000	0.000
	15,9,2	0.002	0.002	0.000	0.000	0.000	0.000	0.000	0.000
	15,9,3	0.063	0.063	0.000	0.000	0.000	0.000	0.000	0.000
	15,9,4	0.245	0.245	0.000	0.000	0.000	0.000	0.000	0.000
	15,9,5	0.634	0.634	0.000	0.000	0.000	0.000	0.000	0.000
	15,9,6	1.476	1.477	0.001	0.001	0.000	0.001	0.000	0.001
	15,9,7	3.146	3.148	0.002	0.002	0.000	0.002	0.000	0.001
	15,9,8	4.593	4.596	0.003	0.003	0.000	0.003	0.000	0.001
	15,9,9	4.594	4.596	0.002	0.002	0.000	0.002	0.000	0.000
	17,15,1	0.011	0.011	0.000	0.000	0.000	0.000	0.000	0.000
	17,15,2	0.011	0.011	0.000	0.000	0.000	0.000	0.000	0.000
	17,15,3	0.488	0.488	0.000	0.000	0.000	0.000	0.000	0.000
	17,15,4	1.847	1.849	0.002	0.002	0.000	0.002	0.000	0.001
	17,15,5	4.572	4.576	0.004	0.004	0.000	0.004	0.000	0.001
	17,15,6	10.000	10.010	0.010	0.010	0.000	0.010	0.000	0.001
	17,15,7	19.540	19.560	0.020	0.020	0.000	0.020	0.000	0.001
	17,15,8	26.420	26.450	0.030	0.030	0.000	0.030	0.001	0.001
	17,15,9	26.430	26.450	0.020	0.020	0.000	0.020	0.000	0.001
sum =		104.074	104.168	0.094	0.094	0.000	0.094	0.002	0.008
total n =		18	18	18	18	18	18	18	18
measure =		5.782	5.787	0.005	0.005	0.000	0.005	0.010	0.000
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
measure =					0.030	0.000	#DIV/0!		0.001
					MPE	MNE	MER		MRE1

MAE = mean absolute error	NME = negative mean error
MB = mean benchmark	PME = positive mean error
ME = mean error	RE (relative error) = residual/num.code value
MNE = maximum negative error	RMSE = root mean square error
MP = mean prediction	MRE1 (mean rel.error) = ME/MP
MPE = maximum positive error	MRE2 (mean rel.error) = sum(RE)/N

IGWMC test #:	FTW-TST-1.7.1
input file name:	USGS0.DAT
IGWMC output file:	USGS0.OUT
code reference:	manual, section 5.2, p.136
description:	Steady-state flow in a system of three-aquifers separated by semi-pervious layers; flow into the system comes from areal recharge; flow out of the system takes place through buried drains, discharging wells, and specified head boundary cells, representing a lake. Drain is represented using drain option; this case creates heads file USGS0.RST for use as initial heads in test 1.7.2 and 1.7.3
tested functions:	drain function
model domain:	a rectangular block containing three aquifers separated by confining layers, bound at one side by a lake and at the other sides and the bottom by impermeable rock; uniform areal recharge in the shallow unconfined aquifer (see fig. 1.7.1)
grid:	rectangular grid of square cells horizontally (15 x 15 cells of 5000 x 5000 ft); three layers of 550 (top), 1, and 1 ft thickness, respectively; vertical flow through confining layers is lumped
boundary conditions:	no flow at three lateral boundaries and bottom, prescibed head at fourth lateral boundary, and uniform recharge with free surface at top boundary; 15 distributed discharging wells, 9 (buried) drains
initial conditions:	0 ft
parameters:	see Fig. 1.7.1; well and drain details are on p. 143 of FTWORK documentation
time-stepping:	steady-state (as determined by iteration error criterion)
benchmark:	code intercomparison (MODFLOW, McDonald & Harbaugh, 1984, Appendix D); IGWMC Level 2
tests performed by:	problem setup for both codes by code developers; FTWORK code run by IGWMC to check output
type of comparison:	tabular listing of heads along a line perpendicular to drain; statistical measures
statistics:	see Table 1.7.1
control parameters:	SSOR relaxation factor=1.80; error criterion=1.0E-3; weighting factor=1.0; tolerance for non-linear iteration=1.0E-3; max. # SSOR iterations=50; max. # of nonlinear iteration=30
iteration performance:	total of 7 nonlinear iterations; # SSOR iterations per nonlinear iteration: 50, 50, 38, 34, 22, 16, 1
water balance accuracy:	-1.22E-4 percent error

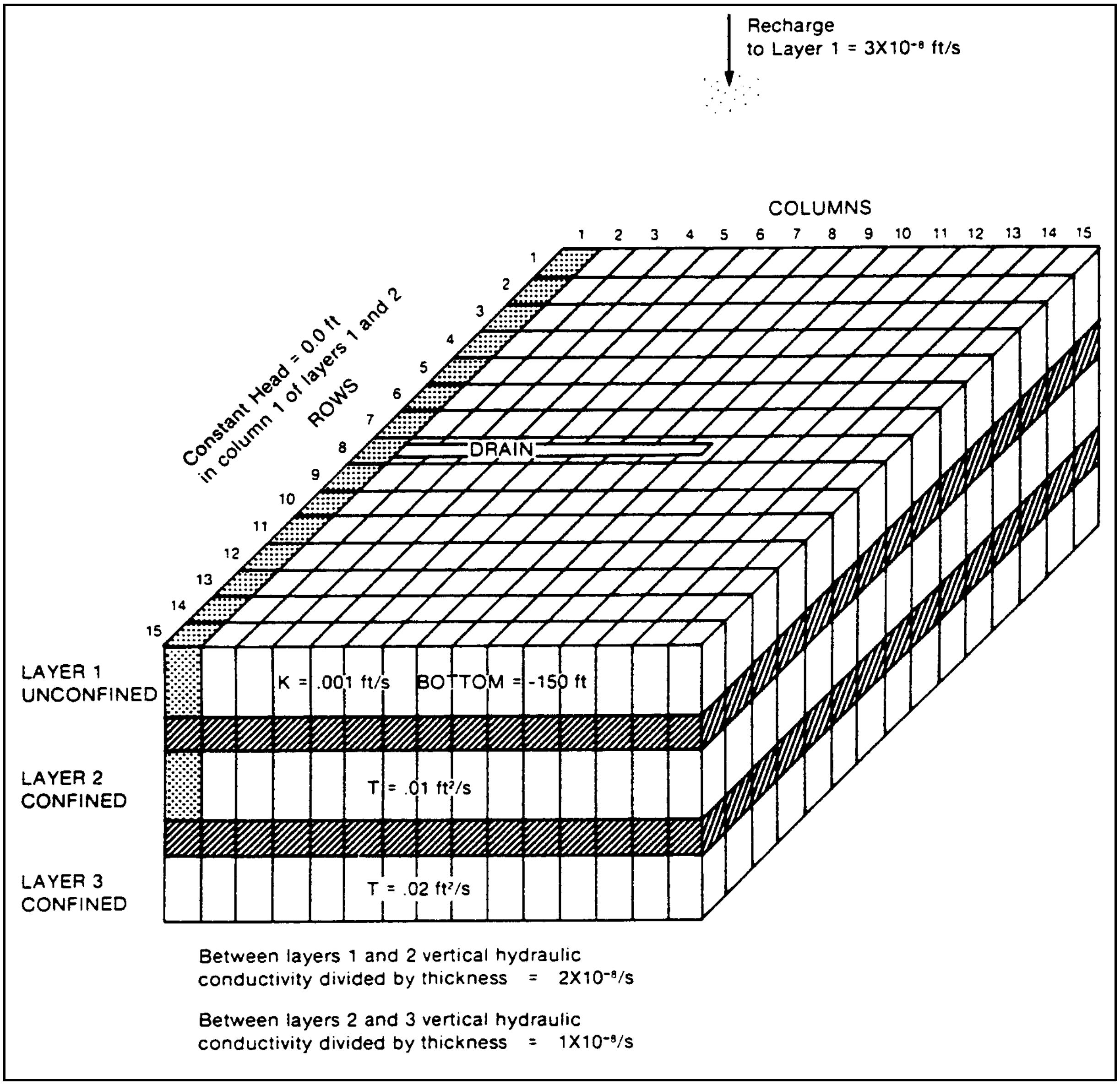


Figure 1.7.1. Model discretization and setup for test FTW-TST 1.7 (from Faust *et al.*, 1993).

Table 1.7.1. Comparison of heads calculated with MODFLOW and FTWORK at selected distance from drainl steady-state (final results)

test 1.7.1	distance from	MODFLOW	FTWORK	residual	pos. res.	neg. res.	abs. res	squared res.	relative res.
heads	drain [ft]	[ft]	[ft]	[ft]	[ft]	[ft]	[ft]		
	-32500	112.600	112.600	0.000	0.000	0.000	0.000	0.000	0.000
	-27500	111.000	111.000	0.000	0.000	0.000	0.000	0.000	0.000
	-22500	107.600	107.600	0.000	0.000	0.000	0.000	0.000	0.000
	-17500	102.500	102.500	0.000	0.000	0.000	0.000	0.000	0.000
	-12500	96.200	96.200	0.000	0.000	0.000	0.000	0.000	0.000
	-7500	90.800	90.800	0.000	0.000	0.000	0.000	0.000	0.000
	-2500	84.900	84.900	0.000	0.000	0.000	0.000	0.000	0.000
	0	77.300	77.300	0.000	0.000	0.000	0.000	0.000	0.000
	2500	66.100	66.100	0.000	0.000	0.000	0.000	0.000	0.000
	7500	65.600	65.600	0.000	0.000	0.000	0.000	0.000	0.000
	12500	59.300	59.300	0.000	0.000	0.000	0.000	0.000	0.000
	17500	63.200	63.200	0.000	0.000	0.000	0.000	0.000	0.000
	22500	60.900	60.900	0.000	0.000	0.000	0.000	0.000	0.000
	27500	68.800	68.800	0.000	0.000	0.000	0.000	0.000	0.000
	32500	72.000	72.000	0.000	0.000	0.000	0.000	0.000	0.000
sum =		1238.800	1238.800	0.000	0.000	0.000	0.000	0.000	0.000
total n =		15	15	15	15	15	15	15	15
measure =		82.587	82.587	0.000	0.000	0.000	0.000	0.000	0.000
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
measure =					0.000	0.000	#DIV/0!		0.000
					MPE	MNE	MER		MRE1
MAE = mean absolute error				NME = negative mean error					
MB = mean benchmark				PME = positive mean error					
ME = mean error				RE (relative error) = residual/num.code value					
MNE = maximum negative error				RMSE = root mean square error					

IGWMC test #:	FTW-TST-1.7.2
input file name:	USGS1.DAT
IGWMC output file:	USGS1.OUT
code reference:	manual, section 5.2, p.136.
description:	transient flow in a system of three-aquifers separated by semi-pervious layers; flow into the system comes from areal recharge; flow out of the system takes place through buried drains, discharging wells, and specified head boundary cells, representing a lake. Drain is represented using drain option; this case uses heads file USGS0.RST created by test 1.7.1 as initial heads (restart option)
tested functions:	drain function switch on/off during transient simulation, initial head file (restart option)
model domain:	a rectangular block containing three aquifers separated by confining layers, bound at one side by a lake and at the other sides and the bottom by impermeable rock; uniform areal recharge in the shallow unconfined aquifer (see fig. 1.7.1)
grid:	rectangular grid of square cells horizontally (15 x 15 cells of 5000 x 5000 ft); three layers of 550 (top), 1, and 1 ft thickness, respectively; vertical flow through confining layers is lumped
boundary conditions:	no flow at three lateral boundaries and bottom, prescibed head at fourth lateral boundary, and uniform recharge with free surface at top boundary; 15 distributed discharging wells, 9 (buried) drains
initial conditions:	generated by test 1.7.1
parameters:	see Fig. 1.7.1; well and drain details are on p. 143 of FTWORK documentation.
time-stepping:	see output file
benchmark:	code intercomparison (MODFLOW, McDonald & Harbaugh, 1984, Appendix D); IGWMC Level 2
tests performed by:	problem setup for both codes by code developers; FTWORK code run by IGWMC to check output; results are slightly different from those in table 5.4 of documentation and are closer to those generated by the authors with MODFLOW
type of comparison:	tabular listing of drain leakage for various times; statistical measures
statistics:	see Table 1.7.2
control parameters:	SSOR relaxation factor=1.80; error criterion=1.0E-3; weighting factor=1.5; tolerance for non-linear iteration=1.0E-3; max. # SSOR iterations=50; max. # of nonlinear iterations=1

iteration															
performance:	1	27	1.05E-2	6	38	3.36E-2	11	40	7.54E-2	16	42	9.86E-2	21	22	2.02E-3
(timestep,	2	29	5.82E-3	7	41	3.95E-2	12	43	9.14E-2	17	41	7.59E-2	22	21	7.69E-4
# iterations,	3	33	1.68E-2	8	40	5.01E-2	13	42	1.01E-1	18	39	4.88E-2	23	12	1.18E-3
% w.b. error)	4	33	2.47E-2	9	41	5.99E-2	14	44	1.05E-1	19	35	2.07E-2	24	1	3.25E-3
	5	35	2.76E-2	10	40	6.49E-2	15	42	1.04E-1	20	29	7.15E-3	25	1	3.95E-3

Table 1.7.2. Comparison of drain leakage for transient case calculated with MODFLOW and FTWORK for various times

test 1.7.2	time	MODFLOW	FTWORK	residual	pos. res.	neg. res.	abs. res	squared res.	relative res.
leakage	[days]	[cfs]	[cfs]	[cfs]	[cfs]	[cfs]	[cfs]		
	10	33.610	33.610	0.000	0.000	0.000	0.000	0.000	0.000
	24	34.840	34.840	0.000	0.000	0.000	0.000	0.000	0.000
	43.6	36.130	36.120	-0.010	0.000	-0.010	0.010	0.000	0.000
	71	37.490	37.480	-0.010	0.000	-0.010	0.010	0.000	0.000
	109.5	39.020	39.000	-0.020	0.000	-0.020	0.020	0.000	-0.001
	163.2	40.790	40.770	-0.020	0.000	-0.020	0.020	0.000	0.000
	238.5	42.900	42.860	-0.040	0.000	-0.040	0.040	0.002	-0.001
	343.9	45.930	45.340	-0.590	0.000	-0.590	0.590	0.348	-0.013
	491.5	49.740	49.780	0.040	0.040	0.000	0.040	0.002	0.001
	698.1	54.110	54.030	-0.080	0.000	-0.080	0.080	0.006	-0.001
	987.4	59.520	59.120	-0.400	0.000	-0.400	0.400	0.160	-0.007
	1392.4	68.670	67.080	-1.590	0.000	-1.590	1.590	2.528	-0.024
	1959.3	78.810	78.960	0.150	0.150	0.000	0.150	0.022	0.002
	2753	80.770	89.610	8.840	8.840	0.000	8.840	78.146	0.099
	3864.2	101.150	100.980	-0.170	0.000	-0.170	0.170	0.029	-0.002
	5419.9	111.970	111.900	-0.070	0.000	-0.070	0.070	0.005	-0.001
	7598	121.000	121.080	0.080	0.080	0.000	0.080	0.006	0.001
	10647	127.370	127.550	0.180	0.180	0.000	0.180	0.032	0.001
	14915	131.060	131.220	0.160	0.160	0.000	0.160	0.026	0.001
	20892	132.770	132.850	0.080	0.080	0.000	0.080	0.006	0.001
sum =		1427.650	1434.180	6.530	9.530	-3.000	12.530	81.320	0.055
total n =		20	20	20	20	20	20	20	20
measure =		71.383	71.709	0.326	0.476	-0.150	0.626	2.016	0.00277473
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
measure =					8.840	-1.590	3.177		0.005
					MPE	MNE	MER		MRE1

MAE = mean absolute error	NME = negative mean error
MB = mean benchmark	PME = positive mean error
ME = mean error	RE (relative error) = residual/num.code value
MNE = maximum negative error	RMSE = root mean square error
MP = mean prediction	MRE1 (mean rel.error) = ME/MP
MPE = maximum positive error	MRE2 (mean rel.error) = sum(RE)/N

IGWMC test #:	FTW-TST-1.7.3
input file name:	USGS2.DAT
IGWMC output file:	USGS2.OUT
code reference:	manual, section 5.2, p.136
description:	transient flow in a system of three-aquifers separated by semi-pervious layers; flow into the system comes from areal recharge; flow out of the system takes place through buried drains, discharging wells, and specified head boundary cells, representing a lake. Drain is represented using stream option; this case uses heads file USGS0.RST created by test 1.7.1 as initial heads (restart option)
tested functions:	stream/river boundary function, including switching between constant flux and variable flux as depends on stream stage
model domain:	a rectangular block containing three aquifers separated by confining layers, bound at one side by a lake and at the other sides and the bottom by impermeable rock; uniform areal recharge in the shallow unconfined aquifer (see fig. 1.7.1)
grid:	rectangular grid of square cells horizontally (15 x 15 cells of 5000 x 5000 ft); three layers of 550 (top), 1, and 1 ft thickness, respectively; vertical flow through confining layers is lumped
boundary conditions:	no flow at three lateral boundaries and bottom, prescibed head at fourth lateral boundary, and uniform recharge with free surface at top boundary; 15 distributed discharging wells, 9 (buried) drains
initial conditions:	generated by test 1.7.1
parameters:	see Fig. 1.7.1; well and drain details are on p. 143 of FTWORK documentation.
time-stepping:	see Table 1.7.3
benchmark:	code intercomparison (MODFLOW, McDonald & Harbaugh, 1984, Appendix D); IGWMC Level 2
tests performed by:	problem setup for both codes by code developers; FTWORK code run by IGWMC to check output; results are slightly different from those in Table 5.4 of documentation and are closer to those generated by the authors with MODFLOW
type of comparison:	tabular listing of stream leakage for various times; statistical measures (note: authors present tabular listing of heads along a line perpendicular to drain for two different times)
statistics:	see Table 1.7.3
control parameters:	SSOR relaxation factor=1.80; error criterion=1.0E-3; weighting factor=1.5; tolerance for nonlinear iteration=1.0E-3; max. # SSOR iterations=50; max. # of nonlinear iteration=1

Table 1.7.3. Comparison of stream leakage for transient case calculated with MODFLOW and FTWORK for various times

test 1.7.3	time	MODFLOW	FTWORK	residual	pos. res.	neg. res.	abs. res	squared res.	relative res.
leakage	[days]	[cfs]	[cfs]	[cfs]	[cfs]	[cfs]	[cfs]		
	10	85.060	85.110	0.050	0.050	0.000	0.050	0.002	0.001
	24	63.270	66.540	3.270	3.270	0.000	3.270	10.693	0.049
	43.6	45.550	44.990	-0.560	0.000	-0.560	0.560	0.314	-0.012
	71	27.380	32.410	5.030	5.030	0.000	5.030	25.301	0.155
	109.5	12.680	11.260	-1.420	0.000	-1.420	1.420	2.016	-0.126
	163.2	1.990	1.480	-0.510	0.000	-0.510	0.510	0.260	-0.345
	238.5	-7.020	-7.240	-0.220	0.000	-0.220	0.220	0.048	0.030
	343.9	-15.490	-15.590	-0.100	0.000	-0.100	0.100	0.010	0.006
	491.5	-23.920	-23.970	-0.050	0.000	-0.050	0.050	0.002	0.002
	698.1	-32.630	-32.640	-0.010	0.000	-0.010	0.010	0.000	0.000
	987.4	-41.910	-41.870	0.040	0.040	0.000	0.040	0.002	-0.001
	1392.4	-52.020	-51.930	0.090	0.090	0.000	0.090	0.008	-0.002
	1959.3	-63.190	-63.050	0.140	0.140	0.000	0.140	0.020	-0.002
	2753	-75.370	-75.180	0.190	0.190	0.000	0.190	0.036	-0.003
	3864.2	-87.990	-87.820	0.170	0.170	0.000	0.170	0.029	-0.002
	5419.9	-99.940	-99.880	0.060	0.060	0.000	0.060	0.004	-0.001
	7598	-109.850	-109.950	-0.100	0.000	-0.100	0.100	0.010	0.001
	10647	-116.790	-116.980	-0.190	0.000	-0.190	0.190	0.036	0.002
	14915	-120.760	-120.920	-0.160	0.000	-0.160	0.160	0.026	0.001
	20892	-122.570	-122.650	-0.080	0.000	-0.080	0.080	0.006	0.001
sum =		-733.520	-727.880	5.640	9.040	-3.400	12.440	38.823	-0.244
total n =		20	20	20	20	20	20	20	20
measure =		-36.676	-36.394	0.282	0.452	-0.170	0.622	1.393	-0.01222461
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
measure =					5.030	-1.420	2.659		-0.008
					MPE	MNE	MER		MRE1
MAE = mean absolute error				NME = negative mean error					
MB = mean benchmark				PME = positive mean error					
ME = mean error				RE (relative error) = residual/num.code value					
MNE = maximum negative error				RMSE = root mean square error					
MP = mean prediction				MRE1 (mean rel.error) = ME/MP					
MPE = maximum positive error				MRE2 (mean rel.error) = sum(RE)/N					

IGWMC test #:	FTW-TST-1.8
input file name:	ETPROB14.DAT
IGWMC output file:	ETPROB14.OUT
code reference:	manual, section 5.5.2, p.172
description:	two-dimensional transient flow in a homogeneous, isotropic unconfined aquifer with depth-limited evapotranspiration and well-pumping
tested functions:	depth-limited evapotranspiration and dewatering
model domain:	a rectangular block containing three aquifers separated by confining layers, bound at one side by a lake and at the other sides and the bottom by impermeable rock; uniform areal recharge in the shallow unconfined aquifer (see Fig. 1.7.1)
grid:	uniform horizontal grid with square cells (7 x 7 cells of 100 x 100 ft)
boundary conditions:	no flow at lateral boundaries and bottom, evapotranspiration along a line of cells (column 4), and free surface at top boundary (unconfined); 1 discharging well (node 1,1; 2500 ft^3/day)
initial conditions:	10 ft
parameters:	bottom elevation = -50 ft; storage coefficient = 0.1; hydraulic conductivity = 10 ft/day; maximum ET = 0.2 ft/day; ET extinction depth = 10 ft; ET surface elevation = 10 ft
time-stepping:	20 time steps in 365 days with multiplier of 1.2 with an additional refinement of 20 steps in the beginning of the simulation to ensure proper mass balance for FTWORK
benchmark:	code intercomparison (MODFLOW; McDonald & Harbaugh, 1984); EPA MODFLOW examples manual by Andersen (1993), problem 14; IGWMC Level 2
tests performed by:	problem setup for both codes by code developers; FTWORK code run by IGWMC to check output
type of comparison:	tabular listing of heads versus time at selected nodes, and of ET rates versus time stepping and number of iterations
statistics:	not generated
control parameters:	SSOR relaxation factor=1.80; error criterion=1.0E-4; weighting factor=1.5; tolerance for nonlinear iteration=1.0E-4; max. # SSOR iterations=50; max. # of nonlinear iteration=1
iteration performance:	water balance error in range 13.0 -- 5.0E-4 percent
comments:	According to the authors, this problem shows significant differences between the efficiency of MODFLOW SIP solver and the FTWORK SSOR solver in cases with significant reduction of saturated thickness during the simulation (p.174 of documentation)

IGWMC test #:	FTW-TST-1.9
input file name:	TRESCOT.DAT
IGWMC output file:	TRESCOT.OUT
code reference:	manual, section 5.5.1, p.164
description:	two-dimensional steady-state flow in an unconfined aquifer which is separated from an underlying aquifer by a leaky confining bed; the shallow aquifer is subject to recharge from precipitation, depth-limited evapotranspiration, pumping, and upward leakage from the underlying confined aquifer
tested functions:	steady-state evapotranspiration option
model domain:	arbitrarily bounded single layer area representing the unconfined aquifer (see Figure 1.9.1)
grid:	14 cells in x-direction, 10 cells in y-direction, and 1 cell in z-direction; grid spacings range from 1850 to 450 ft in x-direction and from 1550 to 250 ft in y-direction (see Figure 1.9.1)
boundary conditions:	combination of specified flux, specified head and zero-flux boundaries; boundary conditions in upper and lower aquifer are indentical (see Figure 1.9.1)
initial conditions:	not discussed in documentation (see input and output files)
parameters:	not discussed in documentation (see input and output files)
time-stepping:	n.a.
benchmark:	code intercomparison using Trescott, Pinder and Larson (1976); sample problem as given by Trescott, Pinder and Larson (1976)
test performed by:	problem setup for FTWORK by code developers; code run and output checked by IGWMC using data set prepared by developers
type of comparison:	graphical display for head comparison along a model row; tabular listing of mass balance results
statistics:	not generated
control parameters:	SSOR relaxation factor=1.86; error criterion=3.0E-4; weighting factor=1.0; tolerance for nonlinear iteration=3.0E-4; max. # SSOR iterations=1; max. # of nonlinear iteration=500
iteration performance:	244 nonlinear iterations; w.b.error=3.40E-2 percent
comments:	According to the authors, this problem shows significant differences between the efficiency of MODFLOW SIP solver and the FTWORK SSOR solver for ET problems (p.169 of documentation)

factor=1.0;

tolerance for nonlinear iteration=3.0E-4; max. # SSOR iterations=1;

max. # of nonlinear iteration=500

iteration performance: 244 nonlinear iterations; w.b.error=3.40E-2 percent

comments: According to the authors, this problem shows significant differences between the efficiency of MODFLOW SIP solver and the FTWORK SSOR solver for ET problems (p.169 of documentation)

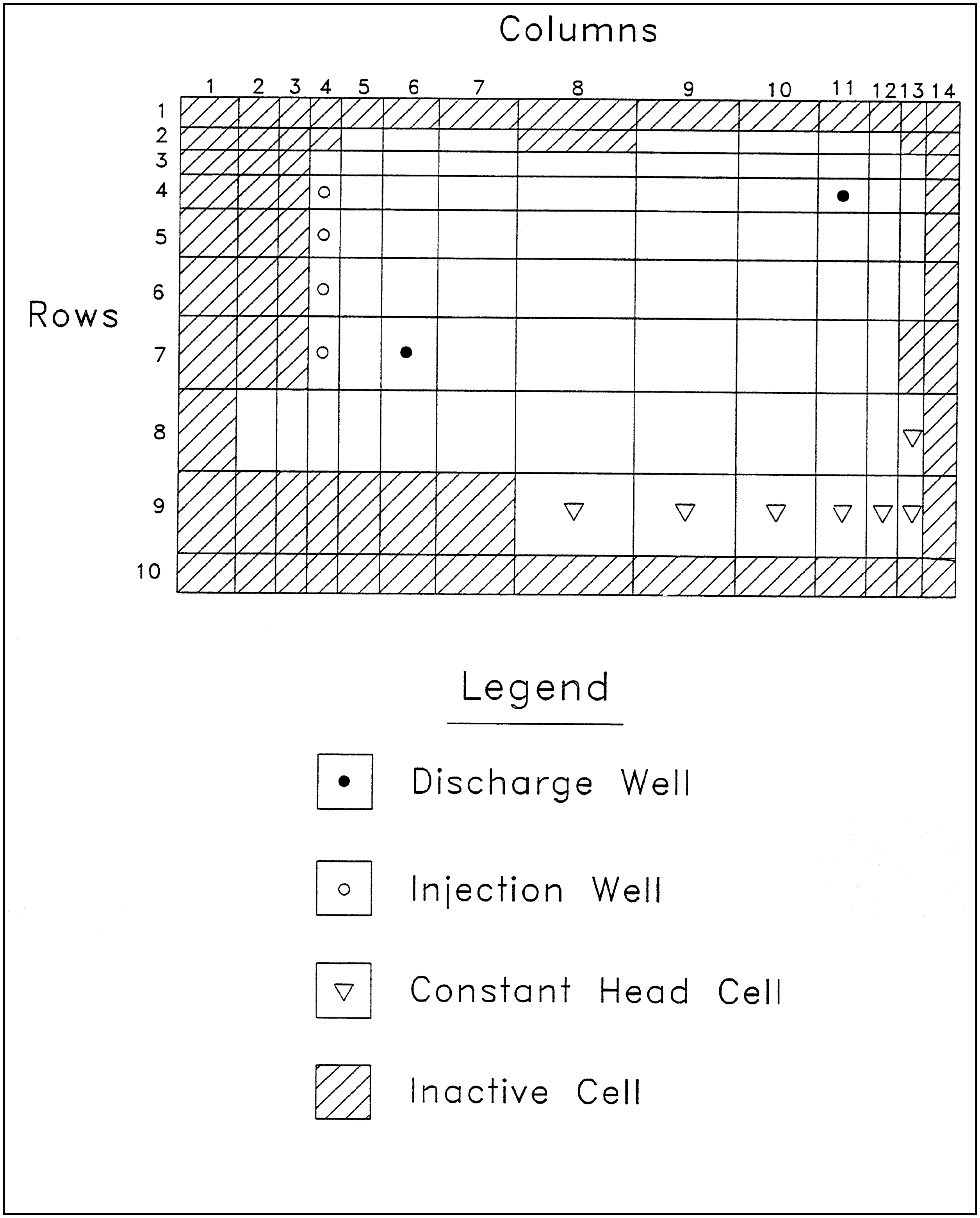


Figure 1.9.1. Model discretization and setup for test FTW-TST 1.9 (from Faust *et al.*, 1993)

IGWMC test #:	FTW-TST-2.1
input file name:	HI-1A.DAT, HI-1B.DAT, HI-1C.DAT, HI-1D.DAT
IGWMC output file:	HI-1A.OUT, HI-1B.OUT,HI-1C.OUT, HI-1D.OUT
code reference:	manual, section 4.2.1, p.81
description:	transient one-dimensional advective-dispersive transport in an infinite porous medium with a uniform flow field (steady-state one-dimensional flow) representing flow and transport from a fully-penetrating stream directly into an aquifer
tested functions:	numerical dispersion as function of alternate numerical approximations using different combinations of spatial- and time-differencing approximations (upstream weighting, time weighting, and central difference)
model domain:	one-dimensional
grid:	41 one-dimensional cells of 10 m length
boundary conditions:	at x=0, $C=C_o$ for t>0 and at $x=\infty$, C=0 for t>0; dispersive flux at outer boundary is 0
initial conditions:	zero-concentrations in aquifer
parameters:	hydraulic conductivity=40 m/d; porosity=0.25; hydraulic gradient=0.025; longitudinal dispersivity=5 m; retardation factor=1; and concentration at the source C_o=1 mg/m^3
time-stepping:	Δt=2.5 days
benchmark:	analytical solution (Bear, 1979, p.269.)
test performed by:	problem setup by code developers; code run and output checked by IGWMC using data set prepared by developers
type of comparison:	graphic representation of concentration profiles for two times; tabular listing of numerical and benchmark results for concentration versus distance
statistics:	not generated
sensitivity analysis:	time weighting factor (0.5 and 1.0); central difference versus upstream weighting

control parameters

case	SSOR relax. factor		error crit.		weighting factor		tolerance for nonlin. iterations	max. # SSOR iterations		max. # nonlin. iterations		weigh-ing in space
	flow	transp.	flow	transp.	nonlin.	time		flow	transp.	flow	transp.	
HI-1A	1.6	1.6	1.0E-5	1.0E-5	1.0	1.0	5.0E-3	1	1	1	1	upstream
HI-1B	1.6	1.6	1.0E-5	1.0E-5	1.0	0.5	5.0E-3	1	1	1	1	upstream
HI-1C	1.6	1.6	1.0E-5	1.0E-5	1.0	1.0	5.0E-3	1	1	1	1	central
HI-1D	1.6	1.6	1.0E-5	1.0E-5	1.0	0.5	5.0E-3	1	1	1	1	central

iteration performance:

case HI-1A; water balance error (%): 8.87E-13; solute balance error (%): in range 1.6E-13 -- 0.00
case HI-1B; water balance error (%): 8.87E-13; solute balance error (%): in range 8.9E-13 -- 0.00
case HI-1C; water balance error (%): 8.87E-13; solute balance error (%): in range 5.7E-14 -- 0.00
case HI-1D; water balance error (%): 8.87E-13; solute balance error (%): in range 5.7E-14 -- 0.00

IGWMC test #: FTW-TST-2.2.1

input file name: RUN1A.DAT

IGWMC output file: RUN1A.OUT

code reference: manual, section 4.2.2, p.87

description: transient two-dimensional advective-dispersive transport of a conservative tracer from a fully-penetrating point source with constant release rate in a uniform flow field in a homogeneous confined aquifer of constant thickness using a parallel grid; cross-products included

tested functions: longitudinal and transverse dispersion in two dimensions; grid orientation effects with or without cross-products for dispersivity (compare with test 2.2.2 and 2.2.3)

model domain: rectangular bounded area replaces infinite domain

grid: regular grid with 39 x 19 square cells of 30 x 30m; single layer of 33.5m

boundary conditions: two parallel prescribed head boundaries at opposite sides of model domain and no-flow conditions at other two parallel boundaries to ensure uniform flow with given flow rate; zero concentration at all boundary segments; constant solute injection rate at location x=180m and y=270m from grid origin

initial conditions: concentration = 0 mg/l

parameters: same as in Wilson and Miller (1978); QC_0=7.04 g/m.d (source strength); q=0.161 m/d (specific discharge); ϕ=0.35 (porosity); α_T=21.3 m (transverse dispersivity); α_L=4.3 m (longitudinal dispersivity); m=33.5 m (aquifer thickness); R=1 (no retardation); λ=0 d^{-1} (decay coefficient)

time-stepping: Δt=100 days; comparison at t=1400 days

benchmark: analytical solution (Wilson & Miller, 1978)

IGWMC implementation: MATHCAD 5.0 fileSOL2D-01.MCD for concentration versus distance along plume centerline; file SOL2D-04.MCD for concentration transverse to plume centerline at distance x=420 m from source

test performed by: problem setup for numerical code by code developers; code run and benchmark comparison by IGWMC using IGWMC generated benchmark

type of comparison: tabular listing and graphical representation of numerical results and benchmark for concentration versus distance from source along centerline and transverse to centerline for different grid orientations (see Table 2.2.1-a and -b. and Figure 2.2-a and -b), statistical measures prepared by IGWMC statistics: computed by IGWMC; see Table 2.2.1-a and -b

sensitivity analysis: grid orientation, dispersion cross-products

control parameters: SSOR relaxation factor (flow) = 1.85; SSOR relaxation factor (transport) = 1.3; error criterion (flow) = 1.0E-5; error criterion (transport) = 1.0E-5; weighting factor for nonlinear iterations = 1.0; weighting factor for time derivative = 0.5; tolerance for non-linear iteration=0.0; max. # SSOR iterations (flow) = 75; max. # SSOR iterations (transport) = 75; max. # of nonlinear iteration (flow) = 1; max. # of nonlinear iterations (transport) = 1; central difference in space

iteration performance: water balance error (percent) = 3.1E-2; solute balance error (percent) in range 1.1E-4 -- 3.0E-7; 60 flow iterations; up to 20 transport iterations per time step

comments: source strength listed in documentation as 704 g/m.d; in data file for numerical code source strength is set at 7.04 g/m.d

---

IGWMC test #: FTW-TST-2.2.2

input file name: RUN2A.DAT

IGWMC output file: RUN2A.OUT

code reference: manual, section 4.2.2, p.87

description: transient two-dimensional advective-dispersive transport of a conservative tracer from a fully-penetrating point source with constant release rate in a uniform flow field in a homogeneous confined aquifer of constant thickness using a skewed grid; cross-products included

tested functions: longitudinal and transverse dispersion in two dimensions; grid orientation effects with or without cross-products for dispersivity (compare with test 2.2.1 and 2.2.3)

model domain: rectangular bounded area replaces infinite domain

grid: regular grid with 39 x 39 square cells of 30 x 30m under 45^{o} with flow direction; single layer of 33.5m

boundary conditions: fixed head along all boundaries such that uniform flow is achieved; zero concentration at all boundary segments; constant injection rate at location x=254.5m and x=0m from grid origin

initial conditions: concentration = 0 mg/l

parameters: see test 2.2.1

time-stepping: 100d; comparison at t=1400d

benchmark: analytical solution (Wilson & Miller, 1978)

IGWMC implementation: MATHCAD 5.0 file SOL2D-02.MCD for concentration versus distance along plume centerline; MATHCAD file SOL2D-03.MCD for concentration transverse to plume centerline at x=424.26 m from source

Table 2.2.1-a. Comparison of concentration values with distance from source along plume centerline for t=1400 days (parallel grid)

test 2.2.1-a	distance	benchmark	num. code	residual	pos. res.	neg. res.	abs. res	squared res.	relative res.
	[m]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]		
along plume	30	0.9647	0.9167	-0.0480		-0.0480	0.0480	0.0023	-0.0524
centerline	60	0.7156	0.7761	0.0605	0.0605		0.0605	0.0037	0.0780
	90	0.6267	0.6707	0.0440	0.0440		0.0440	0.0019	0.0656
	120	0.5425	0.5902	0.0477	0.0477		0.0477	0.0023	0.0808
	150	0.4849	0.5280	0.0431	0.0431		0.0431	0.0019	0.0816
	180	0.4422	0.4788	0.0366	0.0366		0.0366	0.0013	0.0764
	210	0.4086	0.4395	0.0309	0.0309		0.0309	0.0010	0.0703
	240	0.3811	0.4070	0.0259	0.0259		0.0259	0.0007	0.0636
	270	0.3576	0.3799	0.0223	0.0223		0.0223	0.0005	0.0587
	300	0.3369	0.3560	0.0191	0.0191		0.0191	0.0004	0.0537
	330	0.3179	0.3348	0.0169	0.0169		0.0169	0.0003	0.0505
	360	0.2999	0.3149	0.0150	0.0150		0.0150	0.0002	0.0476
	390	0.2823	0.2956	0.0133	0.0133		0.0133	0.0002	0.0450
	420	0.2646	0.2762	0.0116	0.0116		0.0116	0.0001	0.0420
	450	0.2465	0.2563	0.0098	0.0098		0.0098	0.0001	0.0382
	480	0.2277	0.2354	0.0077	0.0077		0.0077	0.0001	0.0327
	510	0.2083	0.2137	0.0054	0.0054		0.0054	0.0000	0.0253
	540	0.1881	0.1912	0.0031	0.0031		0.0031	0.0000	0.0162
	570	0.1675	0.1685	0.0010	0.0010		0.0010	0.0000	0.0059
	600	0.1468	0.1460	-0.0008		-0.0008	0.0008	0.0000	-0.0055
	630	0.1265	0.1242	-0.0023		-0.0023	0.0023	0.0000	-0.0185
	660	0.1068	0.1038	-0.0030		-0.0030	0.0030	0.0000	-0.0289
	690	0.0884	0.0857	-0.0027		-0.0027	0.0027	0.0000	-0.0315
	720	0.0716	0.0687	-0.0029		-0.0029	0.0029	0.0000	-0.0422
	750	0.0567	0.0543	-0.0024		-0.0024	0.0024	0.0000	-0.0442
	780	0.0438	0.0422	-0.0016		-0.0016	0.0016	0.0000	-0.0379
	810	0.0330	0.0323	-0.0007		-0.0007	0.0007	0.0000	-0.0217
sum =		8.1372	8.4867	0.3495	0.4139	-0.0644	0.4783	0.0169	0.6494
total n =		27	27	27	18	9	27	27	27
measure =		0.3014	0.3143	0.0129	0.0153	-0.0024	0.0266	0.0307	0.0722
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
measure =					0.0605	-0.0480	6.4270		0.0412
					MPE	MNE	MER		MRE1
MAE = mean absolute error				MRE1 (mean rel.error) = ME/MP					
MB = mean benchmark				MRE2 (mean rel.error) = sum (RE)/N					
ME = mean error				NME = negative mean error					
MNE = maximum negative error				PME = positive mean error					
MP = mean prediction				RE (relative error) = residual/num.code value					
MPE = maximum positive error				RMSE = root mean square error					

Table 2.2.1-b. Comparison of concentration values transverse to plume centerline at distance x=420m from source for t=1400 days (parallel grid)

test 2.2.1-b	distance	benchmark	num. code	residual	pos. res.	neg. res.	abs. res	squared res.	relative res.
	[m]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]		
	0	0.2646	0.2762	0.0116	0.0116	0.0000	0.0116	0.0001	0.0420
	30	0.2310	0.2339	0.0029	0.0029	0.0000	0.0029	0.0000	0.0124
	60	0.1551	0.1504	-0.0047	0.0000	-0.0047	0.0047	0.0000	-0.0312
	90	0.0818	0.0780	-0.0038	0.0000	-0.0038	0.0038	0.0000	-0.0487
	120	0.0349	0.0341	-0.0008	0.0000	-0.0008	0.0008	0.0000	-0.0235
	150	0.0124	0.0129	0.0005	0.0005	0.0000	0.0005	0.0000	0.0388
	180	0.0037	0.0044	0.0007	0.0007	0.0000	0.0007	0.0000	0.1591
	210	0.0010	0.0013	0.0003	0.0003	0.0000	0.0003	0.0000	0.2308
	240	0.0002	0.0004	0.0002	0.0002	0.0000	0.0002	0.0000	0.5000
	270	0.0000	0.0001	0.0001	0.0001	0.0000	0.0001	0.0000	1.0000
sum =		0.7847	0.7917	0.0070	0.0163	-0.0093	0.0256	0.0002	1.8796
total n =		10	10	10	10	10	10	10	10
measure =		0.0785	0.0792	0.0007	0.0016	-0.0009	0.0026	0.0043	0.1880
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
measure =					0.0116	-0.0047	1.7527		0.0088
					MPE	MNE	MER		MRE1

MAE = mean absolute error
MB = mean benchmark
ME = mean error
MNE = maximum negative error
MP = mean prediction
MPE = maximum positive error

MRE1 (mean rel.error) = ME/MP
MRE2 (mean rel.error) = sum (RE)/N
NME = negative mean error
PME = positive mean error
RE (relative error) = residual/num.code value
RMSE = root mean square error

test performed by: problem setup for numerical code by code developers; code run and benchmark comparison by IGWMC using IGWMC generated benchmark

type of comparison: tabular listing and graphical representation of numerical results and benchmark for concentration versus distance from source along centerline and transverse to centerline for different grid orientations (see Table 2.2.2-a and -b. and Figure 2.2-a and -b), statistical measures prepared by IGWMC

statistics: computed by IGWMC; see Table 2.2.2-a and -b

sensitivity analysis: see test 2.2.1

control parameters: SSOR relaxation factor (flow) = 1.85; SSOR relaxation factor (transport) = 1.3; error criterion (flow) = 1.0E-5; error criterion (transport) = 1.0E-5; weighting factor for nonlinear iterations = 1.0; weighting factor for time derivative = 0.5; tolerance for non-linear iteration=0.0; max. # SSOR iterations (flow) = 75; max. # SSOR iterations (transport) = 75; max. # of nonlinear iteration (flow) = 1; max. # of nonlinear iterations (transport) = 1; central difference in space

iteration performance: water balance error (percent) = 1.2E-7; solute balance error (percent) <8.4E-6; 60 flow iterations (steady-state); up to 15 transport iterations per time step

comments: see test 2.2.1

IGWMC test #: FTW-TST-2.2.3

input file name: RUN4A.DAT

IGWMC output file: RUN4A.OUT

code reference: manual, section 4.2.2, p.87

description: transient two-dimensional advective-dispersive transport of a conservative tracer from a fully-penetrating point source with constant release rate in a uniform flow field in a homogeneous confined aquifer of constant thickness using a skewed grid; lumped cross-products

tested functions: same as tests 2.2.1 and 2.2.2

model domain: rectangualr bounded area replaces infinite domain

grid: regular grid with 39 x 39 square cells of 30 x 30m under 45^{o} with flow direction; single layer of 33.5m

boundary conditions: fixed head along all boundaries such that uniform flow is achieved; zero concentration at all boundary segments; constant injection rate at location x=254.5m and x=0m from grid origin

initial conditions: concentration = 0 mg/l

parameters: see test 2.2.1

Table 2.2.2-a. Comparison of concentration values with distance from source along plume centerline for t=1400 days (skewed grid; cross-products included)

test 2.2.2-a	distance	benchmark	num. code	residual	pos. res.	neg. res.	abs. res.	squared res.	relative res.
	[m]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]		
along plume	42.43	0.8336	0.6398	-0.1938		-0.1938	0.1938	0.0376	-0.3029
centerline	84.85	0.6130	0.5075	-0.1055		-0.1055	0.1055	0.0111	-0.2079
	127.28	0.5267	0.4369	-0.0898		-0.0898	0.0898	0.0081	-0.2055
	169.71	0.4556	0.3902	-0.0654		-0.0654	0.0654	0.0043	-0.1676
	212.13	0.4065	0.3539	-0.0526		-0.0526	0.0526	0.0028	-0.1486
	254.56	0.3693	0.3283	-0.0410		-0.0410	0.0410	0.0017	-0.1249
	296.98	0.3389	0.3045	-0.0344		-0.0344	0.0344	0.0012	-0.1130
	339.41	0.3122	0.2824	-0.0298		-0.0298	0.0298	0.0009	-0.1055
	381.84	0.2871	0.2601	-0.0270		-0.0270	0.0270	0.0007	-0.1038
	424.26	0.2621	0.2366	-0.0255		-0.0255	0.0255	0.0007	-0.1078
	466.69	0.2362	0.2111	-0.0251		-0.0251	0.0251	0.0006	-0.1189
	509.12	0.2088	0.1835	-0.0253		-0.0253	0.0253	0.0006	-0.1379
	551.54	0.1802	0.1546	-0.0256		-0.0256	0.0256	0.0007	-0.1656
	593.97	0.1510	0.1258	-0.0252		-0.0252	0.0252	0.0006	-0.2003
	636.40	0.1222	0.0986	-0.0236		-0.0236	0.0236	0.0006	-0.2394
	678.82	0.0951	0.0743	-0.0208		-0.0208	0.0208	0.0004	-0.2799
	721.25	0.0710	0.0539	-0.0171		-0.0171	0.0171	0.0003	-0.3173
	763.68	0.0506	0.0377	-0.0129		-0.0129	0.0129	0.0002	-0.3422
	806.10	0.0343	0.0255	-0.0088		-0.0088	0.0088	0.0001	-0.3451
	848.53	0.0221	0.0166	-0.0055		-0.0055	0.0055	0.0000	-0.3313
	890.95	0.0135	0.0105	-0.0030		-0.0030	0.0030	0.0000	-0.2857
	933.38	0.0078	0.0065	-0.0013		-0.0013	0.0013	0.0000	-0.2000
sum =		5.5978	4.7388	-0.8590	0.0000	-0.8590	0.8590	0.0731	-4.5511
total n =		22	22	22	0	22	22	22	22
		0.2544	0.2154	-0.0390	#DIV/0!	-0.0390	0.0390	0.0576	-0.2069
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
					0.0000	-0.1938	#DIV/0!		-0.1813
					MPE	MNE	MER		MRE1
MAE = mean absolute error				MRE1 (mean rel.error) = ME/MP					
MB = mean benchmark				MRE2 (mean rel.error) = sum (RE)/N					
ME = mean error				NME = negative mean error					
MNE = maximum negative error				PME = positive mean error					
MP = mean prediction				RE (relative error) = residual/num.code value					
MPE = maximum positive error				RMSE = root mean square error					

Table 2.2.2-b. Comparison of concentration values transverse to plume centerline at distance x= 26m from source for t=1400 days (skewed grid; cross-products includes)

test 2.2.2-b	distance	benchmark	num. code	residual	pos. res.	neg. res.	abs. res	squared res.	relative res.
	[m]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]		
	0	0.2621	0.2366	-0.0255	0.0000	-0.0255	0.0255	0.0007	-0.1078
	42.43	0.2006	0.1963	-0.0043	0.0000	-0.0043	0.0043	0.0000	-0.0219
	84.85	0.0926	0.1088	0.0162	0.0162	0.0000	0.0162	0.0003	0.1489
	127.28	0.0277	0.0355	0.0078	0.0078	0.0000	0.0078	0.0001	0.2197
	169.71	0.0058	0.0029	-0.0029	0.0000	-0.0029	0.0029	0.0000	-1.0000
	212.13	0.0009	0.0000	-0.0009	0.0000	-0.0009	0.0009	0.0000	#DIV/0!
	254.56	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	#DIV/0!
	296.98	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	#DIV/0!
sum =		0.5897	0.5801	-0.0096	0.0240	-0.0336	0.0576	0.0010	#DIV/0!
total n =		8	8	8	8	8	8	8	5
measure =		0.0737	0.0725	-0.0012	0.0030	-0.0042	0.0072	0.0112	#DIV/0!
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
measure =					0.0162	-0.0255	-1.4000		-0.0165
					MPE	MNE	MER		MRE1

MAE = mean absolute error
MB = mean benchmark
ME = mean error
MNE = maximum negative error
MP = mean prediction
MPE = maximum positive error

MRE1 (mean rel.error) = ME/MP
MRE2 (mean rel.error) = sum (RE)/N
NME = negative mean error
PME = positive mean error
RE (relative error) = residual/num.code value
RMSE = root mean square error

time-stepping:	100d; comparison at t=1400d
benchmark:	analytical solution (Wilson & Miller, 1978)
IGWMC implementation:	same as test 2.2.2
test performed by:	problem setup for numerical code by code developers; code run and benchmark comparison by IGWMC using IGWMC generated benchmark
type of comparison:	tabular listing and graphical representation of numerical results and benchmark for concentration versus distance from source along centerline and transverse to centerline for different grid orientations (see Table 2.2.3-a and -b. and Figure 2.2-a and -b), statistical measures prepared by IGWMC
statistics:	computed by IGWMC; see Table 2.2.3-a and -b
control parameters:	SSOR relaxation factor (flow) = 1.85; SSOR relaxation factor (transport) = 1.3; error criterion (flow) = 1.0E-5; error criterion (transport) = 1.0E-5; weighting factor for nonlinear iterations = 1.0; weighting factor for time derivative = 0.5; tolerance for non-linear iteration=0.0; max. # SSOR iterations (flow) = 75; max. # SSOR iterations (transport) = 75; max. # of nonlinear iteration (flow) = 1; max. # of nonlinear iterations (transport) = 1; central difference in space
iteration performance:	water balance error (percent) = 1.2E-7; solute balance error (percent) <4.7E-6; 60 flow iterations (steady-state); up to 15 transport iterations per time step
comments:	As mentioned in the code documentation, the results suggest that when the grid is oriented parallel to the flow direction, the code produces accurate results. When the grid is oriented at a maximum angle with the flow direction, the distribution of the solute in both the flow direction and transverse to the flow direction is poorly simulated. This is of special concern when the plume front or edges are of interest (i.e., low concentration areas); the relative error or residuals reach the same order of magnitude as the actual concentrations. This problem is even exacerbated when the cross-products for the dispersion coefficient are lumped

Table 2.2.3-a. Comparison of concentration values with distance from source along plume centerline for t=1400 days (skewed grid; lumped cross-products)

test 2.2.3-a	distance	benchmark	num. code	residual	pos. res.	neg. res.	abs. res.	squared res.	relative res.
	[m]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]		
along plume	42.43	0.8336	0.4048	-0.4288		-0.4288	0.4288	0.1839	-1.0593
centerline	84.85	0.6130	0.3042	-0.3088		-0.3088	0.3088	0.0954	-1.0151
	127.28	0.5267	0.2539	-0.2728		-0.2728	0.2728	0.0744	-1.0744
	169.71	0.4556	0.2222	-0.2334		-0.2334	0.2334	0.0545	-1.0504
	212.13	0.4065	0.2000	-0.2065		-0.2065	0.2065	0.0426	-1.0325
	254.56	0.3693	0.1831	-0.1862		-0.1862	0.1862	0.0347	-1.0169
	296.98	0.3389	0.1694	-0.1695		-0.1695	0.1695	0.0287	-1.0006
	339.41	0.3122	0.1575	-0.1547		-0.1547	0.1547	0.0239	-0.9822
	381.84	0.2871	0.1461	-0.1410		-0.1410	0.1410	0.0199	-0.9651
	424.26	0.2621	0.1343	-0.1278		-0.1278	0.1278	0.0163	-0.9516
	466.69	0.2362	0.1210	-0.1152		-0.1152	0.1152	0.0133	-0.9521
	509.12	0.2088	0.1060	-0.1028		-0.1028	0.1028	0.0106	-0.9698
	551.54	0.1802	0.0896	-0.0906		-0.0906	0.0906	0.0082	-1.0112
	593.97	0.1510	0.0725	-0.0785		-0.0785	0.0785	0.0062	-1.0828
	636.40	0.1222	0.0561	-0.0661		-0.0661	0.0661	0.0044	-1.1783
	678.82	0.0951	0.0415	-0.0536		-0.0536	0.0536	0.0029	-1.2916
	721.25	0.0710	0.0293	-0.0417		-0.0417	0.0417	0.0017	-1.4232
	763.68	0.0506	0.0198	-0.0308		-0.0308	0.0308	0.0009	-1.5556
	806.10	0.0343	0.0129	-0.0214		-0.0214	0.0214	0.0005	-1.6589
	848.53	0.0221	0.0081	-0.0140		-0.0140	0.0140	0.0002	-1.7284
	890.95	0.0135	0.0049	-0.0086		-0.0086	0.0086	0.0001	-1.7551
	933.38	0.0078	0.0028	-0.0050		-0.0050	0.0050	0.0000	-1.7857
sum =		5.5978	2.7400	-2.8578	0.0000	-2.8578	2.8578	0.6232	-26.5407
total n =		22	22	22	0	22	22	22	22
		0.2544	0.1245	-0.1299	#DIV/0!	-0.1299	0.1299	0.1683	-1.2064
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
					0.0000	-0.4288	#DIV/0!		-1.0430
					MPE	MNE	MER		MRE1

MAE = mean absolute error
MB = mean benchmark
ME = mean error
MNE = maximum negative error
MP = mean prediction
MPE = maximum positive error

MRE1 (mean rel.error) = ME/MP
MRE2 (mean rel.error) = sum (RE)/N
NME = negative mean error
PME = positive mean error
RE (relative error) = residual/num.code value
RMSE = root mean square error

Table 2.2.3-b. Comparison of concentration values transverse to plume centerline at distance x=424.26m from source fro t=1400 days (skewed grid; lumped cross-products)

test 2.2.3-b	distance	benchmark	num. code	residual	pos. res.	neg. res.	abs. res	squared res.	relative res.
	[m]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]	[mg/l]		
	0	0.2621	0.1343	-0.1278	0.0000	-0.1278	0.1278	0.0163	-0.9516
	42.43	0.2006	0.1261	-0.0745	0.0000	-0.0745	0.0745	0.0056	-0.5908
	84.85	0.0926	0.1044	0.0118	0.0118	0.0000	0.0118	0.0001	0.1130
	127.28	0.0277	0.0763	0.0486	0.0486	0.0000	0.0486	0.0024	0.6370
	169.71	0.0058	0.0492	0.0434	0.0434	0.0000	0.0434	0.0019	0.8821
	212.13	0.0009	0.0128	0.0119	0.0119	0.0000	0.0119	0.0001	0.9297
	254.56	0.0000	0.0054	0.0054	0.0054	0.0000	0.0054	0.0000	1.0000
	296.98	0.0000	0.0020	0.0020	0.0020	0.0000	0.0020	0.0000	1.0000
sum =		0.5897	0.5105	-0.0792	0.1231	-0.2023	0.3254	0.0264	3.0194
total n =		8	8	8	8	8	8	8	8
measure =		0.0737	0.0638	-0.0099	0.0154	-0.0253	0.0407	0.0575	0.3774
		MB	MP	ME	PME	NME	MAE	RMSE	MRE2
measure =					0.0486	-0.1278	-1.6434		-0.1551
					MPE	MNE	MER		MRE1

MAE = mean absolute error	MRE1 (mean rel.error) = ME/MP
MB = mean benchmark	MRE2 (mean rel.error) = sum (RE)/N
ME = mean error	NME = negative mean error
MNE = maximum negative error	PME = positive mean error
MP = mean prediction	RE (relative error) = residual/num.code value
MPE = maximum positive error	RMSE = root mean square error

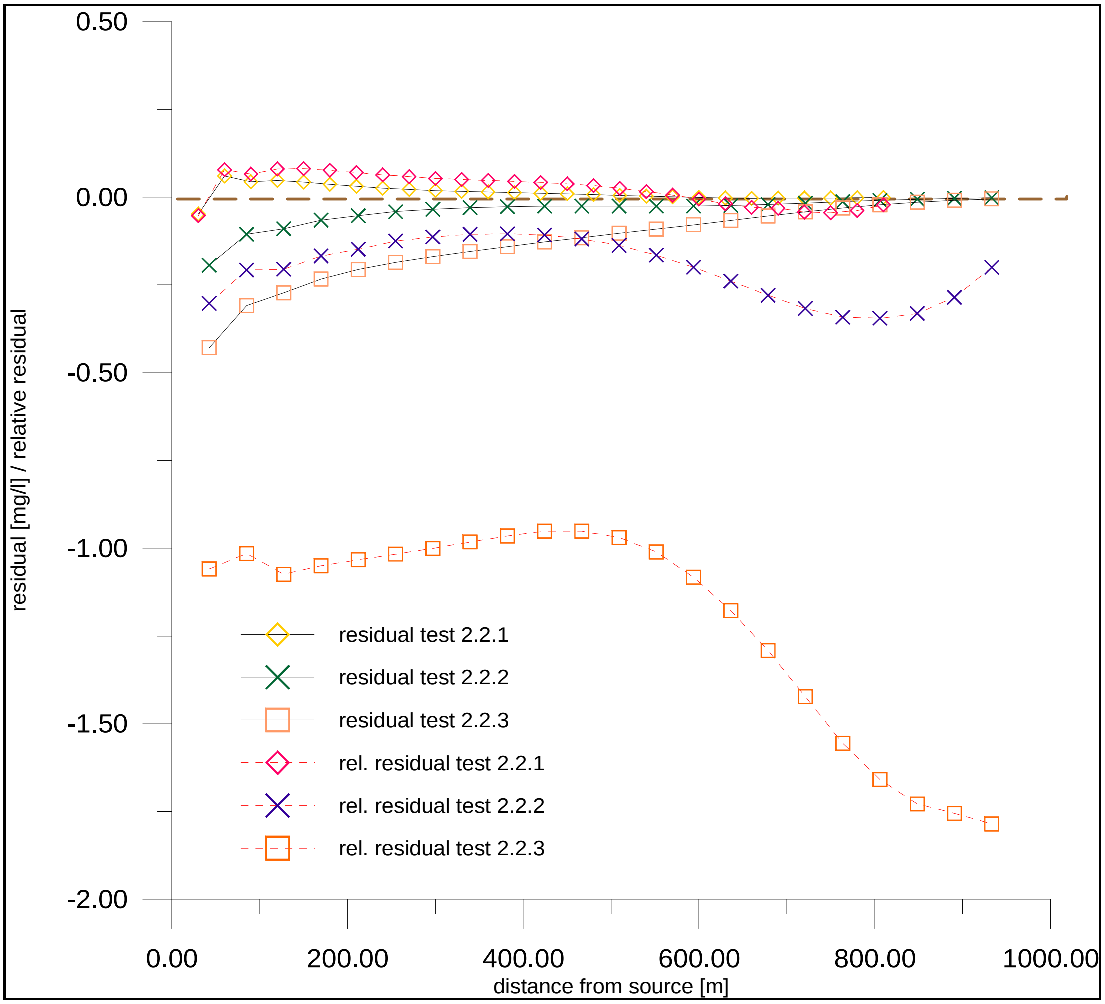


Figure 2.2-a: Combination plot of residuals and relative residuals along plume centerline (relative residuals are obtained by dividing residuals by the numerical results).

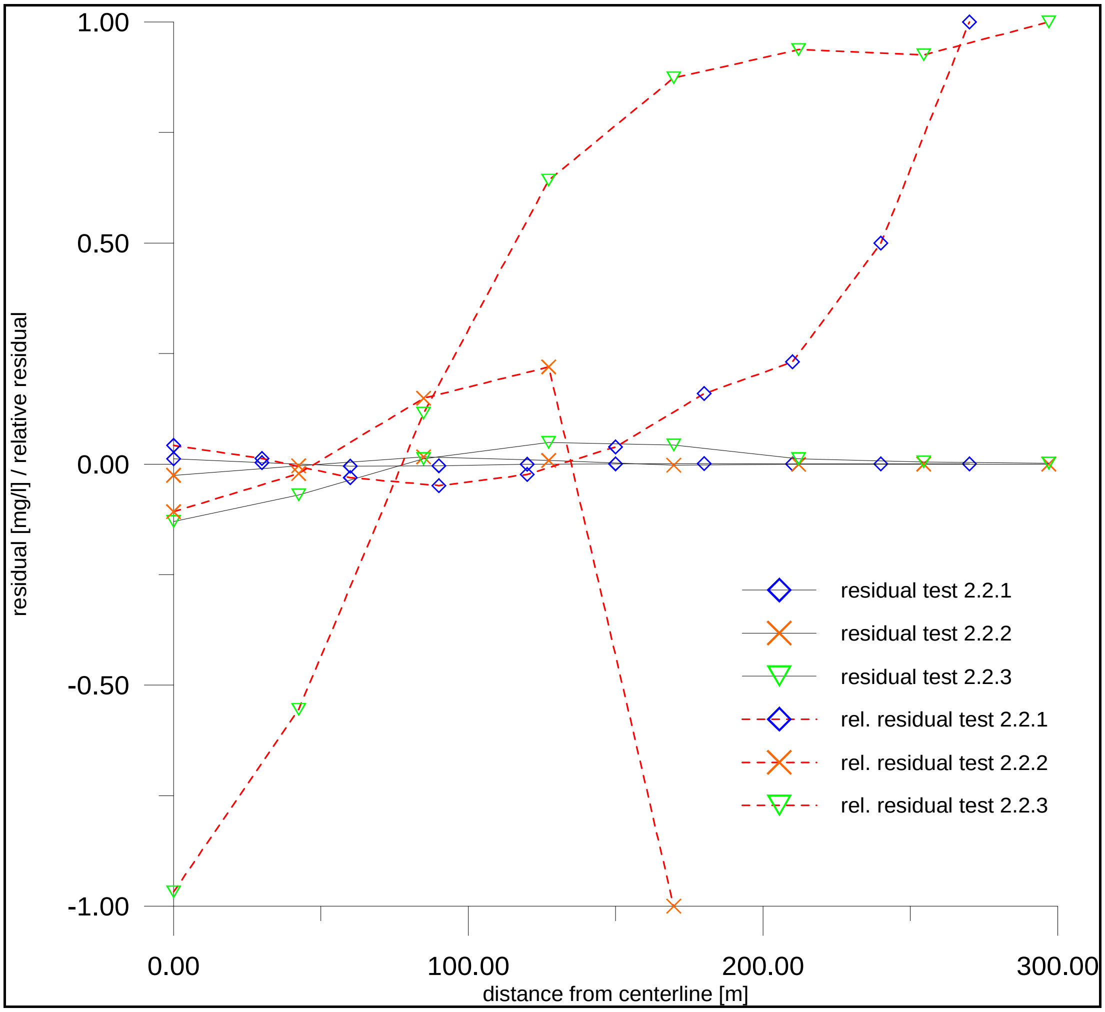


Figure 2.2-b: Combination plot of residuals and relative residuals transverse to plume centerline (relative residuals are obtained by dividing residuals by the numerical results).

IGWMC test #:	FTW-TST-2.3
input file name:	HI3.DAT, HI3_RADN.DAT
IGWMC output file:	HI3.OUT, HI3_RADN.OUT
code reference:	manual, section 4.2.3, p.105
description:	transient two-dimensional advective-dispersive transport of a nonconservative tracer from a fully-penetrating point source with constant release rate in a uniform flow field in a homogeneous confined aquifer of constant thickness using a parallel grid; the tracer is subjected to retardation and first-order (radio-active) decay
tested functions:	retardation and decay
model domain:	rectangular bounded area replaces infinite domain
grid:	regular grid with 39 x 19 square cells of 30 x 30m; single layer of 33.5m
boundary conditions:	two parallel prescribed head boundaries at opposite sides of model domain and no-flow conditions at other two parallel boundaries to ensure uniform flow with given flow rate; zero concentration at all boundary segments; constant solute injection rate at location x=180m and y=270m from grid origin
initial conditions:	concentration = 0 mg/l
parameters:	same as in Wilson and Miller (1978); QC_0=7.04 g/m.d (source strength); q=0.161 m/d (specific discharge); ϕ=0.35 (porosity); α_T=21.3 m (transverse dispersivity); α_L=4.3 m (longitudinal dispersivity); m=33.5 m (aquifer thickness); R=2 (retardation coefficient); λ=0.0019 d^{-1} (decay coefficient).
time-stepping:	Δt=100 days; comparison at t=1400 days
benchmark:	analytical solution (Wilson & Miller, 1978); intracomparison with FTW-TST-2.2.1
IGWMC implementation:	MATHCAD 5.0 fileSOL2D-06.MCD for concentration versus distance along plume centerline; file SOL2D-07.MCD for concentration transverse to plume centerline at distance x=420 m from source
test performed by:	authors; visually checked by IGWMC
type of comparison:	tabular listing and graphical representation of numerical results and benchmark (with and without decay) for concentration versus distance from source along centerline and transverse to centerline, and for point at centerline for various times
statistics:	not generated
control parameters:	SSOR relaxation factor (flow) = 1.85; SSOR relaxation factor (transport) = 1.3; error criterion (flow) = 1.0E-5; error criterion (transport) = 1.0E-7; weighting factor for nonlinear iterations = 1.0; weighting factor for time derivative = 0.5; tolerance for non-linear iteration=0.0; max. # SSOR iterations (flow) = 75; max. # SSOR iterations (transport) = 75; max. # of nonlinear iteration (flow) = 1; max. # of nonlinear iterations (transport) = 1; central difference in space

iteration performance: HI3: water balance error (percent) = 3.1E-2; solute balance error (percent) < 4.9E-8; 65 flow iterations (steady-state); up to 20 transport iterations per time step

HI3_RADN: water balance error (percent) = 3.1E-2; solute balance error (percent) < 4.5E-8; 65 flow iterations (steady-state); up to 20 transport iterations per time step

comments: FTWORK overpredicts slightly along centerline and in time, especially in the steep part of the curve; the code underpredicts slightly in transverse direction

---

IGWMC test #: FTW-TST-2.4.1

input file name: FR-6A.DAT, FR-6B.DAT, FR-6C.DAT

IGWMC output file: FR-6A.OUT, FR-6B.OUT, FR-6C.OUT

code reference: manual, section 4.2.4, p.105

description: transient one-dimensional advective-dispersive transport of a non-conservative tracer in a uniform flow field with nonlinear adsorption as defined by Freundlich isotherms

tested functions: Freundlich-type of adsorption

model domain: 16cm long one-dimensional domain

grid: Δx=0.02, 0.04, 0.08, 0.17, 0.30 cm, followed by 39 x 0.40 cm

boundary conditions: constant concentration at upgradient boundary (0.05 mg/l) and zero solute-flux at the downgradient boundary

initial conditions: concentration = 0 mg/l

parameters: q=0.037 cm/s (Darcy velocity or specific discharge); ϕ=0.37 (porosity); α_L=1.0 cm (longitudinal dispersivity); Qc_o=0.00185 mg cm^{-2} sec^{-1} (contaminant mass flux); n=0.7, 1.0, 0.3 (Freundlich adsorption exponent); C_{f2}=0.3 cm^3/g (Freundlich adsorption coefficient); λ=0.0019 d^{-1} (decay coefficient); ρ_a=2.519 g/cm^3 (aquifer bulk density)

time-stepping: Δt=1 sec; comparison at t=160 sec

benchmark: code intercomparison using BIO1D (Srinivasan and Mercer, 1987)

test performed by: authors; FTWORK code run by IGWMC to check output

type of comparison: tabular listing and graphical representation of numerical results of FTWORK and BIO1D (intercomparison) for concentration versus distance

statistics: not generated

sensitivity analysis: Freundlich isotherms exponents of n=0.7, 1.0, and 0.3

control parameters: SSOR relaxation factor (flow) = 1.6; SSOR relaxation factor (transport) = 1.6; error criterion (flow) = 1.0E-5; error criterion (transport) = 1.0E-5; weighting factor for

	nonlinear iterations = 1.0; weighting factor for time derivative = 0.5; tolerance for non-linear iteration=5.0E-3; max. # SSOR iterations (flow) = 1; max. # SSOR iterations (transport) = 1; max. # of nonlinear iterations (flow) = 1; max. # of nonlinear iterations (transport) = 15; central difference in space
iteration performance:	FR-6A: water balance error (percent) = 2.7E-10; solute balance error (percent) gradually increasing from < 1.0E-10 in the early time steps to 3.4E-2 in the final time step (160 time steps)
	FR-6B: water balance error (percent) = 2.7E-10; solute balance error (percent) varies between 6.1E-2 and 1.6E-4 (160 time steps)
	FR-6C: water balance error (percent) = 2.7E-10; solute balance error (percent) varies between 83.8 and 9.7E-3 with most values > 20.0 (160 time steps)
comments:	Documentation cautions for use of small values for n; may cause convergence problems; tests show poor mass balance for n=0.3

IGWMC test #:	FTW-TST-2.4.2
input file name:	FR-6D.DAT
IGWMC output file:	FR-6D.OUT
code reference:	manual, section 4.2.4, p.105
description:	transient one-dimensional advective transport of a non-conservative tracer in a uniform flow field with nonlinear adsorption as defined by Freundlich isotherms and molecular diffusion
tested functions:	molecular diffusion
model domain:	same as 2.4.1
grid:	same as 2.4.1
boundary conditions:	same as 2.4.1
initial conditions:	same as 2.4.1
parameters:	same as 2.4.1, except longitudinal dispersivity=0.0 cm, molecular diffusion coefficient is 0.1 cm^2/s, and n=0.3 (Freundlich exponent)
time-stepping:	same as 2.4.1
benchmark:	code intercomparison using BIO1D (Srinivasan and Mercer, 1987)
test performed by:	authors; FTWORK code run by IGWMC to check output
type of comparison:	tabular listing and graphical representation of numerical results of FTWORK and BIO1D (intercomparison) for concentration versus distance; intracomparison
statistics:	not generated

control parameters:	SSOR relaxation factor (flow) = 1.6; SSOR relaxation factor (transport) = 1.6; error criterion (flow) = 1.0E-5; error criterion (transport) = 1.0E-5; weighting factor for nonlinear iterations = 1.0; weighting factor for time derivative = 0.5; tolerance for non-linear iteration=5.0E-3; max. # SSOR iterations (flow) = 1; max. # SSOR iterations (transport) = 1; max. # of nonlinear iterations (flow) = 1; max. # of nonlinear iterations (transport) = 15; central difference in space
iteration performance:	water balance error (percent) = 2.7E-10; solute balance error (percent) varies between 61.8 and 1.3E-3 (160 time steps)
comment:	The mass balance error is comparable with the one occuring in test 2.4.1. due to the low Freundlich exponent value

---

IGWMC test #:	FTW-TST-2.5
input file name:	BATU.DAT
IGWMC output file:	BATU.OUT
code reference:	manual, section 4.2.5, p.114
description:	transient two-dimensional advective-dispersive transport of a non-conservative tracer from a constant flux-type source (third type or Cauchy condition at the inlet boundary); uniform flow field in a homogeneous porous medium; vertical plane source from top to bottom of aquifer, perpendicular to the flow direction
tested functions:	constant 3rd type boundary condition, longitudinal and horizontal transverse dispersion, retardation.
model domain:	rectangular bounded domain with source asymmetrically placed at inlet boundary (see Fig. 2.5.1)
grid:	rectangular grid with 19 x 39 varying size cells (see Fig. 2.5.1)
boundary conditions:	two parallel no flow boundaries and two parallel constant head boundaries for flow creating uniform flow perpendicular to source boundary (see Fig. 2.5.1); source width=5 m, source strength=0.0375 g/m/d
initial conditions:	concentration = 0 mg/l
parameters:	specific discharge=0.15 m/d; porosity = 0.25; longitudinal dispersivity = 21.3 m; transverse dispersivity = 4.3 m; aquifer length = 185 m; aquifer width = 53 m; hydraulic conductivity = 13.875 m/d (both horizontal directions); retardation coeff. = 1.0
time-stepping:	180 steps of 1 day
benchmark:	analytical solution (Batu, 1992)
test performed by:	authors; FTWORK code run by IGWMC to check output
type of comparison:	tabular listing and graphical representation of numerical results of FTWORK and benchmark for concentration versus distance from source in flow direction and perpendicular to flow direction, and versus time for a specific location

statistics:	not generated
control parameters:	SSOR relaxation factor (flow) = 1.85; SSOR relaxation factor (transport) = 1.3; error criterion (flow) = 1.0E-5; error criterion (transport) = 1.0E-5; weighting factor for nonlinear iterations = 1.0; weighting factor for time derivative = 0.5; tolerance for non-linear iteration=0.0; max. # SSOR iterations (flow) = 75; max. # SSOR iterations (transport) = 75; max. # of nonlinear iterations (flow) = 1; max. # of nonlinear iterations (transport) = 1; central difference in space
iteration performance:	water balance error (percent) = 2.8E-2 in 70 iterations; solute balance error (percent) decreases from about 5.0E-3 to about 1.5E-5 with time (180 time steps)
comments:	slight differences with benchmark contributed by authors to spatial and temporal discretization

---

IGWMC test #:	FTW-TST-2.6
input data file name:	NEW3.DAT
IGWMC output file:	NEW3.OUT
code reference:	manual, section 5.4, p.156
description:	simulation of three-dimensional steady-state flow and transient transport in a three-aquifer system with variable thickness; the aquifers are separated by aquitards; model includes streams, seeplines, seepage basins, ground-water divides, and near-impermeable confining layers at part of the boundary
tested functions:	applicability to support conceptualization, determining effects of preferential flow paths on plume migration, and studying effects of source removal options (closure and capping of seepage basins) on downgradient concentration distribution
model domain:	irregular shaped bounded model domain simulated in quasi-three-dimensional mode
grid:	44 by 43 variably size cells (see Fig. 2.6.1) and three layers of varying thickness representing the aquifers
boundary conditions:	combination of various 1st, 2nd and 3rd type boundary conditions
benchmark:	no benchmark, no comparison; applicability demonstration
test performed by:	authors; FTWORK code run by IGWMC to check output
type of evaluation:	normalized concentration contours for each aquifer, at beginning of closure; concentration versus time graph for downgradient node
statistics:	n.a.

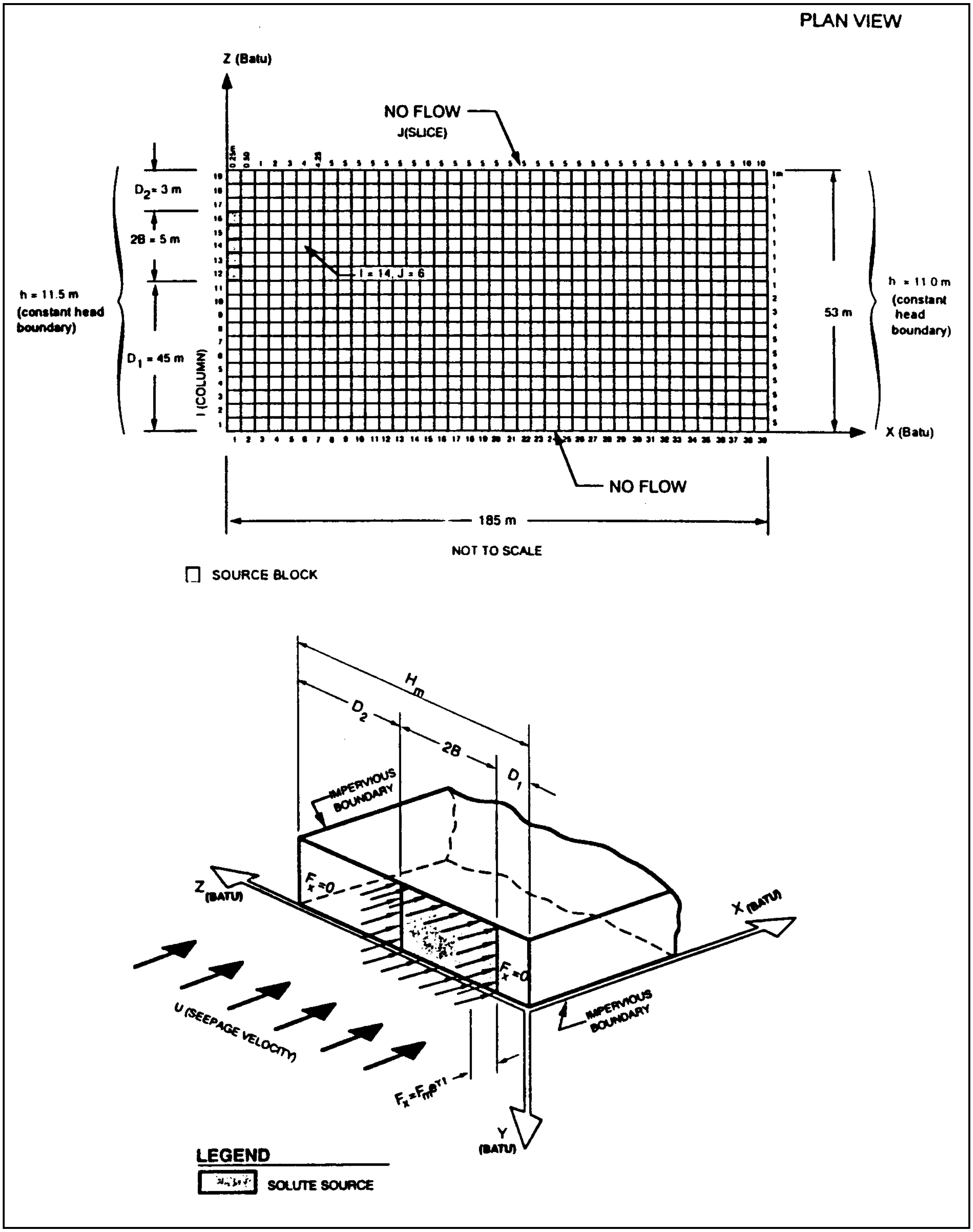


Figure 2.5.1. Model discretization and setup for test FTW-TST 2.5 (from Faust *et al.*, 1993).

control parameters:	SSOR relaxation factor (flow) = 1.80; SSOR relaxation factor (transport) = 1.3; error criterion (flow) = 1.0E-3; error criterion (transport) = 1.0E-5; weighting factor for nonlinear iterations = 0.75; weighting factor for time derivative = 0.5; tolerance for nonlinear iteration=1.0E-3; max. # SSOR iterations (flow) = 30; max. # SSOR iterations (transport) = 35; max. # of nonlinear iterations (flow) = 10; max. # of nonlinear iterations (transport) = 1; upstream weighting in space
iteration performance:	7 nonlinear iterations for flow with diminishing number of SSOR iterations; water balance error (percent) = 1.3E-2; solute balance error (percent) jumps between 1.3E-2 and about 1.0E-7 from step to step (44 time steps)
comments:	documentation cautions for use of quasi-three-dimensional approach in case of significant vertical fluxes through the aquitards

---

IGWMC test #:	FTW-TST-2.7.1
input file name:	ETTRAN0.DAT, ETTRAN7.DAT, ETTRAN14.DAT
IGWMC output file:	ETTRAN0.OUT, ETTRAN7.OUT, ETTRAN14.OUT
code reference:	manual, section 5.5.3, p.174
description:	two-dimensional transient flow and transport in a homogeneous, isotropic unconfined aquifer with depth-limited evapotranspiration or drain-discharge, and well-pumping; an injection well creates solute mass in the model
tested functions:	evapotranspiration as a transport boundary including the evapotranspiration concentration multiplier (ETC) to reflect varying levels of solute uptake by plants
model domain:	a rectangular block containing three aquifers separated by confining layers, bound at one side by a lake and at the other sides and the bottom by impermeable rock; uniform areal recharge in the shallow unconfined aquifer (see Fig. 1.7.1)
grid:	uniform horizontal grid with square cells (7 x 7 cells of 100 x 100 ft)
boundary conditions:	no flow at lateral boundaries and bottom, evapotranspiration along a line of cells (column4), and free surface at top boundary (unconfined); 1 discharging well (node 1,1; 2500 ft^3/day); zero solute flux at all boundaries and 1 injection well (node 7,7) at 100 ft^3/day and 100 ppm; solute outflux through evapotranspiration or internal drains
initial conditions:	10 ft head, zero concentration
parameters:	bottom elevation = -50 ft; storage coefficient = 0.1; hydraulic conductivity = 10 ft/day; maximum ET = 0.2 ft/day; ET extinction depth = 10 ft; ET surface elevation = 10 ft; ETC=1.0, 0.5, and 0.0; drain leakance rate = 0.02 day^{-1}, drain elevation = 0.0 ft
time-stepping:	20 time steps in 365 days with multiplier of 1.2 with an additional refinement of 20 steps in the beginning of the simulation to ensure proper mass balance for FTWORK
benchmark:	intracomparison with drain function (see test FTW-TST-2.7.2)
tests performed by:	problem setup for both codes by code developers; FTWORK code run by IGWMC to check output

type of comparison:	tabular listing and graphic representation of concentration versus time at selected node for both evapotranspiration and drains, and table of concentration versus time for different ETC values
statistics:	not generated
control parameters:	SSOR relaxation factor (flow) = 1.80; SSOR relaxation factor (transport) = 1.2; error criterion (flow) = 1.0E-4; error criterion (transport) = 1.0E-6; weighting factor for nonlinear iterations = 1.5; weighting factor for time derivative = 0.5; tolerance for non-linear iteration=1.0E-4; max. # SSOR iterations (flow) = 50; max. # SSOR iterations (transport) = 30; max. # of nonlinear iterations (flow) = 1; max. # of nonlinear iterations (transport) = 1; upstream weighting in space
iteration performance:	water balance error (percent) varies between 7.6E-1 and 5.0E-3; solute balance error (percent) increases from 1.5E-14 at the start to about 1.0E-5 at the end (320 time steps)

IGWMC test #:	FTW-TST-2.7.2
input file name:	DRTRAN14.DAT
IGWMC output file:	DRTRAN14.OUT
code reference:	manual, section 5.5.3, p.174
description:	drain transport problem to test the evapotranspiration transport function using ETC=1.0; problem set up identical to 2.7.2 with evapotranspiration nodes replaced by drain nodes
tested functions:	see FTW-TST-2.7.1
model domain:	see FTW-TST-2.7.1
grid:	see FTW-TST-2.7.1
boundary conditions:	see FTW-TST-2.7.1
initial conditions:	see FTW-TST-2.7.1
parameters:	see FTW-TST-2.7.1
time-stepping:	see FTW-TST-2.7.1
benchmark:	see FTW-TST-2.7.1
test performed by:	see FTW-TST-2.7.1
type of comparison:	see FTW-TST-2.7.1
statistics:	not performed
control parameters:	SSOR relaxation factor (flow) = 1.80; SSOR relaxation factor (transport) = 1.2; error criterion (flow) = 1.0E-4; error criterion (transport) = 1.0E-6; weighting factor for

	nonlinear iterations = 1.5; weighting factor for time derivative = 0.5; tolerance for non-linear iteration=1.0E-4; max. # SSOR iterations (flow) = 50; max. # SSOR iterations (transport) = 30; max. # of nonlinear iterations (flow) = 1; max. # of nonlinear iterations (transport) = 1; upstream weighting in space
iteration performance:	water balance error (percent) varies between 1.0 and 5.0E-4; solute balance error (percent) increases from 1.0E-14 to about 1.0E-6 in early part and then varies between 1.0E-5 and 1.0E-7 (320 time steps)

IGWMC test #:	FTW-TST-3.1
input file name:	PARA.DAT
IGWMC output file:	PARA.OUT
code reference:	manual, section 5.3, p.148
description:	simulation of steady-state three-dimensional flow in a four-aquifer/three-aquitard system subject to pumping and uniform areal recharge; hydraulic conductivity is homogeneous within each layer but transmissivity varies with layer thickness
tested functions:	automatic parameter estimation for horizontal and vertical hydraulic conductivity and recharge
model domain:	irregularly bounded domain
grid:	30 x 30 uniformly spaced grid with cells of 4000 x 4000 ft; six variable-thickness nodal layers (only two aquitards separately modeled)
boundary conditions:	various 1st, 2nd, and 3rd type boundary conditions
initial conditions:	n.a.
parameters:	see input and output files
time-stepping:	n.a.
benchmark:	manual calibration; applicability demonstration
tests performed by:	problem setup for both codes by code developers; FTWORK code run by IGWMC to check output
type of comparison:	tabular listing of estimates and head residuals for each iteration
statistics:	not generated
comments:	actual calibration of the model took more than 60 runs; example shown in documentation is one of these runs

DOCUMENTATION ERRATA

Program:	FTWORK
Version:	2.8B
Release Date:	3/1993
Custodian:	GeoTrans, Inc., Sterling, Virginia
Prepared by:	Paul K.M. van der Heijde, IGWMC
Date:	May 5, 1995

The following are errors in the documentation and test data sets encountered by IGWMC test running of FTWORK. It should be noted that this list is not complete. Send E-mail or fax if other discrepancies in documentation, coding or test files are encounterd.

Test 4.1.1: typo in column 1 of table 4.1 (page 63, line 4): distance x=360 ft should read x=320 ft

Test 4.1.2: typo in column 1 of table 4.3 (page 69, line 11): distance x=28.00 ft should read x=98.00 ft

Test 4.1.3: documentation lists time maximum as 86,400 seconds instead of 864,000 seconds as used in data file F3.DAT (p. 70, line 21; fig. 4-7 and 4-8 time axis should display from 10^4 - 10^7 seconds for the same curve; legend of fig. 4-9 legend should read time=864,000 seconds; caption of table 4.7 should read time=864,000 seconds)

Test 4.1.4: S_s in figure 4.10 should have as units ft^{-1}

Test 4.2.2: source strength listed in table 4.12 as 704 g/m/d is in actuality in the data files (RUN1A.DAT, RUN2A.DAT, and RUN4A.DAT) set at 7.04 g/m/d; this is calculated as:

$$
\begin{aligned}
Q\,C_0/b &= 0.2\ m^3/d * 1.1792\ kg/m^3 / 33.5\ m \\
&= 0.00704\ kg/m/d \text{ or } 7.04\ g/m/d. \\
Q\,C_0 &= 0.23584\ kg/d
\end{aligned}
$$

The concentration values in tables 4-13, 4-14, and 4-15 are given in kg/m^3 (if multiplyer 1E-03 listed in table headings is used).

Test 4.2.3: Although the documentation lists the same source strength and saturated thickness in Table 4.17 as given in Table 4.12 for test case 4.2.2, in actuality the data files (HI3.DAT and HI3_RADN.DAT) contain a different value for the concentration: C_0=0.11792 kg/m^3. This results in calculated concentrations which are a factor 10 lower than listed in Tables 4-18, 4-19 and 4-20, assuming that the concentrations listed in these tables should be multiplied by a factor 1E-03 as is the case in the tables for Problem 4.2.3. Furthermore, table headings of Table 4-18, 4-19, and 4-20 should include the concentration multiplication factor 1E-03

The data sets HI3.DAT and HI3_RADN.DAT have ITIME in card 8A set as 1 (=seconds); this should be 4 (days); this does not affect numerical results, only time unit display

---

REFERENCES FOR APPENDIX E

Andersen. P.F. 1993. A Manual of Instructional problems for the U.S.G.S. MODFLOW Model. EPA/600/R-93/010, Office of Research and Development, U.S. Environmental Protection Agency, Washington, D.C.

Andersen, P.F., C.R. Faust, and J.W. Mercer. 1984. Analysis of Conceptual Designs for Remedial Measures at Lapper Landfill, New York. Ground Water, Vol. 22(2), pp. 176-190.

Babu, V. 1992. A Generalized Two-Dimensional Analytical Solute Transport Modeling Bounded Media for Flux-Type Finite Multiple Sources. Water Resources Res., Vol.25(6), pp. 1125-1132.

Bear, J. 1979. Hydraulics of Groundwater. McGraw-Hill Comp., New York.

Faust, C.R., P.N. Sims, C.P. Spalding, P.F. Andersen, B.H. Lester, M.G. Shupe, and A. Harrover. 1993. FTWORK: Groundwater Flow and Solute Transport in Three Dimensions; Documentation Version 2.8. GeoTrans, Inc., Sterling, Virginia.

Hantush, M.S. and C.E. Jacob. 1955. Non-Steady Radial Flow in an Infinite Leaky Aquifer. Trans. Am. Geoph. Un., Vol. 36(1), pp. 95-100.

McDonald, M.G., and A.W. Harbaugh. 1984. A Modular Three-Dimensional Finite Difference Ground-Water Flow Model. U.S. Geological Survey Open File report 83-875, Reston, Virginia.

Srinivasen, P. and J.W. Mercer. 1987. BIO1D - One-Dimensional Model for Comparison of Biodegradation and Adsorption Processes in Contaminant Transport - Documentation, GeoTrans,Inc., Sterling, Virginia.

Theis, C.V. 1935. The Relation Between Lowering of the Piezometric Surface and the Rate and Duration of Discharge of a Well Using Ground Water Storage. Trans. A. Geophys. Un., 16th Annual Meeting, Pt.2, pp. 519-524.

Trescott, P.C., G.F. Pinder, and S.P. Larson. 1976. Finite-Difference Model for aquifer simulation in Two Dimesnions with Numerical Experiments. U.S. Geological Survey Techniques of Water Resources Investigations, Book 7, Chapter C1., Reston, Virginia.

Wilson, J.L., and P.J. Miller. 1978. Two-Dimensional Plume in Uniform Ground-Water Flow. Journ. Hydraulic Div., ASCE, Vol. 104(HY): 503-514.

APPENDIX F

SELECTED ANALYTICAL SOLUTIONS PROGRAMMED WITH MATHCAD® FOR WINDOWS

Version 5.0 Plus

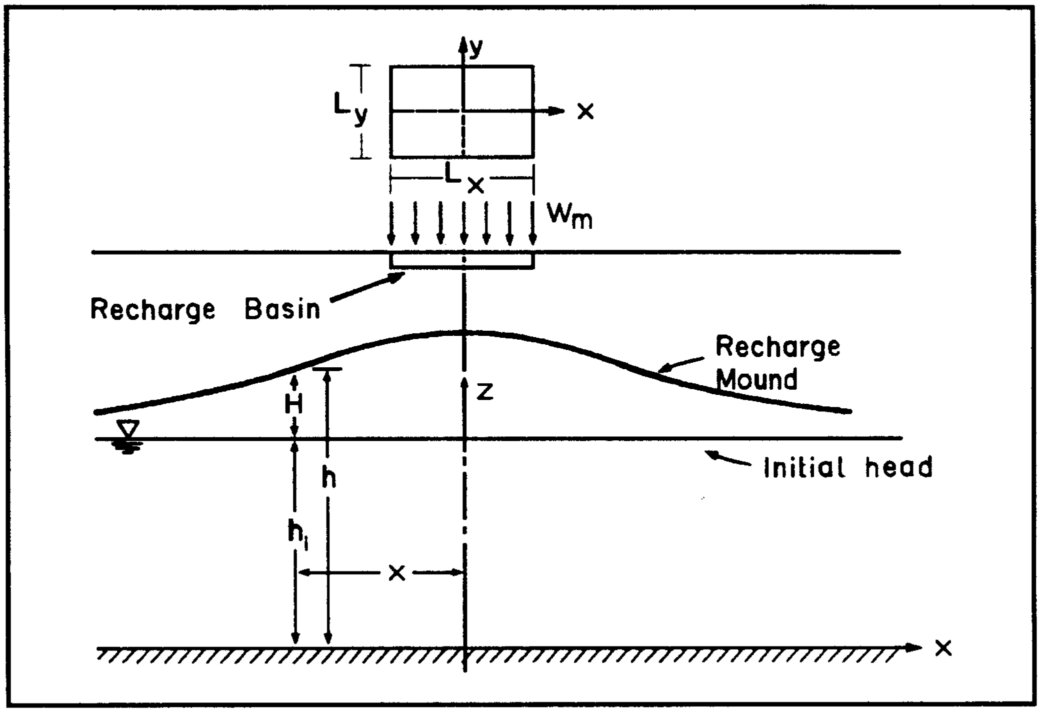


Figure F-1. Definition sketch for mounding due to recharge in a rectangular area.

MND-EPA1 **Analytical solution for transient mounding in a confined aquifer or an unconfined aquifer with constant thickness resulting from recharge in a rectangular area (regular spacing).**

DESCRIPTION:

This model is based on the linearized Boussinesq equation for two-dimensional horizontal flow in a homogeneous, isotropic unconfined aquifer using the Glover (1960) solution for mounding resulting from a continuous recharge from a rectangular surface basin. It uses the Hantush (1967) method of linearizing transmissivity to include the effects of mounding on the average saturated thickness at the point of interest. The governing equation is formulated in an orthogonal coordinate system with its origin in the center of the recharge area. The aquifer is infinite in areal extent. Before recharge starts, the aquifer is at rest at $h=h_i$. Once recharge is initiated, the aquifer is under the influence of an uniform recharge rate W_m applied to the rectangular recharge basin at the surface. The base of the aquifer is taken as the reference level for hydraulic head.

DEFINITION OF VARIABLES:

$K_h := 10$	**hydraulic conductivity [ft/d]**
$h_i := 200$	**initial hydraulic head [ft]**
$W_m := 1$	**recharge rate [ft/d]**
$S_y := .2$	**storage coefficient**
$L_x := 500$	**width of recharge area in X-direction [ft]**
$L_y := 500$	**width of recharge area in Y-direction [ft]**

COMPUTATIONAL DATA:

calculation time: $T_c := 121$ **days** **tolerance:** $TOL = 1 \cdot 10^{-5}$

calculation distance from center of mound [ft]

X-direction:

$Jtotal := 41$ **number of calculation points on X-axis** $j := 0 .. Jtotal - 1$

$stepx := 125$ $startx := 0$ $g := 0 .. \frac{Jtotal - 1}{2}$

$X_g := startx + g \cdot stepx \cdot 2$ $x := startx, startx + stepx .. startx + (Jtotal - 1) \cdot stepx$

Y-direction:

$Ktotal := 1$ **number of calculation points on Y-axis** $k := 0 .. Ktotal - 1$

$stepy := 10$ $starty := 0$

$Y_k := starty + k \cdot stepy$ $y := starty, starty + stepy .. starty + (Ktotal - 1) \cdot stepy$

OPERATIONAL EXPRESSIONS AND EQUATIONS:

Mounding solution for rectangular basin according to Glover (1960) and Hantush (1967) including saturated thickness correction as discussed by Warner et al. (1989):

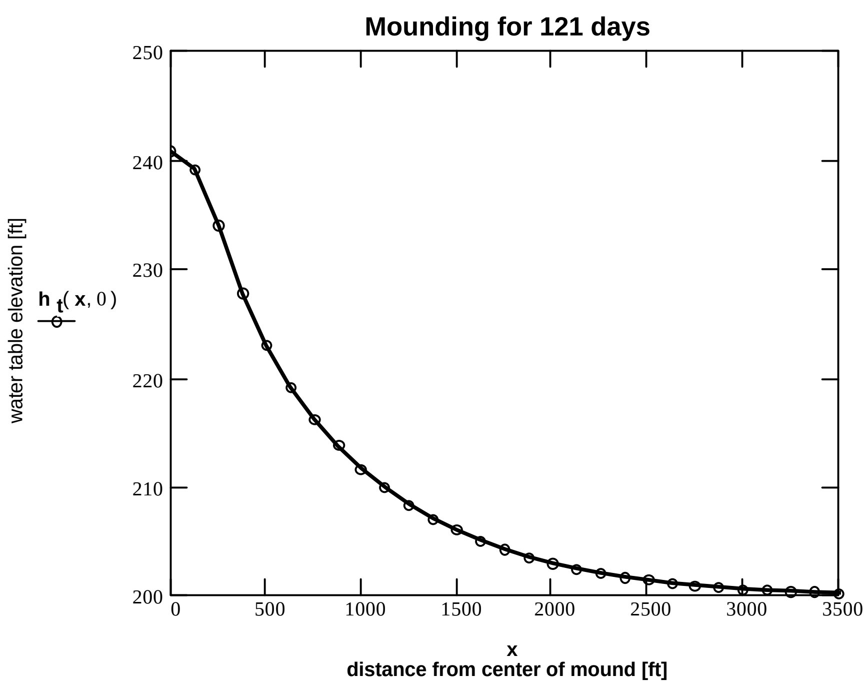
$$\alpha := \frac{K_h \cdot h_i}{S_y} \qquad \beta := \frac{W_m \cdot h_i}{2 \cdot S_y}$$

$$z(x,y) := \beta \cdot \int_0^{T_c} \left(\mathrm{erf}\left(\frac{\frac{L_x}{2} + x}{\sqrt{4 \cdot \alpha \cdot \tau}} \right) + \mathrm{erf}\left(\frac{\frac{L_x}{2} - x}{\sqrt{4 \cdot \alpha \cdot \tau}} \right) \right) \cdot \left(\mathrm{erf}\left(\frac{\frac{L_y}{2} + y}{\sqrt{4 \cdot \alpha \cdot \tau}} \right) + \mathrm{erf}\left(\frac{\frac{L_y}{2} - y}{\sqrt{4 \cdot \alpha \cdot \tau}} \right) \right) d\tau$$

Mounding above initial water table [ft]: $h_m(x,y) := -h_i + \sqrt{h_i^2 + z(x,y)}$

Final position of water table [ft]: $h_t(x,y) := h_i + h_m(x,y)$

RESULTS:



For $T_c = 121$ days

distance from center of recharge area [ft]	mounding [ft]	water table elevation [ft]
X_g	$h_m(X_g, 0)$	$h_t(X_g, 0)$
0	40.774	240.774
250	33.955	233.955
500	22.928	222.928
750	16.195	216.195
1000	11.624	211.624
1250	8.360	208.360
1500	5.979	205.979
1750	4.234	204.234
2000	2.961	202.961
2250	2.041	202.041
2500	1.384	201.384
2750	0.923	200.923
3000	0.604	200.604
3250	0.388	200.388
3500	0.245	200.245
3750	0.151	200.151
4000	0.091	200.091
4250	0.054	200.054
4500	0.031	200.031
4750	0.018	200.018
5000	0.010	200.010

References:

Glover, R.E. 1960. Mathematical Derivations as Pertain to Groundwater Recharge. Agric. Res. Service, USDA, Ft. Collins, Colorado.

Hantush, M.S. 1967. Growth and Decay of Groundwater Mounds in Response to Uniform Percolation. Water Resources Research, Vol. 3, No.1, pp. 227-234.

Warner, J.W., D. Molden, M. Chehata, and D.K. Sunada. 1989. Mathematical Analysis of Artificial Recharge from Basins. Water Resources Bulletin, Vol. 25(2), pp. 4-11.

MND-EPA2: **Analytical solution for transient mounding in a confined aquifer or an unconfined aquifer with constant thickness resulting from recharge in a rectangular area (irregular spacing).**

DESCRIPTION:

This model is based on the linearized Boussinesq equation for two-dimensional horizontal flow in a homogeneous, isotropic unconfined aquifer using the Glover (1960) solution for mounding resulting from a continuous recharge from a rectangular surface basin. It uses Hantush (1967) method of linearizing transmissivity to include the effects of mounding on the average saturated thickness at the point of interest. The governing equation is formulated in an orthogonal coordinate system with its origin in the center of the recharge area. The aquifer is infinite in areal extent. Before recharge starts, the aquifer is at rest at $h=h_i$. Once recharge is initiated, the aquifer is under the influence of an uniform recharge rate W_m applied to the rectangular recharge area at the surface. The base of the aquifer is taken as the reference level for hydraulic head.

DEFINITION OF VARIABLES (unconfined aquifer):

$K_h := 10$	**hydraulic conductivity [ft/d]**
$h_i := 200$	**initial hydraulic head [ft]**
$W_m := 1$	**recharge rate [ft/d]**
$S_y := .2$	**storage coefficient**
$L_x := 500$	**width of recharge area in X-direction [ft]**
$L_y := 500$	**width of recharge area in Y-direction [ft]**

COMPUTATIONAL DATA:

calculation time: $T_c := 121$ **days** **tolerance:** $TOL = 1\cdot 10^{-5}$

<u>calculation distance from center of mound</u> [ft]:

X-direction:

number of calculation points on X-axis: $Xtot1 := 31$ $Xtot2 := 35$ $Xtot := Xtot1 + Xtot2$

define regular spaced points along x-axis:

$stepx := 125$ $startx := 0$ $j1 := 0 .. Xtot1 - 1$

$X1_{j1} := startx + j1\cdot stepx\cdot 2$

read-in additional irregular spaced points from file:

$j2 := 0 .. Xtot2 - 1$ $X2_{j2} := READ(MND\_EPA2)$ **based on one-eighth sector of regular finite difference grid**

Y-direction:

$Ktotal := 1$ **number of calculation points on Y-axis** $k := 0 .. Ktotal - 1$

$stepy := 10$ $starty := 0$

$Y_k := starty + k\cdot stepy$ $y := starty, starty + stepy .. starty + (Ktotal - 1)\cdot stepy$

OPERATION EXPRESSIONS AND EQUATIONS:

Mounding solution for rectangular basin according to Hantush (1967) including saturated thickness correction as discussed by Warner et al. (1989):

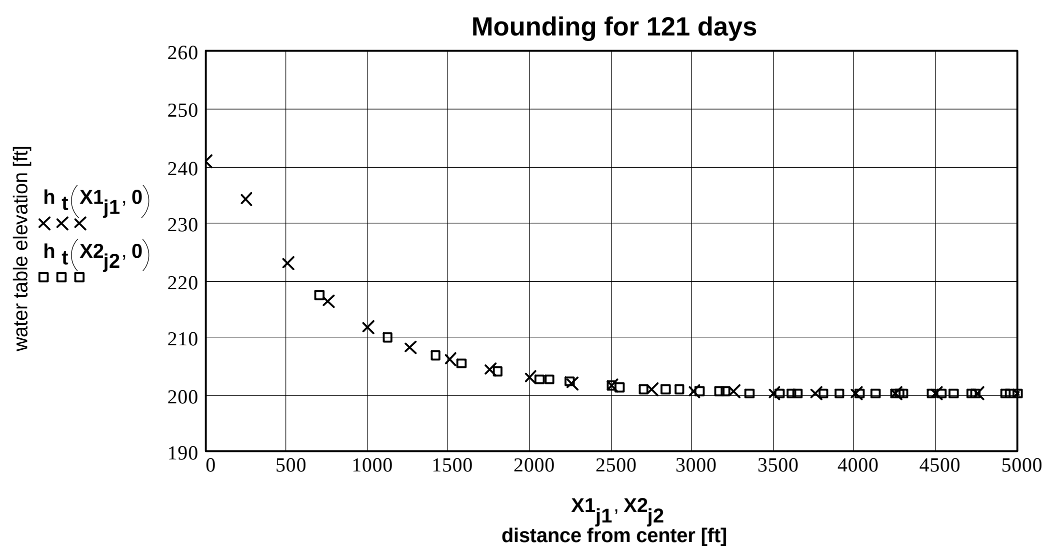
$$\alpha := \frac{K_h \cdot h_i}{S_y} \qquad \beta := \frac{W_m \cdot h_i}{2 \cdot S_y}$$

$$z(x,y) := \beta \cdot \int_0^{T_c} \left(\mathrm{erf}\left(\frac{\frac{L_x}{2}+x}{\sqrt{4\cdot\alpha\cdot\tau}}\right) + \mathrm{erf}\left(\frac{\frac{L_x}{2}-x}{\sqrt{4\cdot\alpha\cdot\tau}}\right)\right) \cdot \left(\mathrm{erf}\left(\frac{\frac{L_y}{2}+y}{\sqrt{4\cdot\alpha\cdot\tau}}\right) + \mathrm{erf}\left(\frac{\frac{L_y}{2}-y}{\sqrt{4\cdot\alpha\cdot\tau}}\right)\right) d\tau$$

Mounding above initial water table [ft]: $h_m(x,y) := -h_i + \sqrt{h_i^2 + z(x,y)}$

Final position of water table [ft]: $h_t(x,y) := h_i + h_m(x,y)$

RESULTS (combination of regular and irregular spaced points):



For $T_c = 121$ days

distance from center of recharge area [ft]	mounding [ft]	water table elevation [ft]	distance from center of recharge area [ft]	mounding [ft]	water table elevation [ft]
$X1_{j1}$	$h_m(X1_{j1}, 0)$	$h_t(X1_{j1}, 0)$	$X2_{j2}$	$h_m(X2_{j2}, 0)$	$h_t(X2_{j2}, 0)$
0	40.77	240.77	707	17.16	217.16
250	33.96	233.96	1118	9.95	209.95
500	22.93	222.93	1414	6.72	206.72
750	16.19	216.19	1581	5.35	205.35
1000	11.62	211.62	1803	3.93	203.93
1250	8.36	208.36	2062	2.7	202.7
1500	5.98	205.98	2121	2.48	202.48
1750	4.23	204.23	2236	2.08	202.08
2000	2.96	202.96	2500	1.38	201.38
2250	2.04	202.04	2550	1.28	201.28
2500	1.38	201.38	2693	1.01	201.01
2750	0.92	200.92	2828	0.81	200.81
3000	0.6	200.6	2915	0.7	200.7
3250	0.39	200.39	3041	0.56	200.56
3500	0.24	200.24	3162	0.45	200.45
3750	0.15	200.15	3202	0.42	200.42
4000	0.09	200.09	3354	0.32	200.32
4250	0.05	200.05	3536	0.23	200.23
4500	0.03	200.03	3606	0.2	200.2
4750	0.02	200.02	3640	0.19	200.19
5000	0.01	200.01	3808	0.13	200.13
5250	0.01	200.01	3905	0.11	200.11
5500	0	200	4031	0.09	200.09
5750	0	200	4123	0.07	200.07
6000	0	200	4243	0.05	200.05
6250	0	200	4272	0.05	200.05
6500	0	200	4301	0.05	200.05
6750	0	200	4472	0.03	200.03
7000	0	200	4528	0.03	200.03
7250	0	200	4610	0.02	200.02
7500	0	200	4717	0.02	200.02
			4743	0.02	200.02
			4924	0.01	200.01
			4950	0.01	200.01
			5000	0.01	200.01

References:

Glover, R.E. 1960. Mathematical Derivations as Pertain to Groundwater Recharge. Agric. Res. Service, USDA, Ft. Collins, Colorado.

Hantush, M.S. 1967. Growth and Decay of Groundwater Mounds in Response to Uniform Percolation. Water Resources Research, Vol. 3, No.1, pp. 227-234.

Warner, J.W., D. Molden, M. Chehata, and D.K. Sunada. 1989. Mathematical Analysis of Artificial Recharge from Basins. Water Resources Bulletin, Vol. 25(2), pp. 4-11.

MND-EPA3: **Analytical solution for transient mounding in a confined aquifer or an unconfined aquifer with constant thickness resulting from recharge in a rectangular area (irregular spacing).**

DESCRIPTION:

This model is based on the linearized Boussinesq equation for two-dimensional horizontal flow in a homogeneous, isotropic unconfined aquifer using the Glover (1960) solution for mounding resulting from a continuous recharge from a rectangular surface basin. It also uses the Hantush (1967) method of linearizing transmissivity (as modified by Warner et al. 1989) to include the effects of mounding on the average saturated thickness at the point of interest. The governing equation is formulated in an orthogonal coordinate system with its origin in the center of the recharge area. The aquifer is infinite in areal extent. Before recharge starts, the aquifer is at rest at $h=h_i$. Once recharge is initiated, the aquifer is under the influence of an uniform recharge rate W_m applied to the rectangular recharge area at the surface. The base of the aquifer is taken as the reference level for hydraulic head.

DEFINITION OF VARIABLES (confined aquifer):

$K_h := 10$	**hydraulic conductivity [ft/d]**
$h_i := 200$	**initial hydraulic head [ft]**
$W_m := 1$	**recharge rate [ft/d]**
$S_y := .001$	**storage coefficient**
$L_x := 500$	**width of recharge area in X-direction [ft]**
$L_y := 500$	**width of recharge area in Y-direction [ft]**

COMPUTATIONAL DATA:

calculation time: $T_c := 21$ **days** **tolerance:** $TOL = 1 \cdot 10^{-5}$

calculation distance from center of mound [ft]:

X-direction:

$Jtotal := 25$ **number of calculation points on X-axis** $j := 0 .. Jtotal - 1$

$x_j := READ(MND\_EPA3)$

Y-direction:

$Ktotal := 1$ **number of calculation points on Y-axis** $k := 0 .. Ktotal - 1$

$stepy := 10$ $starty := 0$

$Y_k := starty + k \cdot stepy$ $y := starty, starty + stepy .. starty + (Ktotal - 1) \cdot stepy$

OPERATION EXPRESSIONS AND EQUATIONS:

Mounding solution for rectangular basin according to Glover (1960):

$$\alpha := \frac{K_h \cdot h_i}{S_y} \qquad \beta := \frac{W_m \cdot h_i}{2 \cdot S_y} \qquad \gamma := \frac{W_m}{4 \cdot S_y}$$

Mounding above initial water table [ft]:

$$h_g(x,y) := \gamma \cdot \int_0^{T_c} \left[\mathrm{erf}\left[\frac{x + \frac{L_x}{2}}{\sqrt{4\cdot\alpha\cdot(T_c - \tau)}}\right] \ldots + \left[-\mathrm{erf}\left[\frac{x - \frac{L_x}{2}}{\sqrt{4\cdot\alpha\cdot(T_c - \tau)}}\right]\right]\right] \cdot \left[\mathrm{erf}\left[\frac{y + \frac{L_y}{2}}{\sqrt{4\cdot\alpha\cdot(T_c - \tau)}}\right] \ldots + \left[-\mathrm{erf}\left[\frac{y - \frac{L_y}{2}}{\sqrt{4\cdot\alpha\cdot(T_c - \tau)}}\right]\right]\right] d\tau$$

Final position of water table [ft]: $H_g(x,y) := h_i + h_g(x,y)$

Mounding solution for rectangular basin according to Hantush (1967) including saturated thickness correction as discused by Warner et al. (1989):

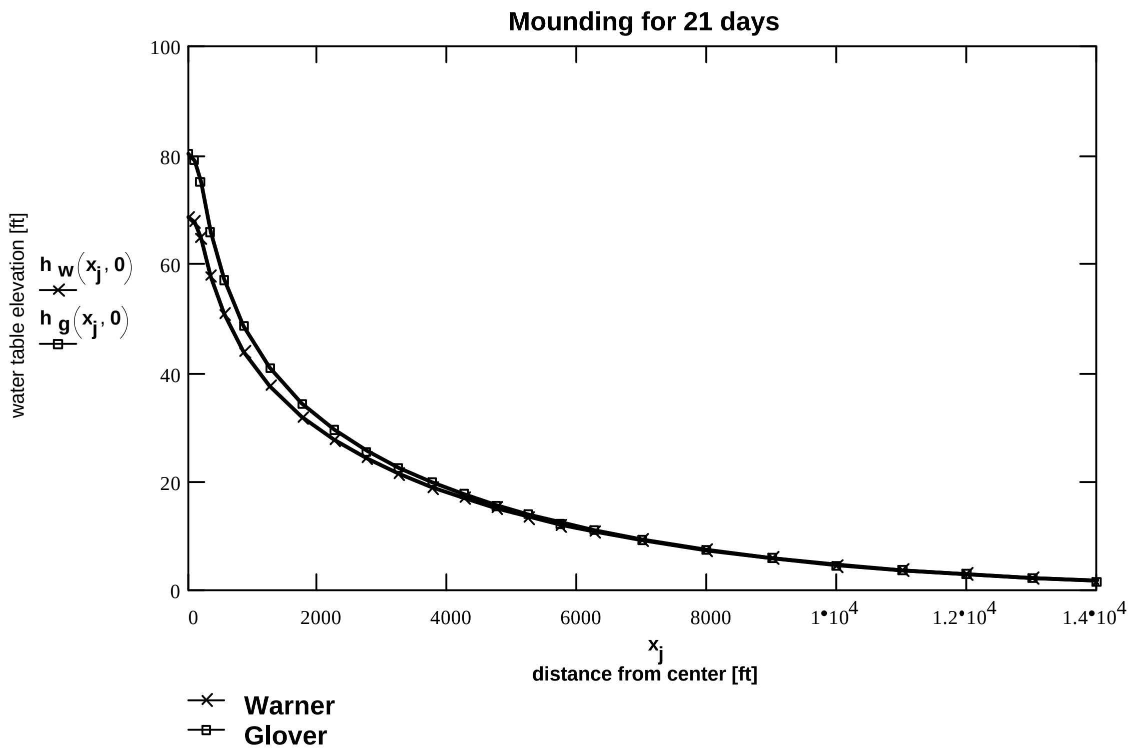
$$Z_w(x,y) := \beta \cdot \int_0^{T_c} \left(\mathrm{erf}\left(\frac{\frac{L_x}{2} + x}{\sqrt{4\cdot\alpha\cdot\tau}}\right) + \mathrm{erf}\left(\frac{\frac{L_x}{2} - x}{\sqrt{4\cdot\alpha\cdot\tau}}\right)\right) \cdot \left(\mathrm{erf}\left(\frac{\frac{L_y}{2} + y}{\sqrt{4\cdot\alpha\cdot\tau}}\right) + \mathrm{erf}\left(\frac{\frac{L_y}{2} - y}{\sqrt{4\cdot\alpha\cdot\tau}}\right)\right) d\tau$$

Mounding above initial water table [ft]: $h_w(x,y) := -h_i + \sqrt{h_i^2 + Z_w(x,y)}$

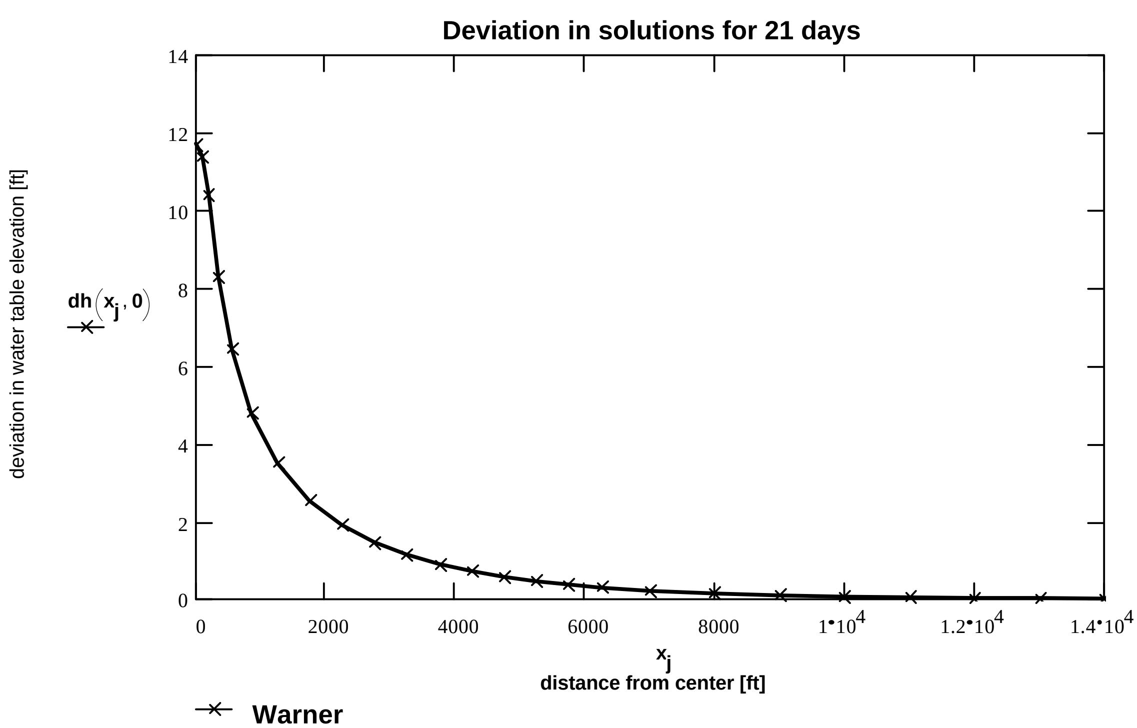
Final position of water table [ft]: $H_w(x,y) := h_i + h_w(x,y)$

Deviation between solutions: $dh(x,y) := h_g(x,y) - h_w(x,y)$

RESULTS:



Mounding for 21 days
100
80
60
40
20
0
water table elevation [ft]
h w(xj, 0)
h g(xj, 0)
0
2000
4000
6000
8000
1•10^4
1.2•10^4
1.4•10^4
xj
distance from center [ft]
Warner
Glover



Deviation in solutions for 21 days
14
12
10
8
6
4
2
0
deviation in water table elevation [ft]
dh(xj, 0)
0
2000
4000
6000
8000
1•10^4
1.2•10^4
1.4•10^4
xj
distance from center [ft]
Warner

For $T_c = 21$ days

mounding [ft] (Warner et al.)	mounding [ft] (Glover)	distance from center of recharge area [ft]	water table elevation [ft] (Warner et al.)	water table elevation [ft] (Glover)
$h_w(x_j, 0)$	$h_g(x_j, 0)$	x_j	$H_w(x_j, 0)$	$H_g(x_j, 0)$
68.43	80.13	0	268.43	280.13
67.49	78.87	100	267.49	278.87
64.55	74.96	200	264.55	274.96
57.54	65.82	350	257.54	265.82
50.67	57.09	550	250.67	257.09
43.72	48.5	850	243.72	248.5
37.39	40.88	1250	237.39	240.88
31.76	34.28	1750	231.76	234.28
27.5	29.39	2250	227.5	229.39
24.1	25.55	2750	224.1	225.55
21.27	22.4	3250	221.27	222.4
18.86	19.75	3750	218.86	219.75
16.77	17.47	4250	216.77	217.47
14.96	15.52	4750	214.96	215.52
13.36	13.81	5250	213.36	213.81
11.94	12.3	5750	211.94	212.3
10.67	10.96	6250	210.67	210.96
9.02	9.22	7000	209.02	209.22
7.19	7.32	8000	207.19	207.32
5.71	5.79	9000	205.71	205.79
4.51	4.56	10000	204.51	204.56
3.55	3.58	11000	203.55	203.58
2.77	2.79	12000	202.77	202.79
2.14	2.15	13000	202.14	202.15
1.65	1.65	14000	201.65	201.65

References:

Glover, R.E. 1960. Mathematical Derivations as Pertain to Groundwater Recharge. Agric. Res. Service, USDA, Ft. Collins, Colorado.

Hantush, M.S. 1967. Growth and Decay of Groundwater Mounds in Response to Uniform Percolation. Water Resources Research, Vol. 3, No.1, pp. 227-234.

Warner, J.W., D. Molden, M. Chehata, and D.K. Sunada. 1989. Mathematical Analysis of Artificial Recharge from Basins. Water Resources Bulletin, Vol. 25(2), pp. 4-11.