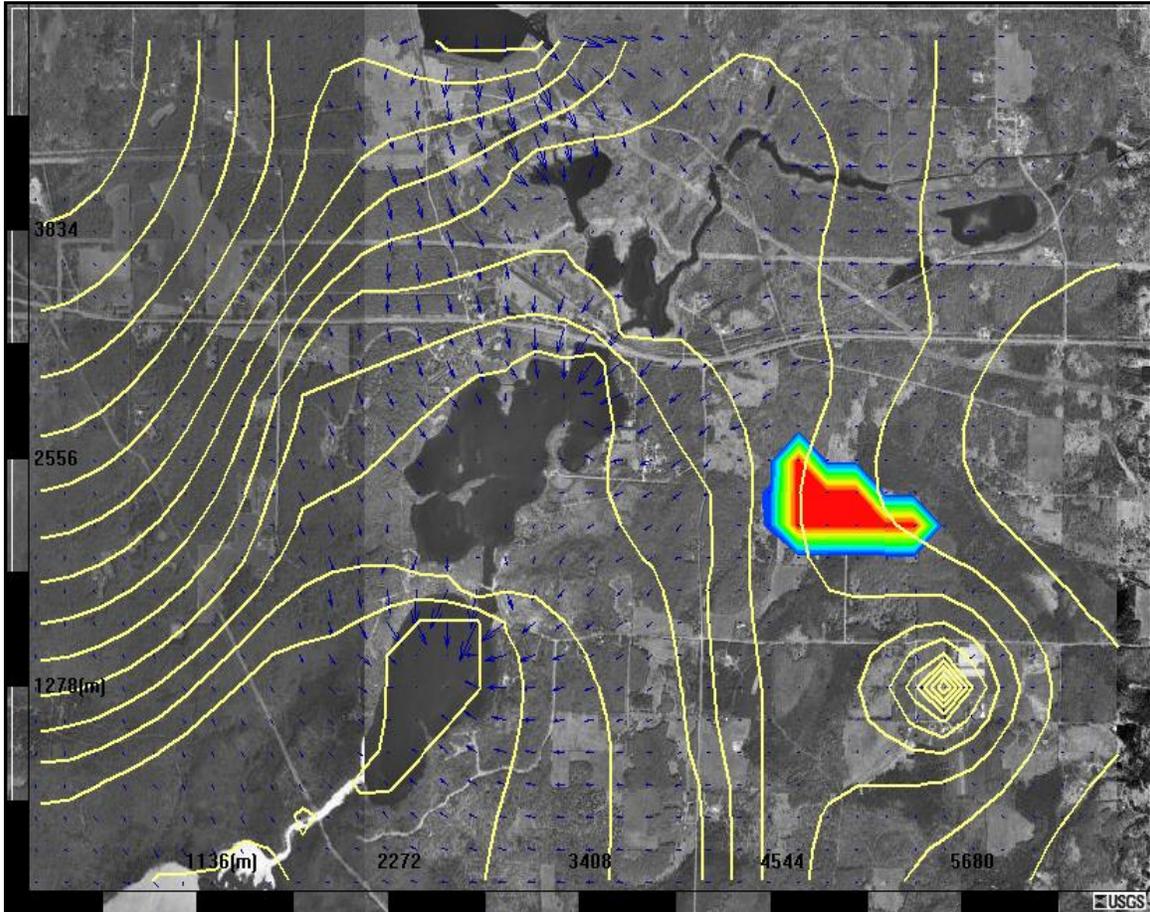


DR. SHUGUANG LI AND ASSOCIATES
INTERACTIVE GROUNDWATER MODELING (IGW)



IGW 3 REFERENCE MANUAL

Dr. Shuguang Li and Associates at Michigan State University



IGW Reference Manual *for Version 3*

Copyright © 2003 by Dr. Shuguang Li and Associates at Michigan State University. All rights reserved.

Dr. Shuguang Li and Associates makes no warranties either express or implied regarding the program IGW 3 and its fitness for any particular purpose or the validity of the information contained in this document.

Authors / Editors:

Huasheng Liao

Kyle Paulson

Dr. Shuguang Li

David Chuen-Fa Ni

Dr. Roger Wallace

Mehmet Oztan

Special Thanks to:

Dr. Qun Liu

DOCUMENT VERSION: **2003-3**

LAST REVISED: **August 15, 2024**

CHAPTER 1: INTRODUCTION

This chapter provides an introduction to the current state of groundwater modeling, the Interactive Groundwater Modeling 3 (IGW 3) software and the new groundwater modeling paradigm it embodies, prospective applications for IGW 3, pointers for using this document and contact information.

1.1 IGW 3 AND THE CURRENT STATE OF GROUNDWATER MODELING

IGW 3 is a two-dimensional, real-time, interactive and visual software system for unified deterministic and stochastic groundwater modeling. It is the culmination of over 4 years of research and testing by a considerable number of dedicated individuals.

IGW 3 has been developed to take advantage of the enormous increases in computational speed and capacity that have led to the increasing importance of computational science and engineering in scientific research, real-world problem solving, and interactive teaching. It is the inability of the current modeling paradigm to take full advantage of these increases that has led Dr. Shuguang Li and his associates to develop a new modeling paradigm, the embodiment of which is the IGW 3 software package.

1.1.1 CURRENT MODELING PARADIGM

The traditional modeling paradigm employed by most groundwater modelers makes use of a sequential scheme based on batch processing and off-line visualization and analysis. A bottleneck occurs because of the inefficient processes employed to move information between various models and to the visualization programs. Modelers repeatedly rely on these transfer processes as they refine the conceptual model and system parameters through an iterative process that is central to the ‘art’ of modeling.

The traditional modeling paradigm employs the following sequence of steps:

- 1) Create, modify, or adopt the code for a conceptual model,
- 2) Assign or modify model stresses, properties, and starting/initial conditions,
- 3) Solve the governing equations over the entire specified time span and store the results,
- 4) Process the results using a visualization package,
- 5) Compare with field data,
- 6) Analyze the results, and
- 7) Repeat the steps incorporating any desired adjustments.

Under the traditional paradigm, modelers typically go off-line to change the conceptual model or the computational scheme. Each change in the model or parameters necessitates that the other steps in the process be repeated. Simple errors such as an incorrect value of a single model parameter may not be detected until many hours have been dedicated to additional and, because of the error, inapplicable data processing and visualization.

It is the inherent inefficiencies of the traditional modeling paradigm that keeps the implementation of large-scale groundwater modeling prohibitively time consuming and expensive and as such restrict our understanding of subsurface flow and contaminant transport due to the fact that so much research cannot be feasibly conducted.

1.1.2 A NEW MODELING PARADIGM

The new modeling paradigm incorporated in IGW 3 allows all of the various models to be coupled and solved in parallel. The computational, modeling, and visualization tasks are integrated. This

allows the user to visualize the flow system's behavior at every time step and evaluate its adequacy. The computations can be suspended, the model parameters adjusted, and the computations restarted with very little effort.

The basic concept is simple. Instead of treating flow and transport separately, they are modeled concurrently. Instead of treating regional-scale modeling, local-scale modeling, and site-scale modeling as different phases in a long sequential process, they are coupled and modeled simultaneously. Instead of relegating the graphical presentation of results and their analysis to the 'post-processing' phase, they are incorporated and updated as the computational results are obtained, after each time step.

The new paradigm is summarized below.

- For time level n , perform the following steps:
- 1) Model the flow,
 - 2) Model the subscale flow (if needed),
 - 3) Track particles (if needed),
 - 4) Model contaminant plume transport (if needed),
 - 5) Model subscale contaminant plume transport (if needed),
 - 6) Process data, perform mass balance and water budget,
 - 7) Visualize the results, and
 - 8) Proceed to time level $n + 1$ and repeat the steps.

This paradigm makes the IGW 3 software package unique and extremely powerful.

1.2 IGW 3 SYNOPSIS

The IGW 3 programming environment, its capabilities, and its applications are discussed in the following subsections.

1.2.1 PROGRAMMING ENVIRONMENT

The IGW 3 software package is comprised of two main programming modules: 1) the Visual Fortran (VF) Dynamically Linked Library (DLL) code, and 2) the Visual Basic (VB) user interface code. In addition, a number of graphical and visualization libraries are bundled into the software.

The VF DLLs are embedded into the VB code and perform the intensive finite-difference and statistical calculations. The VF code is limited to application in the innermost loop of the VB code – the single time step computation. Thus the VB code controls the overall algorithmic logic to control the time steps, nonlinear iterations, iterations between submodels and their respective parent model, intermediate data processing, analysis, integration, and visualization. It is this configuration of the codes that allows for the unprecedented interactivity that is the hallmark of the IGW 3 software package.

1.2.2 CAPABILITIES

IGW 3 provides an interactive, graphical environment for defining the aquifer framework, for inputting parameters, properties and stresses, for changing grid resolution, solvers, numerical schemes and modeling methods, for controlling and managing program execution, and for integrating, overlaying, and visualizing data and results.

The software takes advantage of object-oriented programming and is designed such that the user can, at any time, pause the simulation to edit any aspects of the modeling process and subsequently restart the simulation with the adjusted parameters being incorporated into the model at that exact point. At any time the modeler is able to initiate or stop particle tracking, plume modeling, subscale modeling, and stochastic modeling. After each time step, the user can see updated results presented in a meaningful way.

Specifically, the IGW 3 interactive environment allows an investigator, at any time during the modeling process:

1) To modify the flow model and/or analyze data;

The modeler can input and edit model boundaries, conceptual assumptions, aquifer structures and properties, and stresses. These changes can be imposed over any graphically specified area - independent of the spatial and temporal discretization scheme employed. In addition, data describing any aquifer property or spatial parameter at scattered locations through out the modeled region can be analyzed using advanced regression, interpolation, and statistical simulation techniques.

2) To convert the conceptual model into a numerical model;

The modeler may select and change numerical parameters such as time step and grid spacing, the discretization schemes, solution methods, solver parameters, and spatial interpolation techniques.

3) To initiate particle tracking and/or reactive contaminant transport modeling;

This will permit description of common sources of groundwater contamination, including polluted rivers and lakes, polluted rainfall and artificial recharge, waste-well injections, as well as, instantaneous spills and continuous sources with a time-dependent loading rate.

4) To develop nested submodels of flow and transport;

Modelers may define sub-model regions within a larger model. Boundary conditions for the sub models are extracted from their parent model at every time step and they are solved immediately after the parent solution is obtained and thus essentially run in parallel.

5) To examine the impact of un-modeled small-scale heterogeneity, data limitations, and uncertainty;

The modeler may implement Monte Carlo simulations for the entire parent model. A number of parameters may be modeled as a random field or a random constant, and any temporal stress to be modeled as a 1-D stochastic process characterized by any of a variety of statistical models¹. Flow and transport simulations are automatically “recomputed” for the various property and/or stress realizations. The most recent realizations will be employed as they become available to generate point statistics (e.g., probabilities at any interactively specified monitoring well) and spatial statistics (means, uncertainty, and correlations) that can be mapped and visualized as the simulation proceeds. Best available probabilistic characterizations are presented and recursively improved or updated as the number of realizations increases.

¹ At the time of publishing, only hydraulic conductivity is available to be modeled as a random field. All other parameters are held constant throughout the Monte Carlo simulation.

6) To graphically present model characteristics and results;

The user may opt to present any combination of model inputs and/or outputs in the graphical display (e.g., conductivity, aquifer thickness, recharge, heads, velocities, plume concentration distributions, drawdown distributions, well influence areas, source areas, and wellhead protection areas). In addition, the user may opt to have the software compute and graphically display solute mass, water fluxes and/or water budgets over any specified zones or along any specified “compliance surfaces” or compute and graphically display heads and contaminant concentrations as a function of time at a monitoring well.

and 7) To customize graphical presentations.

The modeler may modify, among other things, the method of presentation, number and order of parameter layers displayed, legends, levels of detail, and display-mode of contours, isosurfaces, velocity vectors, fence diagrams, and x-y plots.

IGW 3 is also very flexible in that it allows the modeler to adjust the degree of steering at any time, from extremely fine to very coarse. Specifically, the software allows the investigator:

1) To visually step through the iterative process of solving a matrix system;

This provides an intuitive feel for the rate of iterative convergence and the performance of the matrix solver. In many cases, this pinpoints visually and directly the cause of many commonly encountered numerical problems (e.g., slow convergence or divergence caused by bad inputs, localized singular characteristics, localized extreme heterogeneity, locally very thin geological layer thickness.).

2) To visually step through the iterative process of solving the non-linear governing groundwater equations;

This is useful for helping a scientist to obtain an intuitive feel for the nonlinear aquifer dynamics or reactive kinetics. This also helps pinpoint directly and visually possible sources of common numerical problems associated with nonlinear iterations (e.g., solution divergence or slow convergence caused highly nonlinear locally desaturated aquifer dynamics).

3) To visually step through the iterations between subscale and parent-scale modeling;

This provides an intuitive feel for the interaction among flow and transport processes at different spatial scales (e.g., among regional scale, local scale, and site-scale; between subsystems) and the effectiveness and significance of the various schemes and techniques used for model downscaling and upscaling.

4) To visually step through time increments;

This is the default steering mode. It allows scientists and engineers to visualize “instantly” the aquifer and plume dynamics in a naturally animated fashion. This also provides flexibility and efficiency in the flow and transport simulations and allows decreasing the time-step size when the simulation becomes difficult (e.g., when a plume moves close to a localized heterogeneity or an area in which a sharp change in the velocity occurs) and increasing it when the difficulty passes.

5) **To visually step through stochastic model realizations;**

This allows scientists and engineers to visualize how heterogeneity translates into uncertainty because of data limitation and plausible realizations of flow and plume dynamics. The on-line recursive analysis dramatically decreases the turnaround time in distributed stochastic modeling. An investigator is able to visualize continuously updated probabilistic characterizations of the groundwater system. Although it often takes thousands of realizations before the final Monte Carlo simulation converges, one can obtain a good qualitative feel of the general statistical behavior of the system (i.e. ensemble means) after 20 to 30 realizations.

and 6) **To run the program in batch mode with interruption and feedback until the end of the overall simulation.**

This is useful towards the end of the overall modeling project when the system is reasonably understood, the conceptual representations, model parameters, modeling scenarios, and numerical discretizations and resolution and solution methods are finalized. In this case, we may be only interested in the ultimate end result (e.g., the plume distribution at the end of the simulation time span). There is no need to interact with the data or model solver, or visualize the temporal or incremental dynamics in between. This is dramatically more efficient than the traditional method of running multiple models under a fragmented batch paradigm.

IGW 3 incorporates an innovative hierarchical and patch dynamics approach that effectively reduces a large-scale complex problem into a sequence of smaller tractable problems with many fewer degrees of freedom. This dramatically decreases computational time and improves the matrix system condition and robustness of the solution process. Practically speaking, this allows for an investigator:

1) **To define a submodel that runs in parallel with its parent model;**

2) **To define a hierarchy of nested models (i.e. a submodel within a submodel);**

Successive sub-models are needed when the area of detailed interest is very small and yet a large modeling domain is needed in order to capture the large-scale regional control. It is particularly useful when the aquifer in question exhibits multiple scales of variations, as is often the case in real-world situations.

and 3) **To define multiple submodels or multiple nested hierarchies in any level of the overall model hierarchy.**

Modeling multiple streams of hierarchical systems is needed when there exist multiple detailed areas of critical importance (e.g., multiple pumping centers in a groundwater basin; multiple contamination sites in an industrial area, multiple hot spots at one site, etc.)

IGW 3 is capable of simulating unsteady flow, reactive transport, and transformation, including diffusion, dispersion, advection, decay, and sorption under linear and nonlinear isotherms in saturated geological media. It also allows simulating non-ideal transport in dual porosity and dual permeability media (e.g., fractured rocks), rate-limited mass transfer, linear and nonlinear reaction kinetics, and flow and transport and uncertainty propagation in “randomly” heterogeneous media. These capabilities are made possible through state-of-the-art computational methodologies and many significant algorithmic innovations such as:

1) An innovative flow solver for general anisotropic aquifer systems;

Most traditional finite difference discretization schemes lead to ill-conditioned matrix systems when the number of degrees of freedom is large and the major orientation of anisotropy deviates significantly from the rectilinear coordinate system. The ill-posed matrices often cause slow convergence, numerical oscillations, physically unrealistic solution, or total solution failure. The IGW 3 flow model adopts an innovative numerical scheme that accounts analytically for the variable orientation of anisotropy. The numerical coefficients, which characterize the performance of the scheme, vary naturally and analytically with the orientation and magnitude of the anisotropy. The result is a significantly improved approach that is robust and more accurate than the traditional finite difference method. The improved scheme is monotonic and always leads to physically meaningful solution.

2) Innovative solvers for advection-dominated transport;

Solute transport studies rely on numerical solutions of the classical advection-diffusion equation. Unfortunately, solutions obtained with traditional finite difference and finite element techniques typically exhibit excessive numerical damping or spurious oscillations when advection dominates. Despite the intensive research and many significant developments over the past decades, there is still no one approach that can provide an accurate, efficient, and robust solution for all flow and transport situations. The IGW 3 transport code adopts a number of techniques to solve the advection-dominated reactive transport equation. These include:

- 1) the Fully Implicit Finite Difference Method,
- 2) the Modified Method of Characteristics, and
- 3) the Random Walk method .

Common in these schemes is that they all take into account the special character (e.g., almost hyperbolic nature) of the governing differential equations in their numerical approximation and provide analytically based “upwinding” that naturally adapts to the magnitude and direction of flow. These numerical schemes are highly accurate with little numerical dispersion or oscillations over a wide range of pecelet numbers.

3) An innovative scheme for general anisotropic dispersion;

Traditional finite difference methodologies can lead to significant unphysical oscillations and negative concentrations when the dispersion is strongly anisotropic and the principal direction of anisotropy deviates from the grid orientation (as is almost always the case in reality). The IGW 3 transport code adopts an improved methodology for approximating anisotropic dispersion in general non-uniform flows. The new approach eliminates the numerical difficulty associated with the cross-dispersion by introducing a local analytical transformation. The result is a numerical scheme that is significantly more accurate and robust than the traditional finite difference schemes.

4) **Efficient sparse matrix solvers;**

IGW 3 employs the highly efficient symmetric successive over relaxation method as its matrix solver. In almost any case it is able to provide, a converged solution with a work count that increases almost linearly with the number of degrees of freedom. This solver is especially effective when they are applied with the new flow and transport schemes and under the new modeling paradigm. The new schemes guarantee that the matrix systems produced are diagonally dominant and positive-definite and the new paradigm allows modeling local sharp gradients and “hotspots” in high resolution without significantly increasing the overall number of nodes. This incremental strategy makes it possible to obtain a converged solution even under most numerically difficult conditions

5) **Advanced interpolation and geostatistical simulation techniques;**

Spatial interpolation is critically important for groundwater modeling since the subsurface environment is inherently heterogeneous and yet available data are often very limited and sparsely scattered. Different techniques may lead to significantly different interpolations and different parameters may require different interpolation techniques. The IGW 3 model code adopts a set of advanced interpolation, regression, and statistical interpolation and simulation techniques that can adapt to a wide variety of subsurface conditions and parameter attributes. The statistical interpolation techniques used include ordinary and universal Kriging and inverse distance weighting. The simulation techniques used include multi-gaussian, simulated annealing, P-field simulation, and various indicator-based simulations.

and 6) **Interblock Interpolation techniques.**

Inter-grid block interpolation is important for heterogeneous flow and transport modeling, especially when the heterogeneity is strong or the small-scale variability is explicitly modeled/resolved as in stochastic simulations. The IGW 3 model code currently uses harmonic averaging.

IGW 3 employs the advanced and efficient OpenGL rendering library and its Visual Tool Kit (VTK) as the main graphics engine. These components allow for on the fly, integrated visualizations. OpenGL is an industry standard in computer graphics, image processing, and visualization and the VTK adds additional support for such algorithms as scalar, vector, tensor, texture, and volumetric methods. Advanced modeling techniques such as implicit modeling, polygon reduction, mesh smoothing, local cutting, slicing, contouring, and fence diagrams are supported in the VTK.

1.2.3 PROSPECTIVE APPLICATIONS OF IGW 3

The IGW 3 software package is relevant and applicable to a number of disciplines within the broad field of groundwater modeling.

RESEARCH

The proposed software environment can be used as an on-line numerical research laboratory. It provides a new and unique way for conducting groundwater investigation and experimentation. Traditional offline modeling and visualization systems work like taking pictures with film. The photographer (scientist or engineer) repeatedly arranges his subjects (conceptual features), releases the shutter (runs the model) until his 24 exposures (the entire batch simulation) are complete. He or she takes his film (saved data) to the processor (a visualization package), waits for the pictures to be developed and then looks at the pictures. Based on the photographs, he or she repeats the

process after rearranging the subjects, lighting and other imaging parameters. The proposed real-time software environment works more like a video camera. The user arranges his subjects while watching the monitor to see the results. Changing the subjects or adjusting the light, immediately changes the image on the monitor. The user gradually changes parameters such as lighting while monitoring the results on the fly.

The software's ability to provide real-time modeling, real-time analysis, real-time visualization, and real-time presentation and the fact that the modeling environment is configurable on line virtually at any time (with respect to, e.g., the aquifer framework, the parameters and stresses and their variability, the numerical discretization, the interpolation schemes, the matrix solvers, and the ways of visualization) makes it an ideal tool for exploratory research, scientific discovery, conceptual modeling, hypothesis testing, trial and error, and process understanding. Within the new environment, a researcher can create visually an aquifer of desired configuration, characteristics, and properties, interactively apply desired stresses, and then immediately see the effect and investigate, analyze, and visualize the processes of flow and contaminant transport and transformation.

The proposed on-line environment is ideally suited for subsurface research. It transforms the "dynamics" and "time scale" of model-based simulation. The new paradigm allows scientists' thought processes to progress naturally and intuitively with the desired information visualized at the instant it is required. It enables them to visualize their thinking and significantly accelerate the interpretation process and better understand the subsurface. Being able to watch natural subsurface processes evolve over time and visualize instantaneously the complex interrelationships among hydrological and environmental variables under meaningful contexts sparks pivotal insights, giving rise to an intuitive grasp of the hydrogeological and biochemical processes that can't be readily obtained otherwise. The new real-time tool provides the much-needed direct link between ideas/hypothesis and significance/implications. It enables critical questions – known and unknown – to be answered quickly and allows the investigator to rapidly move from the inception of a concept to the testing of that concept. This new real-time software provides an environment that gives a real sense of continuous exploration and allows scientists singularly focus on the processes, the problems, the ideas, and the interpretations, pursue "every" lead without breaking his/her train of thoughts. It allows scientists to work much more in "environmental science and interpretive space" rather than "computer science and debugging space". The new technology offers a method of seeing the unseen and understanding the invisible.

PROFESSIONAL PRACTICE AND OUTREACH

On a more practical level, IGW 3 dramatically improves the productivity of groundwater site investigations. It changes the problem-solving role of engineers and professionals in large-scale modeling from heavily physical to cognitive and decision making. It represents an enabling technology. It creates new possibilities. The seamless real-time integration, real-time visual interaction, and the real-time processing capability allow a user to focus on the critical modeling issues, quickly and iteratively examine conceptual approximations, test modeling assumptions, identify dominant processes, evaluate data worth and sensitivity, and calibrate and validate the numerical representation.

IGW 3 also provides an innovative and highly effective platform for professional communication and for facilitating community-based environmental protection related to groundwater resources. It broadens opportunities and enables the informed participation of citizens and improves interactions between government institutions, their constituents, and consultants. The benefits to the stakeholders from the real-time simulation technology include the following:

- 1) Site planners, managers, and regulators could experience the impact and effectiveness of management, sampling, and cleanup scenarios to improve policy-making decisions; They can become much more effective in engaging the general public and informing high-level

decision makers about the implications of the fate and transport of contamination and the impact on the groundwater environment and the affected communities.

- 2) Consultants can make much more effective use of the subsurface data, design better monitoring network to collect additional data, and characterize more accurately contamination site dynamics at much less cost. They also can more easily communicate a solution, a design, or strategy to their clients;
- 3) The local community could visualize the invisible subsurface and experience and understand the impact of a proposed management and cleanup schemes and pollution control measures in an intuitive, vivid, and interactive way. They can also visualize the potential impact of their own activities on the groundwater environment and their drinking water supply. Thus, they are motivated and empowered to engage in the intricate process of community-based environmental management, planning, protection, and cleanup.
- 4) Policymakers and politicians can use real-time interactive simulation as a public relations effort to reveal future environmental plans related to groundwater resources management and remediation.

Coupled with the information explosion and the widespread popularity of computers, IGW 3 stands to dramatically improve the public involvement in groundwater protection where the 'public' is a major stakeholder.

EDUCATION

The innovative software technology also provides a unique tool for training and education. It provides an environment for active student learning and exploration, learning by doing, by solving problems, by engaging in authentic investigations. The software makes it possible to introduce research and complex problem solving into the classroom in a substantial way and on a routine basis.

IGW 3 provides the first systematic environment of its kind for teaching groundwater flow and pollutant transport. Its unique capability of real-time interactive modeling, real-time visualization, real-time analysis, and real-time presentation makes it ideal for teaching and learning in the classroom. It is particularly effective for implementing hands on, interactive, and problem and project based learning in an action-oriented and student centered curriculum. The software can be used as an interactive electronic "chalkboard" for professors to teach and demonstrate live groundwater flow and contaminant transport, contaminated site characterization and remediation design using vivid real-time simulations. It can also be used as an interactive "notepad" or a virtual testing ground for students to engage in active exploration and collaborative investigation under realistic conditions.

ACTIVE LEARNING

In particular, IGW 3 allows a student to interact with and instantly visualize aquifer flow, well dynamics, groundwater and surface water connections, contaminant advection, diffusion, dispersion, sorption, retardation, and decay under different geological, hydrological, hydraulic, and chemical conditions interactively and graphically specified by students. The software can be used to vividly illustrate and investigate the effect of natural variability, the interaction of different scales of heterogeneities, the interactions among geological, chemical, and hydrological heterogeneity on flow and pollutant migration, and how these heterogeneities and their interactions may significantly complicate groundwater remediation. The software also provides an interactive environment for students to perform statistical data analysis, site characterization, geological mapping, pump test analysis and design, well capture zone design, wellhead protection area delineation, monitoring network and remediation extraction system design under meaningful conditions.

Additionally, the software environment can be used to teach computational mathematics and statistical and probabilistic methods in water resources and environmental engineering. Within the new interactive environment, mathematics becomes concrete and differential equations meaningful. Students can interact and experiment hands-on with the model solvers, algorithms, and solution techniques for a concrete and physically meaningful situation and instantly visualize their practical implications (e.g., the impact of solver selection on the rate of the predicted plume spreading). Students can visualize on-line the process of matrix solution and iterations of nonlinear differential equations. They can compare different methods for solving sparse matrix systems and different discretization schemes for approximating elliptic, hyperbolic, and parabolic partial differential equations. They can visually observe the effect of grid spacing and time steps on the solution accuracy and the effect of numerical dispersion and spurious oscillations. Students can also interactively learn, investigate and visualize statistics and probability and conditional probability within a meaningful engineering context. They can interact with and visualize the techniques of numerical integration (particle tracking), spatial interpolation, statistical regression and interpolation, spatial data analysis, histogram and correlation and variogram modeling, random field generation, conditional geostatistical simulation, Monte Carlo simulation, and conditional Monte Carlo simulation.

COLLABORATIVE LEARNING

IGW 3 is a collaborative work platform. By providing instantaneous feedback, and making a student's thinking explicit, visible, and understandable to all in a naturally expressive manner, the software is ideal for effective interdisciplinary interactions, collaborative learning, communication, for involving others with different skills and cultural backgrounds in sharing information, brainstorming, and developing ideas.

The computer screen can be treated as virtual experimental field site or 'testing ground'. The classroom becomes a knowledge-building learning community. Acting as investigators and working in teams, students confront tangible practical problems - e.g., cleaning up an accidental spill, evaluating the environmental impact of a landfill, developing a wellhead protection program for a municipal well field, conducting a remedial and feasibility study for a hazardous waste site, or providing expert testimony in a legal dispute. Students learn by conducting guided site investigations and solving authentic problems.

Working in groups, they discuss the monitoring issues, report back, present findings, challenge and debate each other, explain their points of view, and search for cleanup strategies that build on the strengths of all the group members. In this setting, the instructor becomes a mentor, a facilitator, a co-learner, and a co-investigator with the student. The instructor moves among groups, directing students' discussions and energies when appropriate. The instructor provides coaching and support. At critical times the instructor teaches students the skills, strategies, and links they need to complete the tasks they define for themselves. Rather than simply lecture, the instructor instead cultivates skills, focuses effort, fosters resourcefulness, and maintains an interactive climate of learning, exploration, and discovery.

INDIVIDUALIZED LEARNING

The interactive groundwater environment can also be used for students to conduct independent projects and facilitate independent and deep thinking and individualized learning. Students have different backgrounds, experiences, abilities, cultural origins, learning styles, family responsibilities and personalities. The software allows maximizing teaching and learning for all students by providing a platform for students to engage in independent site investigation. Students may start an individual project in the classroom and finish the bulk of the investigation at home or wherever a computer is connected to the network. In an independent project, a student works on a site by him/herself to

accomplish learning goals. Individual projects complement the collaborative and have many advantages. They can:

- take into account variations in student learning styles as well as ability, background and cultural origins;
- allow students to go as far as they can at their own pace and at the place they choose;
- provide variable time and flexible schedules that enhances quality and in-depth study;
- provide incentives for self-direction, self-motivation, and self-activity;
- promote independent thinking and reduce reliance upon the instructor; and,
- provide self-motivated learning that may continue throughout life: Slow students are seldom discouraged and the gifted are rarely bored

The proposed curriculum innovation will provide leadership in engineering for utilizing technology and creating new models of learning and teaching environments. Educators who are eager to organize their classrooms so as to guarantee students the kind of quality education that will enable them to reach their full potential as learners and as human beings will find such an action-oriented and student-centered approach a practical and exciting direction in which to move.

1.3 REFERENCE MANUAL INTRODUCTION

This ‘Reference Manual’ is intended to give in-depth information concerning the IGW 3 solver techniques, applicable theory, and mathematical foundation. It describes the VF code only. Information concerning the VB programming techniques can be obtained by contacting Dr. Li or his associates.

For information about the implementation of the software and examples of such implementation please consult the *IGW 3 User’s Manual* and *IGW 3 Tutorials* documents.

It should be noted that some details such as basic numerical methods and control volume formulation that are commonly encountered in the field of groundwater modeling have been omitted in the interest of succinctness. Where information has been omitted, it will be noted and appropriate references identified.

1.3.1 REFERENCE MANUAL UPDATES

It is important to note that the IGW 3 software and the associated documents are undergoing constant revision. Check the website (see **Section 1.4**) often for updates.

1.3.2 REFERENCE MANUAL ACRONYMS / ABBREVIATIONS

There are a great number of acronyms used throughout this text. The acronyms, their meanings, and the section in which they are discussed are presented in **Table 1.3.2-1**.

TABLE 1.3.2-1

Reference Manual acronyms

ACRONYM / ABBREV.	DEFINITION	REFERENCE
IGW	Interactive Groundwater Modeling	-
VF	Visual Fortran	Section 1.2.1
DLL	Dynamically Linked Library	Section 1.2.1
VB	Visual Basic	Section 1.2.1
VTK	Visual Tool Kit	Section 1.2.2
FD	Finite Difference	Section 2.1
CV	Control Volume	Section 2.3.1.1
RCVT	Rotational Control Volume Technique	Section 2.3.2
FIFD	Fully Implicit Finite Difference	Section 3.2
MMOC	Modified Method of Characteristics	Section 3.2
RW	Random Walk	Section 3.2
MC	Monte Carlo	Chapter 4
FFT	Fast Fourier Transform	Section 4.1
PDF	Probability Density Function	Section 4.3
CDF	Cumulative Density Function	Section 4.3
IDW	Inverse Distance Weighting	Section 6.1
SOR	Successive Over Relaxation	Chapter 10

1.4 ADDITIONAL INFORMATION

Additional information concerning IGW 3 can be obtained from the IGW website:

<http://www.egr.msu.edu/~lishug/research/igw/index.htm>

The site contains links for (among other things):

- exploring the capabilities and algorithms associated with the software,
- viewing software demonstrations and associated presentations,
- downloading verification papers,
- obtaining software documentation,
- downloading the software,
- providing feedback,
- accessing the IGW Forum,
- contacting Dr. Li and his associates, and
- acknowledgements and team members.

CHAPTER 2: FLOW SOLVER

The flow solver is the backbone of the IGW 3 software and therefore is discussed first. The following sections discuss various components and features of the flow solver.

2.1 GRID LAYOUT

The IGW 3 finite difference (FD) technique employs a grid that is uniform along each axis. In other words, the grid spacing along the x-axis is constant and the grid spacing along the y-axis is constant, but the two spacings may not necessarily be equivalent to each other. With future software improvements in mind, the code has been written in a very general format that allows for non-uniform grid spacing along each axis (although, as stated previously, the currently implemented technique allows for uniform grid spacing only).

Figure 2.1-1 shows a conceptual example of the IGW 3 grid and lists associated parameters.

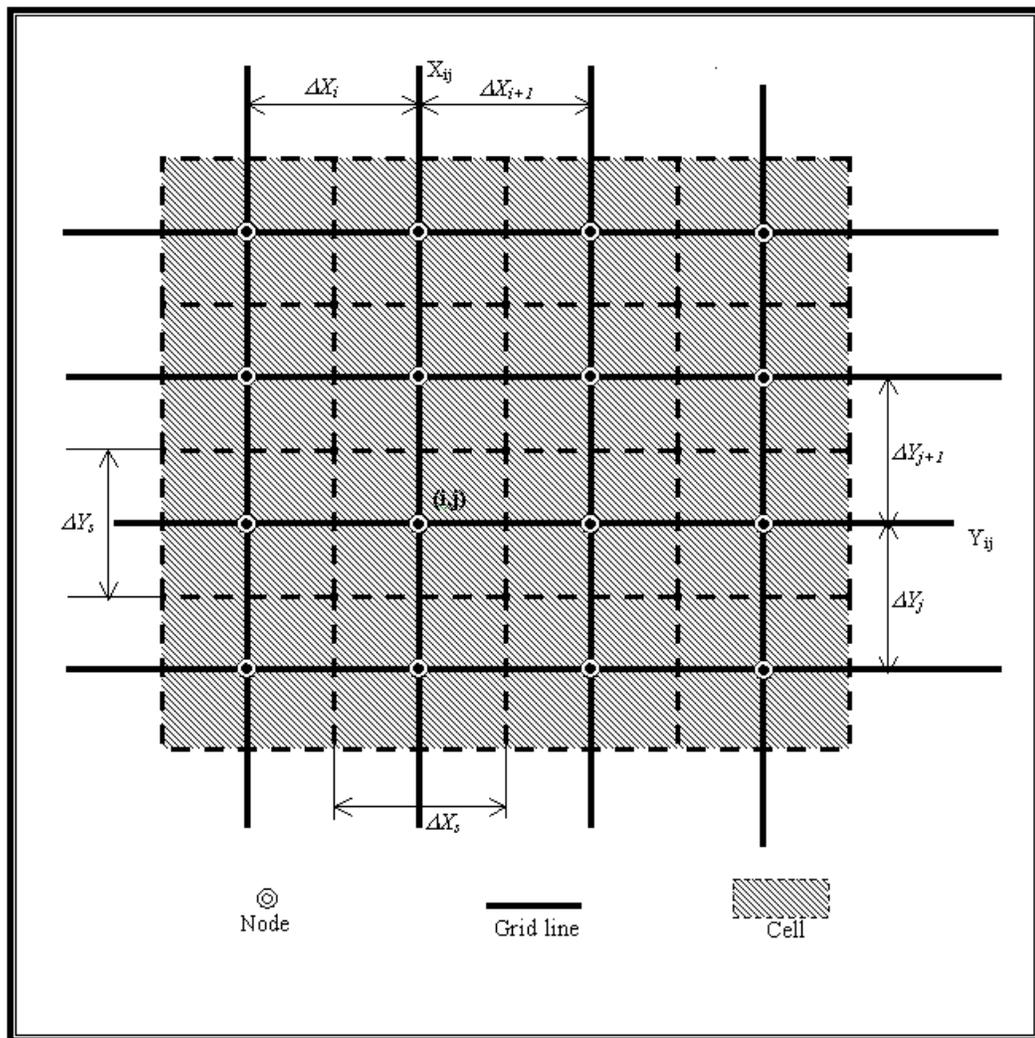


FIGURE 2.1-1
Conceptual example of the IGW 3 grid.

Figure 2.1-2 shows a single cell and the surrounding nodes and gives additional parameter notation.

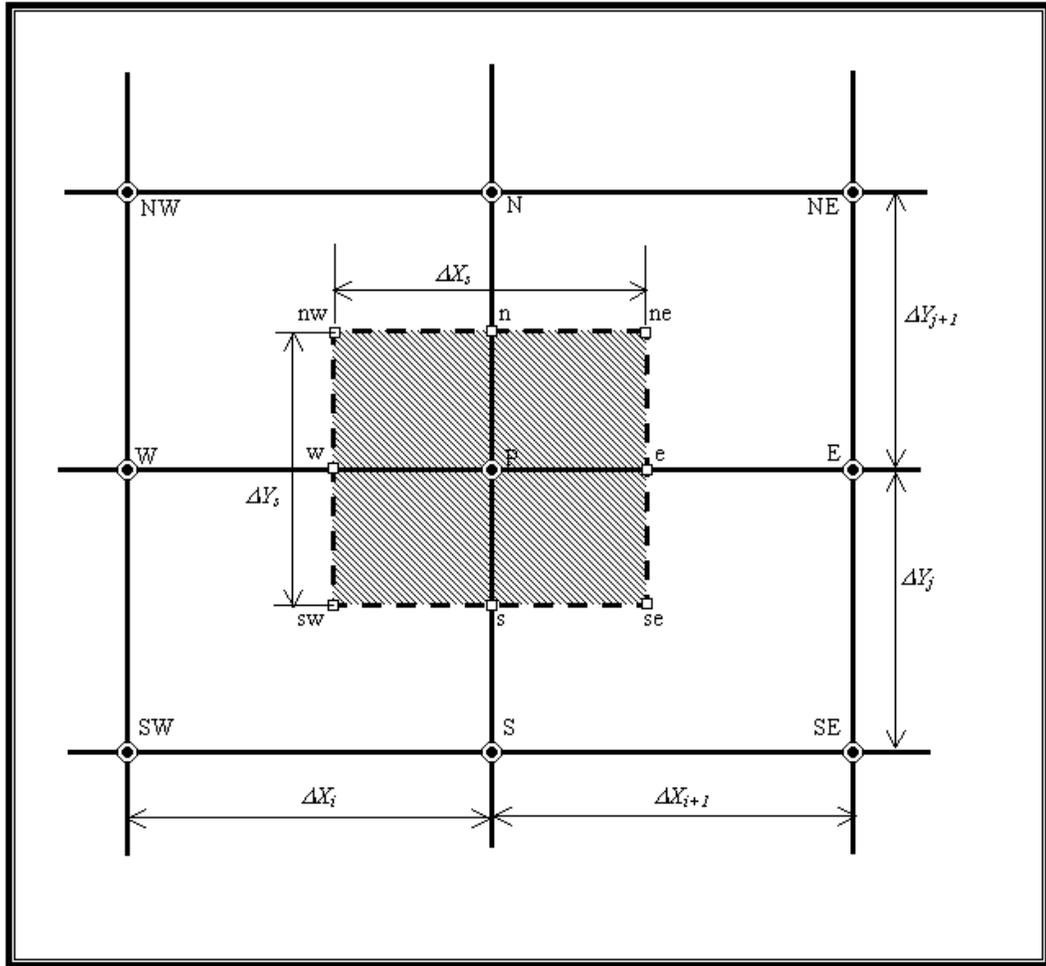


FIGURE 2.1-2
Conceptual example of a single cell and the surrounding nodes.

Table 2.1-1 lists parameters employed in the code and their associated code variables.

TABLE 2.1-1
Parameters and associated code words

Parameter	Notation in Figures 2.1-1 and 2.1-2	Code variable
X-coordinate	X_{ij}	Xmesh(I,J)
Y-coordinate	Y_{ij}	Ymesh(I,J)
X-direction grid spacing	ΔX_i	HX(I)
Y-direction grid spacing	ΔY_i	HY(J)
X-direction control volume spacing	ΔX_s	DXS(I)
Y-direction control volume spacing	ΔY_s	DYS(J)

The user-defined grid spacing determines (through IGW 3 internal calculations) the number and location of the nodes and the size and shape of the associated cells. Spatial parameters are associated with the nodes and the value that exists at a certain node is applicable for the entire area of the cell.

The grid is set up over the entire Working Area (refer to *IGW 3 User's Manual*). The nodes that do not fall within any user-defined feature are referred to as inactive. These cells have zero values for every parameter and although the solver considers them in the solution, they do not affect the modeling output. The inactive cells essentially form a no-flow boundary around the user-defined computational domain.

2.2 GOVERNING EQUATION

The partial differential equation describing flow in porous medial is usually written as

$$S_s \frac{\partial h}{\partial t} = \frac{\partial}{\partial X_i} (K_{ij} \frac{\partial h}{\partial X_j}) + q_s \quad (2.2-1)$$

where S_s = the specific storage of the aquifer materials [L^{-1}],
 h = the hydraulic head [L],
 K_{ij} = the hydraulic conductivity tensor [L/T]
 X_i = the Cartesian coordinate [L], and
 q_s = the source/sink term [T^{-1}].

The depth-averaged form of the equation (where depth is b) can be presented as

$$S \frac{\partial h}{\partial t} = \frac{\partial}{\partial X_i} (T_{ij} \frac{\partial h}{\partial X_j}) + Q_s \quad (2.2-2)$$

where $S = S_s b$ = the storage coefficient [-],
 $T_{ij} = K_{ij} b$ = the transmissivity tensor [L^2/T], and
 $Q_s = q_s b$ = the source/sink term [L/T].

Equation 2.2-2 is the form that is employed in the IGW 3 flow solver. The user inputs the first principal hydraulic conductivity (K'_{ii}), the anisotropy ratio (K'_{ii}/K'_{jj}), and the slope angle (θ). K'_{jj} is determined from the first principal hydraulic conductivity and the anisotropy ratio. The four hydraulic conductivity tensors (with respect to the global model coordinate system) are determined from the principal hydraulic conductivities and slope angle through

$$K_{ii} = \frac{K'_{ii} + K'_{jj}}{2} + \frac{K'_{ii} - K'_{jj}}{2} \cos(2\theta), \quad (2.2-3)$$

$$K_{jj} = \frac{K'_{ii} + K'_{jj}}{2} - \frac{K'_{ii} - K'_{jj}}{2} \cos(2\theta), \quad (2.2-4)$$

and

$$K_{ij} = K_{ji} = \frac{K'_{ii} - K'_{jj}}{2} \sin(2\theta). \quad (2.2-5)$$

The transmissivities used in **Equation 2.1-2** are based on the hydraulic conductivities calculated from **Equations 2.2-3 – 2.2-5**.

Note that a harmonic mean method was used to evaluate K values at the cell faces e, w, n and s.

2.3 DISCRETIZATION OF THE GOVERNING EQUATION

As the above equations indicate, the hydraulic conductivity tensor will have four terms. While most existing flow models ignore the cross terms (K_{ij} and K_{ji}) by assuming that the principal components (K_{ii} and K_{jj}) are aligned with the x- and y-coordinate axes, IGW 3 incorporates them into the solution. The reason that some models ignore them is that when they are considered in a traditional FD scheme, some correlation coefficients in the FD equation are negative and thus imply an inverse relationship between the head in the cell of interest and the surrounding cells. This is not intuitive and is incorrect. IGW 3 alleviates this problem by employing an improved FD scheme that involves a local rotation of the global coordinate system on a cell-by-cell basis. This approach allows all of the conductivity tensors to be considered in the model and thus provides for more accurate solutions.

While the improved FD method approach is the default solver technique in IGW 3, a traditional FD method can be employed for comparison purposes. The traditional FD method is presented in **Section 2.3.1** while the improved approach is presented in **Section 2.3.2**.

The mathematics in the following subsections are presented for an equivalent form of the governing equation (2.2-2) that takes into account the area of the cell

$$S \frac{\partial h}{\partial t} \Delta X_s \Delta Y_s = \frac{\partial}{\partial X_i} \left(T_{ij} \frac{\partial h}{\partial X_j} \right) \Delta X_s \Delta Y_s + Q_s \Delta X_s \Delta Y_s \quad (2.3-1)$$

2.3.1 THE TRADITIONAL FD METHOD

The following subsections present the steps and equations employed when the traditional FD method is implemented.

2.3.1.1 APPROXIMATION OF SPATIAL TERMS

Applying a control volume (CV) technique to each cell (refer to **Figure 2.1-2** for reference), the spatial term on the right-hand side of **Equation 2.1-2** can be approximated by

$$\frac{\partial}{\partial X_i} \left(T_{ij} \frac{\partial h}{\partial X_j} \right) \Delta X_s \Delta Y_s = \Delta X_s \Delta Y_s \frac{J_e - J_w}{\Delta X_s} + \Delta X_s \Delta Y_s \frac{J_n - J_s}{\Delta Y_s} \quad (2.3.1.1-1)$$

where J_e , J_w , J_n , and J_s are the fluxes through the east, west, north, and south faces, respectively and are defined as

$$J_e = T_{xx}^e \frac{h_E - h_P}{\Delta X} + T_{xy}^e \frac{h_{ne} - h_{se}}{\Delta Y_s}, \quad (2.3.1.1-2)$$

$$J_w = T_{xx}^w \frac{h_P - h_W}{\Delta X} + T_{xy}^w \frac{h_{nw} - h_{sw}}{\Delta Y_s}, \quad (2.3.1.1-3)$$

$$J_n = T_{yy}^n \frac{h_N - h_P}{\Delta Y} + T_{yx}^n \frac{h_{ne} - h_{nw}}{\Delta X_s}, \quad (2.3.1.1-4)$$

and

$$J_s = T_{yy}^s \frac{h_p - h_s}{\Delta Y} + T_{yx}^s \frac{h_{se} - h_{sw}}{\Delta X_s}, \quad (2.3.1.1-5)$$

where the superscripts on the transmissivity terms indicate the appropriate cell face.

The non-nodal heads appearing in **Equations 2.3.1.1-2 – 2.3.1.1-5**, h_{ne} , h_{se} , h_{nw} , and h_{sw} are evaluated in terms of the nodal heads using a simple four-point average scheme.

Equations 2.3.1.1-1 – 2.3.1.1-5 are combined and simplified to

$$\begin{aligned} \frac{\partial}{\partial X_i} \left(T_{ij} \frac{\partial h}{\partial X_j} \right) \Delta X_s \Delta Y_s = & a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_{NE} h_{NE} \\ & + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} - a_P h_P \end{aligned} \quad (2.3.1.1-6)$$

where the a_i coefficients are referred to as the coefficients of the discretized matrix and are defined as

$$a_E = \frac{\Delta Y_s T_{xx}^e}{\Delta X} + \frac{T_{yx}^n - T_{yx}^s}{4}, \quad (2.3.1.1-7)$$

$$a_W = \frac{\Delta Y_s T_{xx}^w}{\Delta X} - \frac{T_{yx}^n - T_{yx}^s}{4}, \quad (2.3.1.1-8)$$

$$a_N = \frac{\Delta X_s T_{yy}^n}{\Delta Y} + \frac{T_{xy}^e - T_{xy}^w}{4}, \quad (2.3.1.1-9)$$

$$a_E = \frac{\Delta X_s T_{yy}^x}{\Delta Y} - \frac{T_{xy}^e - T_{xy}^w}{4}, \quad (2.3.1.1-10)$$

$$a_{NE} = \frac{T_{xy}^e + T_{yx}^n}{4}, \quad (2.3.1.1-11)$$

$$a_{NW} = -\frac{T_{xy}^w + T_{yx}^n}{4}, \quad (2.3.1.1-12)$$

$$a_{SE} = -\frac{T_{xy}^e + T_{yx}^s}{4}, \quad (2.3.1.1-13)$$

$$a_{SW} = \frac{T_{xy}^w + T_{yx}^s}{4}, \quad (2.3.1.1-14)$$

and

$$a_p = a_E + a_W + a_N + a_S + a_{NE} + a_{NW} + a_{SE} + a_{SW}. \quad (2.3.1.1-15)$$

The coefficients of the discretized matrix are calculated by the source code subroutine COEFFFLOW. This subroutine uses a derived type variable, CST2, to store the values. **Table 2.3.1-1** lists the variable variants for each individual coefficient.

TABLE 2.3.1-1
The individual coefficient CST2 variable variants

Coefficient	Variable Variant
a_E	CST2(I,J)%SE
a_W	CST2(I,J)%SW
a_N	CST2(I,J)%SN
a_S	CST2(I,J)%SS
a_{NE}	CST2(I,J)%SNE
a_{NW}	CST2(I,J)%SNW
a_{SE}	CST2(I,J)%SSE
a_{SW}	CST2(I,J)%SSW
a_p	CST2(I,J)%SP

2.3.1.2 APPROXIMATION OF TIME DERIVATIVE TERM

The time derivative term on the left-hand side of **Equation 2.3-1** is: a) equal to zero if the model is being solved at steady state, or b) defined by

$$S \frac{\partial h}{\partial t} \Delta X_x \Delta Y_s = S \frac{h^{n+1} - h^n}{\Delta t} \Delta X_x \Delta Y_s = a_p^t h^{n+1} - S_f^t \quad (2.3.1.2-1)$$

when the model is being solved for transient flow. In **Equation 2.3.1.2-1**, the superscript on h indicates the time level, Δt is the time step,

$$a_p^t = \Delta X_s \Delta Y_s \frac{S}{\Delta t}, \quad (2.3.1.2-2)$$

and

$$S_f^t = \Delta X_s \Delta Y_s \frac{S}{\Delta t} h^n. \quad (2.3.1.2-3)$$

2.3.1.3 APPROXIMATION OF SOURCE/SINK TERM

The source/sink term can be expressed as

$$Q_s \Delta X_s \Delta Y_s = a_p^Q h_p + S_f^Q \quad (2.3.1.3-1)$$

where a_p^Q includes any head-dependent sources and sinks and S_f^Q includes any head-independent sources and sinks.

Table 2.3.1.3-1 lists the types of sources and sinks available in IGW 3, their associated head-dependent and head-independent components, and any conditions that determine when the source/sink is active.

TABLE 2.3.1.3-1
Types of source/sinks in IGW 3 and associated properties

Source/Sink	Head Dependent Component	Head Independent Component	Condition
Well	0	Q_{well}	none
Recharge	0	$q\Delta X_s\Delta Y_s$	none
River	$L_{river}\Delta X_s\Delta Y_s$	$L_{river}\Delta X_s\Delta Y_s h_{river}$	$h > R_{bed}$
	0	$L_{river}\Delta X_s\Delta Y_s (h_{river} - R_{bed})$	$h < R_{bed}$
Drain	$L_{drain}\Delta X_s\Delta Y_s$	$L_{drain}\Delta X_s\Delta Y_s D_{bed}$	$h > D_{bed}$

Table 2.3.1.3-2 lists the variables used in **Table 2.3.1.3-1**, their definitions, and associated dimensions. Note that the values of these variables are set in IGW 3 through user input.

TABLE 2.3.1.3-2
Variables used in Table 2.3.1.3-1

Variables	Definition	Dimensions
Q_{well}	well flow rate	L^3/T
q	recharge rate	L/T
L_{river}	river leakance	T^{-1}
L_{drain}	drain leakance	T^{-1}
h_{river}	river stage	L
R_{bed}	elevation of river bed	L
D_{bed}	elevation of drain invert	L

This process is implemented in subroutines ADDQS1 (for head independent) and ADDQS2 (for head dependent) in the source code.

2.3.1.4 COEFFICIENT MATRIX ASSEMBLY

When **Equations 2.3.1.1-6**, **2.3.1.2-1**, and **2.3.1.3-1** are substituted into **Equation 2.3-1** the result is

$$\begin{aligned} (a_p + a_p^t + a_p^Q)h_p = a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_{NE} h_{NE} \\ + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} + S_f^Q + S_f^t \end{aligned} \quad (2.3.1.4-1)$$

2.3.1.5 MATRIX SOLUTION

A matrix solver is required to obtain a solution to **Equation 2.3.1.4-1**. Refer to **Chapter 10** for a discussion of the available solver methods.

2.3.2 THE IMPROVED FD METHOD

The improved FD method employed in IGW 3 is also referred to as the rotational CV technique (RCVT). Basically, the RCVT rotates the X - Y coordinate system by θ to adapt to the preferential flow direction. The new coordinate system is referred to as the X_L - Y_L coordinate system and provides the basis for a rotated CV and associated numerical equation. It is important to note that

the size of the CV changes with respect to the magnitude of θ . **Figure 2.3.2-1** illustrates the RCVT concept.

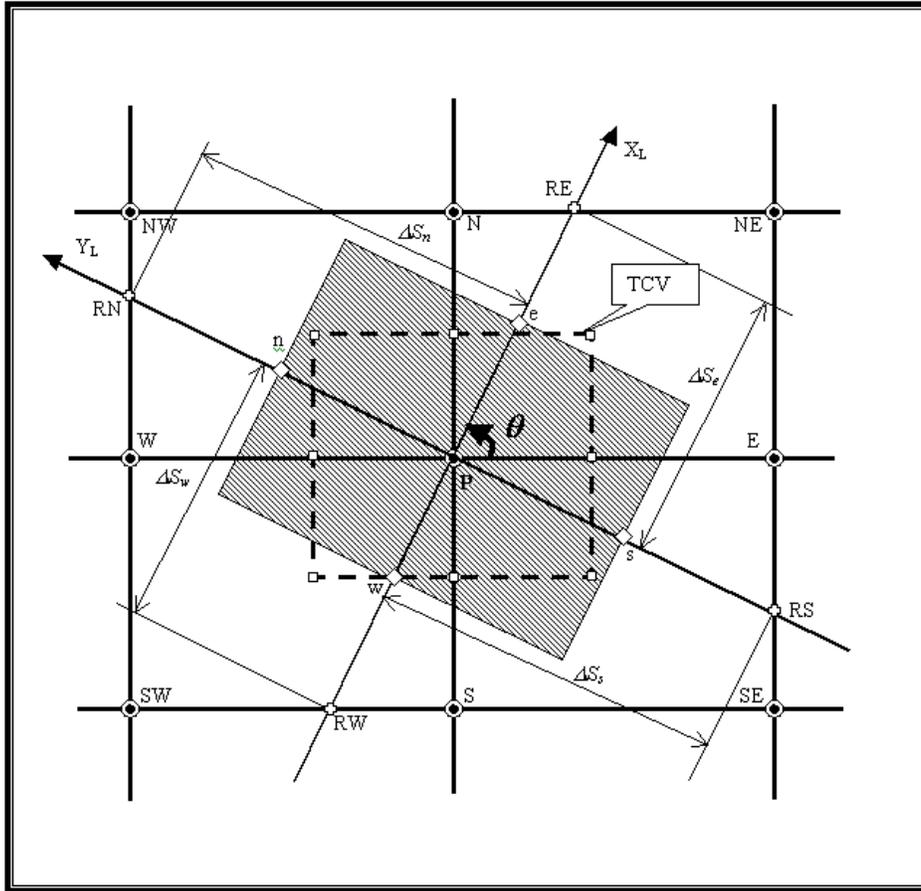


FIGURE 2.3.2-1
The RCVT concept

The following subsections present the steps and equations employed when the improved FD method is implemented.

2.3.2.1 APPROXIMATION OF SPATIAL TERMS

Applying the RCVT to each cell (refer to **Figure 2.3.2-1** for reference), the spatial term on the right-hand side of **Equation 2.3-1** can be approximated by

$$\frac{\partial}{\partial X_i} \left(T_{ij} \frac{\partial h}{\partial X_j} \right) \Delta X_s \Delta Y_s = \frac{\Delta S_n + \Delta S_s}{2} (J_e - J_w) + \frac{\Delta S_e + \Delta S_w}{2} (J_n - J_s) \quad (2.3.2.1-1)$$

where J_e , J_w , J_n , and J_s are the fluxes through the east, west, north, and south faces (after the rotation), respectively, and are defined as

$$J_e = T_{xx}'^e \frac{h_{RE} - h_P}{\Delta S_e}, \quad (2.3.2.1-2)$$

$$J_w = T_{xx}'^w \frac{h_P - h_{RW}}{\Delta S_w}, \quad (2.3.2.1-3)$$

$$J_n = T_{yy}'^n \frac{h_{RN} - h_P}{\Delta S_n}, \quad (2.3.2.1-4)$$

and

$$J_s = T_{yy}'^s \frac{h_P - h_{RS}}{\Delta S_s}, \quad (2.3.2.1-5)$$

and ΔS_n , ΔS_s , ΔS_e , and ΔS_w represent the distance between the cell center and the trans-nodal reference points RN , RS , RE , and RW , respectively.

The prime notation on the transmissivity terms (**Equations 2.3.2.1-2 – 2.3.2.1-5**) indicates that they are the principal components of the transmissivity tensor. Please note the following notation equivalencies:

$$T_{xx}'^e \equiv T_{x_L x_L}^e, \quad T_{xx}'^w \equiv T_{x_L x_L}^w, \quad T_{yy}'^n \equiv T_{y_L y_L}^n, \quad \text{and} \quad T_{yy}'^s \equiv T_{y_L y_L}^s.$$

The heads at the trans-nodal reference points h_{RE} , h_{RW} , h_{RN} , and h_{RS} are evaluated in terms of the nodal values through a simple linear interpolation scheme that has the general form

$$\phi(x) = \frac{L-x}{L} \phi_1 + \frac{x}{L} \phi_2 = \alpha(x) \phi_1 + \beta(x) \phi_2 \quad (2.3.2.1-6)$$

where ϕ_1 and ϕ_2 are nodal quantities and α and β are line element shape functions. **Figure 2.3.2.1-1** illustrates the shape function concept.

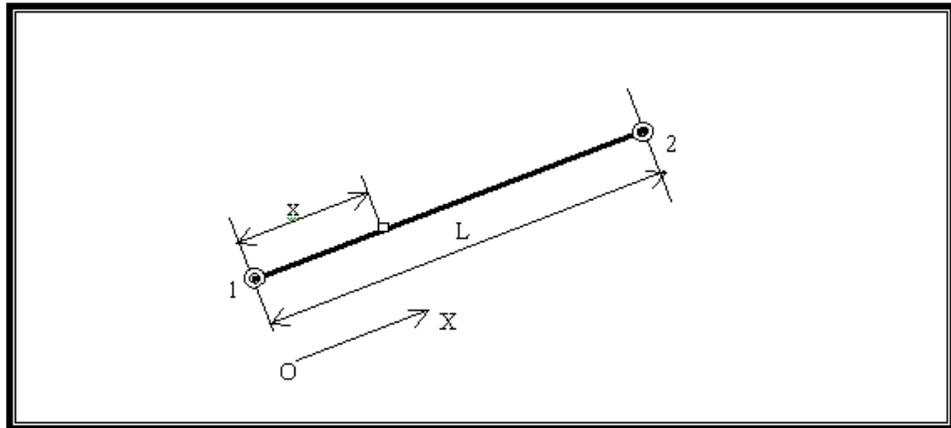


FIGURE 2.3.2.1-1

An illustration of the shape function concept. Points 1 and 2 represent nodes, L represents the distance between the nodes, and x represents the distance from node 1 to the trans-nodal point.

It should be noted that based on the slope angle, the trans-nodal points will be associated with a different set of nodes from which the associated head will be determined. The general formulation for this relationship is expressed through

$$h_{RE} = \alpha_E^E h_E + \alpha_W^E h_W + \alpha_N^E h_N + \alpha_S^E h_S + \alpha_{NE}^E h_{NE} + \alpha_{SE}^E h_{SE} + \alpha_{SW}^E h_{SW} + \alpha_{NW}^E h_{NW} \quad (2.3.2.1-7)$$

$$h_{RW} = \alpha_E^W h_E + \alpha_W^W h_W + \alpha_N^W h_N + \alpha_S^W h_S + \alpha_{NE}^W h_{NE} + \alpha_{SE}^W h_{SE} + \alpha_{SW}^W h_{SW} + \alpha_{NW}^W h_{NW} \quad (2.3.2.1-8)$$

$$h_{RN} = \alpha_E^N h_E + \alpha_W^N h_W + \alpha_N^N h_N + \alpha_S^N h_S + \alpha_{NE}^N h_{NE} + \alpha_{SE}^N h_{SE} + \alpha_{SW}^N h_{SW} + \alpha_{NW}^N h_{NW} \quad (2.3.2.1-9)$$

and

$$h_{RE} = \alpha_E^S h_E + \alpha_W^S h_W + \alpha_N^S h_N + \alpha_S^S h_S + \alpha_{NE}^S h_{NE} + \alpha_{SE}^S h_{SE} + \alpha_{SW}^S h_{SW} + \alpha_{NW}^S h_{NW} \quad (2.3.2.1-10)$$

where α_i^j are shape functions (where $i = E, W, N, S, NE, SE, SW, \text{ and } NW; j = E, W, N, \text{ and } S$) with the properties that: 1) six of them will be zero for any given index j , 2) the remaining two for a given index j will be positive, and 3) the sum of each for a given index i is unity.

Table 2.3.2.1-1 shows list of the shape functions and the associated source code variable.

TABLE 2.3.2.1-1

A list of shape functions and associated source code variables

Shape function	Source code variable
α_E	WPLT(4)
α_W	WPLT(8)
α_N	WPLT(6)
α_S	WPLT(2)
α_{NE}	WPLT(5)
α_{SE}	WPLT(3)
α_{SW}	WPLT(1)
α_{NW}	WPLT(7)

Equations 2.3.2.1-1 – 2.3.2.1-10 are combined to yield

$$\frac{\partial}{\partial X_i} \left(T_{ij} \frac{\partial h}{\partial X_j} \right) \Delta X_s \Delta Y_s = a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_{NE} h_{NE} + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} - a_P h_P \quad (2.3.2.1-11)$$

Note that this equation is of the same form as **Equation 2.3.1.1-6** but in this case the coefficients are defined as

$$a_E = \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T'_{xx}{}^e \sum_j a_E^j, \quad (2.3.2.1-12)$$

$$a_W = \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T'_{xx}{}^w \sum_j a_W^j, \quad (2.3.2.1-13)$$

$$a_N = \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T'_{yy}{}^n \sum_j a_N^j, \quad (2.3.2.1-14)$$

$$a_S = \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T'_{yy}{}^s \sum_j a_S^j, \quad (2.3.2.1-15)$$

$$a_{NE} = \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T'_{xx}{}^e \alpha_{NE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T'_{xx}{}^w \alpha_{NE}^W + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T'_{yy}{}^n \alpha_{NE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T'_{yy}{}^s \alpha_{NE}^S, \quad (2.3.2.1-16)$$

$$a_{NW} = \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T'_{xx}{}^e \alpha_{NW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T'_{xx}{}^w \alpha_{NW}^W + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T'_{yy}{}^n \alpha_{NW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T'_{yy}{}^s \alpha_{NW}^S, \quad (2.3.2.1-17)$$

$$a_{SE} = \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T'_{xx}{}^e \alpha_{SE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T'_{xx}{}^w \alpha_{SE}^W + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T'_{yy}{}^n \alpha_{SE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T'_{yy}{}^s \alpha_{SE}^S, \quad (2.3.2.1-18)$$

$$a_{SW} = \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T'_{xx}{}^e \alpha_{SW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T'_{xx}{}^w \alpha_{SW}^W + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T'_{yy}{}^n \alpha_{SW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T'_{yy}{}^s \alpha_{SW}^S, \quad (2.3.2.1-19)$$

and

$$a_P = a_E + a_W + a_N + a_S + a_{NE} + a_{NW} + a_{SE} + a_{SW}. \quad (2.3.2.1-20)$$

where $j = E, W, N,$ or S .

The implementation of this process was done in the source code subroutine NEWCOEFFLOW. The same derived variable (CST2) that was employed in the traditional FD approach is employed here to store the coefficient values. Refer to the end of **Section 2.3.1.1** for more details.

2.3.2.2 APPROXIMATION OF TIME DERIVATIVE TERM

The time derivative term on the left-hand side of **Equation 2.3-1** is: a) equal to zero if the model is being solved at steady state, or b) defined by

$$S \frac{\partial h}{\partial t} \Delta X_x \Delta Y_s = S \frac{h^{n+1} - h^n}{\Delta t} \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} = a_p^t h^{n+1} - S_f^t \quad (2.3.2.2-1)$$

where

$$a_p^t = \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \frac{S}{\Delta t}, \quad (2.3.2.2-2)$$

and

$$S_f^t = \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \frac{S}{\Delta t} h^n. \quad (2.3.2.2-3)$$

2.3.2.3 APPROXIMATION OF SOURCE/SINK TERM

The source/sink term can be expressed as

$$Q_s \Delta X_s \Delta Y_s = a_p^Q h_p + S_f^Q \quad (2.3.2.3-1)$$

where a_p^Q includes any head-dependent sources and sinks and S_f^Q includes any head-independent sources and sinks.

Table 2.3.2.3-1 lists the types of sources and sinks available in IGW 3, their associated head-dependent and head-independent components, and any conditions that determine when the source/sink is active.

TABLE 2.3.2.3-1

Types of source/sinks in IGW 3 and associated properties

Source/ Sink	Head Dependent Component	Head Independent Component	Con- dition
Well	0	Q_{well}	none
Re-charge	0	$q \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2}$	none
River	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{river}$	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{river} h_{river}$	$h > R_{bed}$
	0	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{river} (h_{river} - R_{bed})$	$h < R_{bed}$
Drain	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{drain}$	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{drain} D_{bed}$	$h > D_{bed}$

Table 2.3.2.3-2 lists the variables used in **Table 2.3.2.3-1**, their definitions, and associated dimensions. Note that the values of these variables are set in IGW 3 through user input.

TABLE 2.3.2.3-2
Variables used in Table 2.3.1.3.1

Variables	Definition	Dimensions
Q_{well}	well flow rate	L^3/T
q	recharge rate	L/T
L_{river}	river leakance	T^{-1}
L_{drain}	drain leakance	T^{-1}
h_{river}	river stage	L
R_{bed}	elevation of river bed	L
D_{bed}	elevation of drain invert	L

This process is implemented in subroutines ADDQS1 (for head independent components) and ADDQS2 (for head dependent components) in the source code. Note that these are the same subroutines associated with the traditional FD method. The only difference is in the areas associated with each cell.

2.3.2.4 COEFFICIENT MATRIX ASSEMBLY

When **Equations 2.3.2.1-11**, **2.3.2.2-1**, and **2.3.2.3-1** are substituted into **Equation 2.3-1** the result is

$$\begin{aligned} (a_p + a_p^t + a_p^q)h_p = & a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_{NE} h_{NE} \\ & + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} + S_f^q + S_f^t \end{aligned} \quad (2.3.2.4-1)$$

This equation has the same form as **Equation 2.3.1.4-1** but the coefficients are different when the slope angle is not equal to zero.

2.3.2.5 MATRIX SOLUTION

A matrix solver is required to obtain a solution to **Equation 2.3.2.4-1**. Refer to **Chapter 10** for a discussion of the available solver methods.

2.4 VELOCITY CALCULATION

Once the head values have been solved for, the seepage velocities are calculated by

$$v_i = -\frac{K_{ij}}{n_e} \frac{\partial h}{\partial X_j} \quad (2.4-1)$$

where n_e is the effective porosity of the porous medium.

The velocities are calculated at the cell faces using the appropriately determined K values (see the end of **Section 2.2**) and head gradients (obtained from a central difference scheme). A simple arithmetic mean is used to determine the nodal velocities from the cell face velocities. In future versions of IGW 3, non-uniform grids may be present and the arithmetic mean calculation will be replaced with a linear interpolation technique.

The velocity calculation functions are coded in subroutine CALUVXJQT.

2.5 ADDITIONAL SOLVER INFORMATION

This section presents some additional information about the IGW flow solver techniques.

2.5.1 SLOPE ANGLE SPECIFICATIONS

For any node that is adjacent to an inactive cell (refer to **Appendix A** in the *IGW 3 User's Manual*), the anisotropy orientation, or slope, angle is set to zero. In IGW 3, inactive cells are not allowed to have any flux through their boundaries. In essence, this means that the associated coefficient that corresponds to the interface between the active and inactive cell should be zero. However, the no flow boundary condition is set in IGW 3 by setting $K_{ij} = 0$. If there exists a non-zero slope angle, the associated K_{ij} terms, which also affect the magnitude of the coefficient, will be non-zero and subsequently lead to a non-zero coefficient. Thus the slope needs to be set to zero in these cases to avoid the scenario in which a cell is set to a no-flow boundary but water still 'leaks' into it from an adjacent cell or cells.

2.5.2 INNER ITERATIONS

An additional water table iteration (also referred to as 'inner' iteration) must be performed for every flow, or outer, iteration when the model involves an unconfined aquifer. This inner iteration scheme is implemented in the VB interface code.

Also part of the inner loop: 1) a non-linear iteration technique that accounts for any head dependent source/sinks, and 2) the scheme for classifying cells as 'dry' or 'wet' (refer to **Appendix A** in the *IGW 3 User's Manual*).

2.6 FLOW SOLVER FLOW CHART

Figure 2.6-1 shows a flow chart that lists the steps involved in the IGW 3 flow solver procedure.

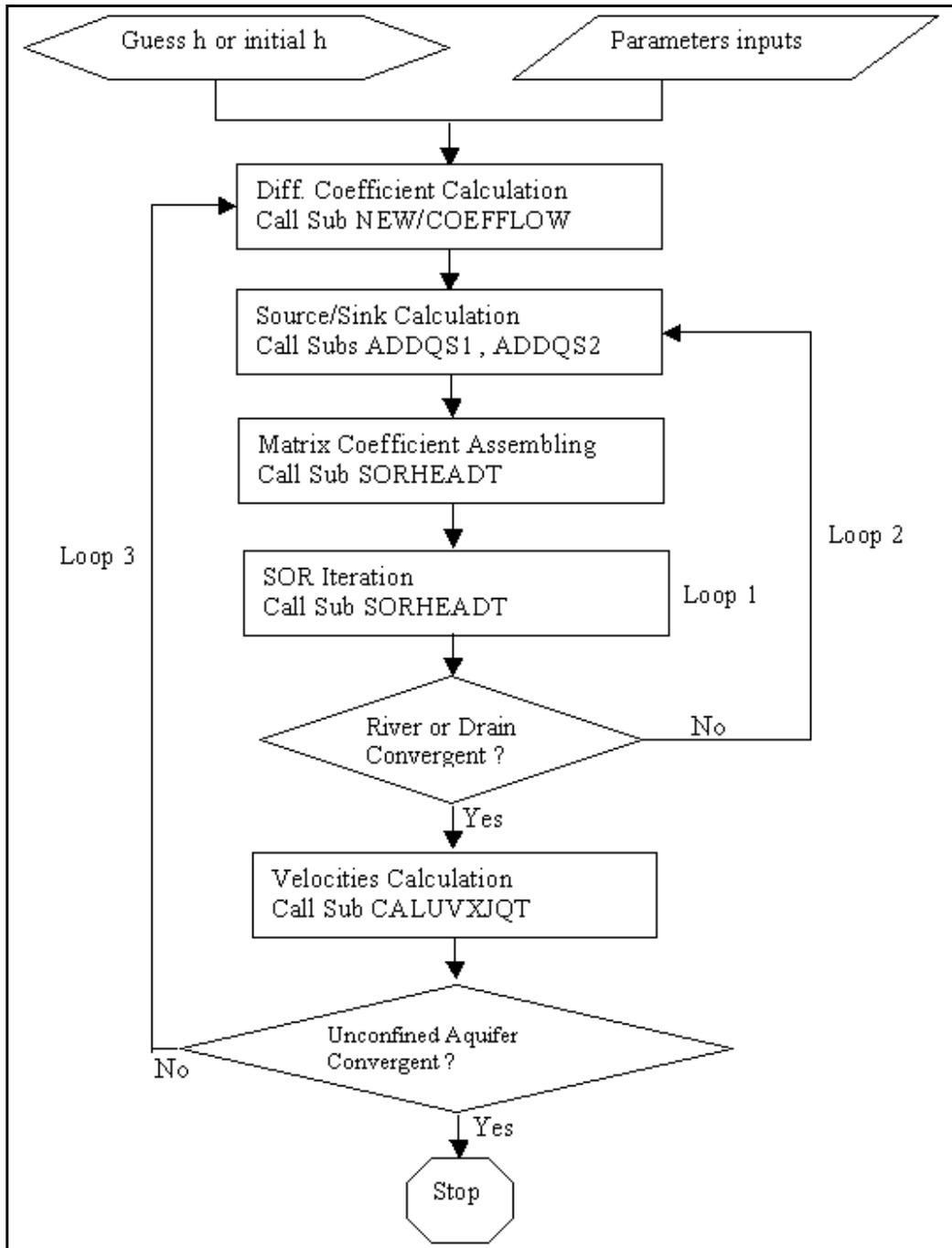


FIGURE 2.6-1
Flow chart for the IGW 3 flow solver

CHAPTER 3: TRANSPORT SOLVER

The transport solver is discussed in this chapter. The following sections discuss various components and features of the transport solver.

3.1 GOVERNING EQUATION

The partial differential equation describing 2-D solute transport in porous media can be written as

$$\frac{\partial(n_e BC)}{\partial t} + \frac{\partial(n_e B u_i C)}{\partial X_i} = \frac{\partial}{\partial X_i} (n_e B D_{ij} \frac{\partial C}{\partial X_j}) - \rho_b \frac{\partial(BC^*)}{\partial t} - \lambda n_e BC + q_s C_s \quad (3.1-1)$$

where C = the solute concentration [M/L³],
 B = the aquifer thickness [L],
 n_e = the effective porosity [-],
 u_i = seepage or averaged pore velocity in the X_i direction [L/T],
 D_{ij} = the dispersion coefficient tensor [L²/T],
 X_i = the Cartesian coordinate [L],
 C* = the concentration of solute species adsorbed to solids [M/M],
 ρ_b = the bulk density of the solids [M/L³],
 λ = the decay coefficient [T⁻¹],
 q_s = the volume flow rate per unit area of the source or sink [L/T], and
 C_s = the solute concentration in the source or sink fluid [M/L³].

Consider

$$C^* = K_d C \quad (3.1-2)$$

where K_d = the distribution coefficient [L³/M].

Note that this relationship is based on the assumption that the adsorption isotherm can be described with a linear and reversible equation.

If **Equation 3.1-2** is substituted into **Equation 3.1-1** the result is

$$\frac{\partial(n_e B R_d C)}{\partial t} + \frac{\partial(n_e B u_i C)}{\partial X_i} = \frac{\partial}{\partial X_i} (n_e B D_{ij} \frac{\partial C}{\partial X_j}) - \lambda n_e BC + q_s C_s \quad (3.1-3)$$

or

$$\begin{aligned} \frac{\partial C}{\partial t} + \frac{u_i}{R_d} \frac{\partial C}{\partial X_i} &= \frac{1}{n_e B R_d} \frac{\partial}{\partial X_i} (n_e B D_{ij} \frac{\partial C}{\partial X_j}) \\ &- \left(\frac{q_s + \lambda n_e B}{n_e B R_d} + \frac{R_d - 1}{B R_d} \frac{\partial B}{\partial t} \right) C + \frac{q_s C_s}{n_e B R_d} \end{aligned} \quad (3.1-4)$$

where R_d is the retardation factor and is defined in the relationship

$$R_d = 1 + \frac{\rho_b K_d}{n_e} . \quad (3.1-5)$$

Equation 3.1-4 is the form that is employed in the IGW 3 flow solver. In comparison to most texts, this form of the equation incorporates an extra term that contains $(\partial B/\partial t)$ to address the unconfined aquifer case.

The hydrodynamic dispersion tensor for isotropic porous media is defined through the components

$$D_{xx} = \alpha_L \frac{u_x u_x}{\sqrt{u_x^2 + u_y^2}} + D^* , \quad (3.1-6)$$

$$D_{yy} = \alpha_T \frac{u_y u_y}{\sqrt{u_x^2 + u_y^2}} + D^* , \quad (3.1-7)$$

and

$$D_{xy} = D_{yx} = (\alpha_L - \alpha_T) \frac{u_x u_y}{\sqrt{u_x^2 + u_y^2}} \quad (3.1-8)$$

where α_L = the longitudinal dispersivity [L],
 α_T = the transverse dispersivity [L], and
 D^* = the effective molecular diffusion coefficient [L²/T].

3.2 DISCRETIZATION OF THE GOVERNING EQUATION

IGW 3 incorporates three different methods for discretizing **Equation 3.1-3** or **3.1-4**. These methods include: 1) a fully implicit finite difference method (FIFD), 2) the modified method of characteristics (MMOC), and 3) the random walk method (RW). The user may choose between the MMOC and the RW method as the primary transport solver. The FIFD will be applied to cells that contain wells when the MMOC is selected. This is because the MMOC relies on particle tracking to approximate the advection term of the governing equation and particle tracking is not available in these cells due to the lack of a unique characteristic curve at a well node.

The various methods are discussed in the following subsections.

3.2.1 FULLY IMPLICIT FINITE DIFFERENCE MEHTOD

The FIFD method discretizes **Equation 3.1-3**. In the software discretization scheme, each term in the equation is multiplied by $\Delta X_s \Delta Y_s$ to take into account the area of each term.

The steps involved are discussed in the following subsections.

3.2.1.1 APPROXIMATION OF ADVECTION TERM

The advection term can be broken down into its x- and y-direction components, where

$$\Delta X_s \Delta Y_s \frac{\partial(n_e B u_i C)}{\partial X_i} = \Delta X_s \Delta Y_s \frac{\partial(n_e B u_x C)}{\partial X} + \Delta X_s \Delta Y_s \frac{\partial(n_e B u_y C)}{\partial Y}. \quad (3.2.1.1-1)$$

This relationship can be cast in the form

$$\begin{aligned} \Delta X_s \Delta Y_s \frac{\partial(n_e B u_x C)}{\partial X} + \Delta X_s \Delta Y_s \frac{\partial(n_e B u_y C)}{\partial Y} \\ = \Delta Y_s (F_e - F_w) + \Delta X_s (F_n - F_s) \end{aligned}, \quad (3.2.1.1-2)$$

where the solute fluxes are defined by

$$F_e = C_e q_e, \quad (3.2.1.1-3)$$

$$F_w = C_w q_w, \quad (3.2.1.1-4)$$

$$F_n = C_n q_n, \quad (3.2.1.1-5)$$

and

$$F_s = C_s q_s, \quad (3.2.1.1-6)$$

and where the subscripts on the terms indicate the values are those at the respective cell face.

The non-nodal C_e , C_w , C_n , and C_s are evaluated in terms of the nodal values using a simple upwind scheme where the cell-face concentration between two neighboring nodes in a particular direction is equal to the concentration at the upstream node along the same direction. In equation form,

$$C_e = \begin{cases} C_E & \text{if } q_e > 0 \\ C_P & \text{if } q_e < 0 \end{cases}, \quad (3.2.1.1-7)$$

$$C_e = \begin{cases} C_W & \text{if } -q_w > 0 \\ C_P & \text{if } -q_w < 0 \end{cases}, \quad (3.2.1.1-8)$$

$$C_e = \begin{cases} C_N & \text{if } q_n > 0 \\ C_P & \text{if } q_n < 0 \end{cases}, \quad (3.2.1.1-9)$$

and

$$C_e = \begin{cases} C_S & \text{if } -q_s > 0 \\ C_P & \text{if } -q_s < 0 \end{cases}. \quad (3.2.1.1-10)$$

Note that the IGW 3 convention for flux direction is “+” for flux into the cell and “-” for flux leaving the cell.

Substituting **Equations 3.2.1.1-3 – 3.2.1.1-10** into **Equation 3.2.1.1-2** yields

$$\Delta X_s \Delta Y_s \frac{\partial(n_e B u_i C)}{\partial X_i} = a_E^{ADV} C_E + a_W^{ADV} C_W + a_N^{ADV} C_N + a_S^{ADV} C_S - a_P^{ADV} C_P \quad (3.2.1.1-11)$$

where

$$a_E^{ADV} = \Delta Y_s \max[q_e, 0] , \quad (3.2.1.1-12)$$

$$a_W^{ADV} = \Delta Y_s \max[-q_w, 0] , \quad (3.2.1.1-13)$$

$$a_N^{ADV} = \Delta Y_s \max[q_n, 0] , \quad (3.2.1.1-14)$$

$$a_S^{ADV} = \Delta X_s \max[-q_s, 0] , \quad (3.2.1.1-15)$$

and

$$a_P^{ADV} = \Delta Y_s (\max[-q_e, 0] + \max[q_w, 0]) + \Delta X_s (\max[-q_n, 0] + \max[q_s, 0]) . \quad (3.2.1.1-16)$$

The four cell-face flux values (q_e , q_w , q_n , and q_s) are stored in the FLUX(Nwell,4) array and calculated by source code subroutine LHSWELL. The entire process is controlled by subroutine FDCOEF.

3.2.1.2 APPROXIMATION OF DIFFUSION TERM

The user has the option of applying either the traditional CV technique or the RCVT (see **Section 2.3.2** for more information) to approximate the diffusion term. They are both presented in this section.

TRADITIONAL CV TECHNIQUE

Applying the traditional CV technique to each cell (refer to **Figure 2.1-2** for reference), the diffusion term on the right-hand side of **Equation 3.1-3** can be approximated by

$$\frac{\partial}{\partial X_i} \left(n_e B D_{ij} \frac{\partial C}{\partial X_j} \right) \Delta X_s \Delta Y_s = \Delta X_s \Delta Y_s \frac{J_e - J_w}{\Delta X_s} + \Delta X_s \Delta Y_s \frac{J_n - J_s}{\Delta Y_s} \quad (3.2.1.2-1)$$

where J_e , J_w , J_n , and J_s are the fluxes through the east, west, north, and south faces, respectively, and are defined as

$$J_e = n_e B D_{xx}^e \frac{C_E - C_P}{\Delta X} + n_e B D_{xy}^e \frac{C_{ne} - C_{se}}{\Delta Y_s} , \quad (3.2.1.2-2)$$

$$J_w = n_e B D_{xx}^w \frac{C_P - C_W}{\Delta X} + n_e B D_{xy}^w \frac{C_{nw} - C_{sw}}{\Delta Y_s} , \quad (3.2.1.2-3)$$

$$J_n = n_e B D_{yy}^n \frac{C_N - C_P}{\Delta Y} + n_e B D_{yx}^n \frac{C_{ne} - C_{nw}}{\Delta X_s} , \quad (3.2.1.2-4)$$

and

$$J_s = n_e B D_{yy}^s \frac{C_P - C_S}{\Delta Y} + n_e B D_{yx}^s \frac{C_{se} - C_{sw}}{\Delta X_s} , \quad (3.2.1.2-5)$$

where the superscripts on the dispersion terms indicate the appropriate cell face.

The non-nodal heads appearing in **Equations 2.3.1.1-2 - 2.3.1.1-5**, C_{ne} , C_{se} , C_{nw} , and C_{sw} are evaluated in terms of the nodal heads using a simple four-point average scheme.

Equations 3.2.1.2-1 - 3.2.1.2-5 are combined and simplified to

$$\begin{aligned} \frac{\partial}{\partial X_i} \left(n_e B D_{ij} \frac{\partial C}{\partial X_j} \right) \Delta X_s \Delta Y_s = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N \\ & + a_S^{Diff} C_S + a_{NE}^{Diff} C_{NE} + a_{NW}^{Diff} C_{NW} \\ & + a_{SE}^{Diff} C_{SE} + a_{SW}^{Diff} C_{SW} - a_P^{Diff} C_P \end{aligned} \quad (3.2.1.2-6)$$

where the a_i^{Diff} coefficients are referred to as the coefficients of the discretized matrix and are defined as

$$a_E^{Diff} = n_e B \frac{\Delta Y_s D_{xx}^e}{\Delta X} + n_e B \frac{D_{yx}^n - D_{yx}^s}{4} , \quad (3.2.1.2-7)$$

$$a_W^{Diff} = n_e B \frac{\Delta Y_s D_{xx}^w}{\Delta X} - n_e B \frac{D_{yx}^n - D_{yx}^s}{4} , \quad (3.2.1.2-8)$$

$$a_N^{Diff} = n_e B \frac{\Delta X_s D_{yy}^n}{\Delta Y} + n_e B \frac{D_{xy}^e - D_{xy}^w}{4} , \quad (3.2.1.2-9)$$

$$a_E^{Diff} = n_e B \frac{\Delta X_s D_{yy}^x}{\Delta Y} - n_e B \frac{D_{xy}^e - D_{xy}^w}{4} , \quad (3.2.1.2-10)$$

$$a_{NE}^{Diff} = n_e B \frac{D_{xy}^e + D_{yx}^n}{4} , \quad (3.2.1.2-11)$$

$$a_{NW}^{Diff} = -n_e B \frac{D_{xy}^w + D_{yx}^n}{4} , \quad (3.2.1.2-12)$$

$$a_{SE}^{Diff} = -n_e B \frac{D_{xy}^e + D_{yx}^s}{4} , \quad (3.2.1.2-13)$$

$$a_{SW}^{Diff} = n_e B \frac{D_{xy}^w + D_{yx}^s}{4} , \quad (3.2.1.2-14)$$

and

$$a_p^{Diff} = a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_{NE}^{Diff} + a_{NW}^{Diff} + a_{SE}^{Diff} + a_{SW}^{Diff} . \quad (3.2.1.2-15)$$

The coefficients of the discretized matrix are calculated by the source code subroutine OLDCOEFTRSP. This subroutine uses a derived type variable, CST1, to store the values. **Table 3.2.1.2-1** lists the variable variants for each individual coefficient.

TABLE 3.2.1.2-1
The individual coefficient CST1 variable variants

Coefficient	Variable Variant
a_E^{Diff}	CST1(I,J)%SE
a_W^{Diff}	CST1(I,J)%SW
a_N^{Diff}	CST1(I,J)%SN
a_S^{Diff}	CST1(I,J)%SS
a_{NE}^{Diff}	CST1(I,J)%SNE
a_{NW}^{Diff}	CST1(I,J)%SNW
a_{SE}^{Diff}	CST1(I,J)%SSE
a_{SW}^{Diff}	CST1(I,J)%SSW
a_p^{Diff}	CST1(I,J)%SP

RCVT

Applying the RCVT (refer to **Section 2.3.2** for more information) to each cell (refer to **Figure 2.3.2-1** for reference), the spatial term on the right-hand side of **Equation 3.1-3** can be approximated by

$$\frac{\partial}{\partial X_i} \left(n_e B D_{ij} \frac{\partial C}{\partial X_j} \right) \Delta X_s \Delta Y_s = \frac{\Delta S_n + \Delta S_s}{2} (J_e - J_w) + \frac{\Delta S_e + \Delta S_w}{2} (J_n - J_s) \quad (3.2.1.2-16)$$

where J_e , J_w , J_n , and J_s are the fluxes through the east, west, north, and south faces (after the rotation), respectively, and are defined as

$$J_e = n_e B D_{xx}^{te} \frac{C_{RE} - C_P}{\Delta S_e} , \quad (3.2.1.2-17)$$

$$J_w = n_e B D_{xx}^{tw} \frac{C_P - C_{RW}}{\Delta S_w} , \quad (3.2.1.2-18)$$

$$J_n = n_e BD_{yy}'^n \frac{C_{RN} - C_P}{\Delta S_n} \quad , \quad (3.2.1.2-19)$$

and

$$J_s = n_e BD_{yy}'^s \frac{C_P - C_{RS}}{\Delta S_s} \quad , \quad (3.2.1.2-20)$$

and ΔS_n , ΔS_s , ΔS_e , and ΔS_w represent the distance between the cell center and the trans-nodal reference points RN , RS , RE , and RW , respectively.

The prime notation on the dispersion variables (in **Equations 3.2.1.2-17 – 3.2.1.2-20**) indicates that they are the principal components of the dispersion tensor. These are defined as

$$D'_{xx} = \alpha_L \sqrt{u_x^2 + u_y^2} \quad (3.2.1.2-21)$$

and

$$D'_{yy} = \alpha_T \sqrt{u_x^2 + u_y^2} \quad . \quad (3.2.1.2-22)$$

The concentrations at the trans-nodal reference points C_{RE} , C_{RW} , C_{RN} , and C_{RS} are evaluated in terms of the nodal values through a simple linear interpolation scheme that is described in **Section 2.3.2.1** (reference to **Equation 2.3.2.1-6** and **Figure 2.3.2.1-1**).

The general formulation for this relationship is expressed through

$$C_{RE} = \alpha_E^E C_E + \alpha_W^E C_W + \alpha_N^E C_N + \alpha_S^E C_S + \alpha_{NE}^E C_{NE} + \alpha_{SE}^E C_{SE} + \alpha_{SW}^E C_{SW} + \alpha_{NW}^E C_{NW} \quad , \quad (3.2.1.2-23)$$

$$C_{RW} = \alpha_E^W C_E + \alpha_W^W C_W + \alpha_N^W C_N + \alpha_S^W C_S + \alpha_{NE}^W C_{NE} + \alpha_{SE}^W C_{SE} + \alpha_{SW}^W C_{SW} + \alpha_{NW}^W C_{NW} \quad , \quad (3.2.1.2-24)$$

$$C_{RN} = \alpha_E^N C_E + \alpha_W^N C_W + \alpha_N^N C_N + \alpha_S^N C_S + \alpha_{NE}^N C_{NE} + \alpha_{SE}^N C_{SE} + \alpha_{SW}^N C_{SW} + \alpha_{NW}^N C_{NW} \quad , \quad (3.2.1.2-25)$$

and

$$C_{RE} = \alpha_E^S C_E + \alpha_W^S C_W + \alpha_N^S C_N + \alpha_S^S C_S + \alpha_{NE}^S C_{NE} + \alpha_{SE}^S C_{SE} + \alpha_{SW}^S C_{SW} + \alpha_{NW}^S C_{NW} \quad . \quad (3.2.1.2-26)$$

Equations 3.2.1.2-16 – 3.2.1.2-26 are combined to yield

$$\begin{aligned} \frac{\partial}{\partial X_i} \left(n_e B D_{ij} \frac{\partial C}{\partial X_j} \right) \Delta X_s \Delta Y_s = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N \\ & + a_s^{Diff} C_s + a_{NE} C_{NE} + a_{NW} C_{NW} \\ & + a_{SE} C_{SE} + a_{SW} C_{SW} - a_p^{Diff} C_p \end{aligned} \quad (3.2.1.2-27)$$

The coefficients are defined as

$$a_E^{Diff} = n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{\prime e} \sum_j a_E^j, \quad (3.2.1.2-28)$$

$$a_W^{Diff} = n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{\prime w} \sum_j a_W^j, \quad (3.2.1.2-29)$$

$$a_N^{Diff} = n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{\prime n} \sum_j a_N^j, \quad (3.2.1.2-30)$$

$$a_s^{Diff} = n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{\prime s} \sum_j a_s^j, \quad (3.2.1.2-31)$$

$$\begin{aligned} a_{NE} = & n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{\prime e} \alpha_{NE}^E + n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{\prime w} \alpha_{NE}^W \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{\prime n} \alpha_{NE}^N + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{\prime s} \alpha_{NE}^S \end{aligned}, \quad (3.2.1.2-32)$$

$$\begin{aligned} a_{NW} = & n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{\prime e} \alpha_{NW}^E + n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{\prime w} \alpha_{NW}^W \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{\prime n} \alpha_{NW}^N + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{\prime s} \alpha_{NW}^S \end{aligned}, \quad (3.2.1.2-33)$$

$$\begin{aligned} a_{SE} = & n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{\prime e} \alpha_{SE}^E + n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{\prime w} \alpha_{SE}^W \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{\prime n} \alpha_{SE}^N + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{\prime s} \alpha_{SE}^S \end{aligned}, \quad (3.2.1.2-34)$$

$$\begin{aligned} a_{SW} = & n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{\prime e} \alpha_{SW}^E + n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{\prime w} \alpha_{SW}^W \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{\prime n} \alpha_{SW}^N + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{\prime s} \alpha_{SW}^S \end{aligned}, \quad (3.2.1.2-35)$$

and

$$a_p^{Diff} = a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_{NE}^{Diff} + a_{NW}^{Diff} + a_{SE}^{Diff} + a_{SW}^{Diff} \quad (3.2.1.2-36)$$

where $j = E, W, N, \text{ or } S$.

The implementation of this process was done in the source code subroutine NEWCOEFTRSP. The same derived variable (CST1) that was employed in the traditional FD approach is employed here to store the coefficient values. Refer to the end of **Section 2.3.1.1** for more details.

3.2.1.3 APPROXIMATION OF TIME DERIVATIVE TERM

The time derivative term on the left-hand side of **Equation 3.1-3** is: a) equal to zero if the model is being solved at steady state, or b) defined by

$$\Delta X_x \Delta Y_s n_e BR_d \frac{\partial C}{\partial t} = n_e BR_d \frac{C_p^{n+1} - C_p^n}{\Delta t} \Delta X_x \Delta Y_s = a_p^t C_p^{n+1} - S_f^t \quad (3.2.1.3-1)$$

when the model is being solved for transient flow. In **Equation 3.2.1.3-1**, the superscript on C indicates the time level, Δt is the time step,

$$a_p^t = \Delta X_s \Delta Y_s \frac{S}{\Delta t}, \quad (3.2.1.3-2)$$

and

$$S_f^t = \Delta X_s \Delta Y_s \frac{S}{\Delta t} C_p^n. \quad (3.2.1.3-3)$$

When the RCVT is implemented, the following substitution occurs

$$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \text{ for } \Delta X_s \Delta Y_s. \quad (3.2.1.3-4)$$

3.2.1.4 APPROXIMATION OF SOURCE/SINK TERM

The source/sink term represents the solute mass entering the model domain through the source or leaving the model domain through the sink. For sources, the concentration is specified by the user; for sinks it is equal to the concentration in the groundwater near the sink location. This term can be expressed as

$$q_s C_s \Delta X_s \Delta Y_s = a_p^Q C_p + S_f^Q \quad (3.2.1.4-1)$$

where

$$a_p^Q = \max[-q_s, 0] \quad (3.2.1.4-2)$$

and

$$S_f^Q = \max[q_s, 0]C_s \quad . \quad (3.2.1.4-3)$$

Note that q_s is the contribution of all source/sink terms. The individual terms are stored in the QT(I,J) array in the source code and the calculation of q_s is performed in subroutine QTOTAL.

When the RCVT is implemented, the substitution in **Equation 3.2.1.3-4** occurs.

3.2.1.5 APPROXIMATION OF DECAY TERM

The decay term in **Equation 3.1-3** can be approximated by

$$-\lambda n_e BC \Delta X_s \Delta Y_s = a_p^D C_p \quad (3.2.1.5-1)$$

where

$$a_p = \lambda n_e B \quad (3.2.1.5-2)$$

Note that λ is user defined through the VB interface and is assumed to be a nodal value.

When the RCVT is implemented, the substitution in **Equation 3.2.1.3-4** occurs.

3.2.1.6 COEFFICIENT MATRIX ASSEMBLY

Equations 3.2.1.1-11, 3.2.1.2-15 or 3.2.1.2-28, 3.2.1.3-1, 3.2.1.4-1, and 3.2.1.5-1 are substituted into **Equation 3.1-3** yielding

$$\begin{aligned} a_p C_p = & a_E C_E + a_W C_W + a_N C_N + a_S C_S + a_{NE} C_{NE} \\ & + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} + S_f^Q + S_f^t \end{aligned} \quad (3.2.1.6-1)$$

where

$$a_p = a_p^{ADV} + a_p^{Diff} + a_p^t + a_p^Q + a_p^D, \quad (3.2.1.6-2)$$

$$a_E = a_E^{ADV} + a_E^{Diff}, \quad (3.2.1.6-3)$$

$$a_W = a_W^{ADV} + a_W^{Diff}, \quad (3.2.1.6-4)$$

$$a_N = a_N^{ADV} + a_N^{Diff}, \quad (3.2.1.6-5)$$

and

$$a_S = a_S^{ADV} + a_S^{Diff}. \quad (3.2.1.6-6)$$

3.2.1.7 MATRIX SOLUTION

A matrix solver is required to obtain a solution to **Equation 3.2.1.6-1**. Refer to **Chapter 10** for a discussion of the available solver methods.

3.2.2 MMOC

Equation 3.1-4 is a Eulerian expression in which $\partial C / \partial t$ represents the rate of change of solute concentration at a fixed point in space. The Lagrangian form of the equation is

$$\begin{aligned} \frac{DC}{Dt} = & \frac{1}{n_e BR_d} \frac{\partial}{\partial X_i} (n_e B D_{ij} \frac{\partial C}{\partial X_j}) \\ & - \left(\frac{q_s + \lambda n_e B}{n_e BR_d} + \frac{R_d - 1}{BR_d} \frac{\partial B}{\partial t} \right) C + \frac{q_s C_s}{n_e BR_d} \end{aligned} \quad (3.2.2-1)$$

where the substantial derivative, DC/Dt , is defined by

$$\frac{DC}{Dt} = \frac{\partial C}{\partial t} + \bar{u}_i \frac{\partial C}{\partial X_i} \quad (3.2.2-2)$$

and represents the rate of change of solute concentration along the path line of a contaminant particle.

Note that

$$\bar{u}_i = \frac{u_i}{R_d} \quad (3.2.2-3)$$

and is the retarded velocity of the contaminant particle.

Using the FD method to approximate the substantial derivative yields

$$\frac{DC}{Dt} = \frac{C_P^{n+1} - C_P^{n*}}{\Delta t} \quad (3.2.2-4)$$

where C_P^{n+1} = the average solute concentration for cell P at the time level n+1, and

C_P^{n*} = the average solute concentration for cell P at time level n+1 due to advection alone (n* refers to the intermediate time level).

This substitution is addressed in **Section 3.2.2.4**.

In the software discretization scheme, each term in **Equation 3.2.2-1** is multiplied by $\Delta X_s \Delta Y_s$ to take into account the area of each cell. When the RCVT is implemented, the substitution in **Equation 3.2.1.3-4** occurs.

3.2.2.1 PARTICLE TRACKING TECHNIQUE

The MMOC employs the fourth-order Runge-Kutta method for particle tracking. This technique involves calculating 4 different velocities along the particles path (at the initial position, at 2 midpoint positions, and at the final position) and then using a weighted average of the four as the velocity in the tracking calculation. This process is summarized by

$$X^{n+1} = X^n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \quad (3.2.2.1-1)$$

and

$$Y^{n+1} = Y^n + \frac{l_1 + 2l_2 + 2l_3 + l_4}{6} \quad (3.2.2.1-2)$$

where

$$k_1 = \Delta t u_x(X^n, Y^n, t^n), \quad (3.2.2.1-3)$$

$$l_1 = \Delta t u_y(X^n, Y^n, t^n), \quad (3.2.2.1-4)$$

$$k_2 = \Delta t u_x\left(X^n + \frac{k_1}{2}, Y^n + \frac{l_1}{2}, t^n + \frac{\Delta t}{2}\right), \quad (3.2.2.1-5)$$

$$l_2 = \Delta t u_y\left(X^n + \frac{k_1}{2}, Y^n + \frac{l_1}{2}, t^n + \frac{\Delta t}{2}\right), \quad (3.2.2.1-6)$$

$$k_3 = \Delta t u_x\left(X^n + \frac{k_2}{2}, Y^n + \frac{l_2}{2}, t^n + \frac{\Delta t}{2}\right), \quad (3.2.2.1-7)$$

$$l_3 = \Delta t u_y\left(X^n + \frac{k_2}{2}, Y^n + \frac{l_2}{2}, t^n + \frac{\Delta t}{2}\right), \quad (3.2.2.1-8)$$

$$k_4 = \Delta t u_x(X^n + k_3, Y^n + l_3, t^n + \Delta t), \quad (3.2.2.1-9)$$

and

$$l_4 = \Delta t u_y(X^n + k_3, Y^n + l_3, t^n + \Delta t). \quad (3.2.2.1-10)$$

Because the technique requires velocities from non-nodal locations, a simple bilinear interpolation scheme has been employed to determine these velocities. The general form is given as

$$u(\xi, \eta) = u_1\phi_1(\xi, \eta) + u_2\phi_2(\xi, \eta) + u_3\phi_3(\xi, \eta) + u_4\phi_4(\xi, \eta) \quad (3.2.2.1-11)$$

where u_i are nodal quantities, ϕ_i are shape functions defined by

$$\phi_1(\xi, \eta) = \frac{(1-\xi)(1-\eta)}{4}, \quad (3.2.2.1-12)$$

$$\phi_2(\xi, \eta) = \frac{(1+\xi)(1-\eta)}{4}, \quad (3.2.2.1-13)$$

$$\phi_3(\xi, \eta) = \frac{(1 + \xi)(1 + \eta)}{4}, \quad (3.2.2.1-14)$$

and

$$\phi_4(\xi, \eta) = \frac{(1 - \xi)(1 + \eta)}{4}, \quad (3.2.2.1-15)$$

and where

$$\xi = \frac{X - X_C}{a}, \quad (3.2.2.1-16)$$

and

$$\eta = \frac{Y - Y_C}{b}. \quad (3.2.2.1-17)$$

a , b and the other terms are shown in the conceptual representation of the bilinear interpolation scheme given in **Figure 3.2.2.1-1**.

The bilinear interpolation scheme is coded in subroutine PTLINEAR; the Runge-Kutta method in subroutine FORWARDTK0.

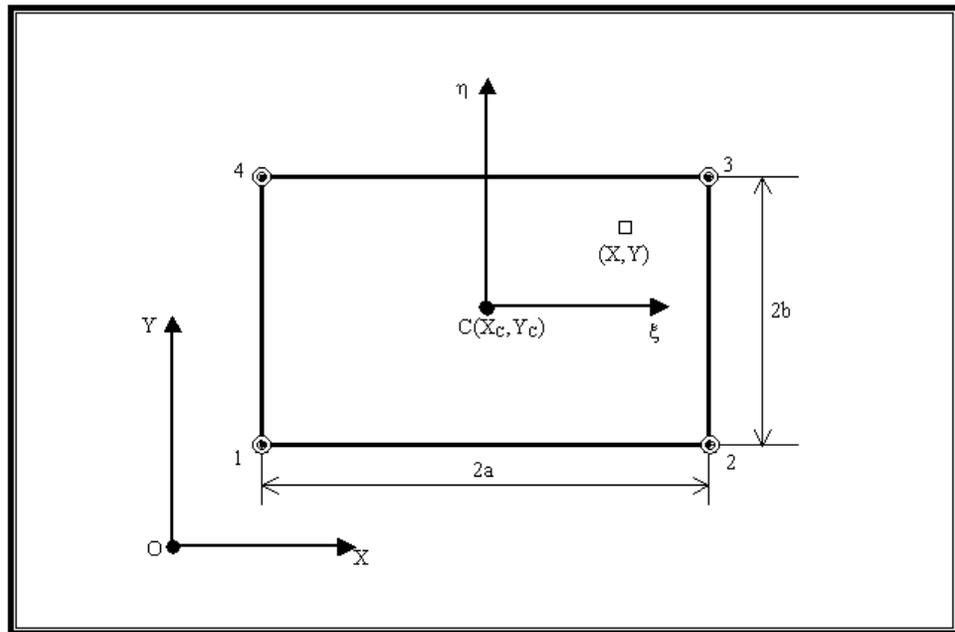


FIGURE 3.2.2.1-1
Conceptual representation of the bilinear interpolation scheme

3.2.2.2 DETERMINATION OF C_P^{n*}

In determining C_P^{n*} , the MMOC involves placing one fictitious particle at each node for the time level $n + 1$. Using a backwards particle tracking technique, the position for each

particle at time level n is determined and the concentration at that position is determined. This concentration is then assigned to C_P^{n*} . In mathematical form

$$C_P^{n*} = C^n(X_P, Y_P) \quad (3.2.2.2-1)$$

where (X_P, Y_P) is the n time level coordinate for a particle that is at location P at time level $n+1$ and tracked backwards over the time step Δt (this location is denoted p). Refer to **Figure 3.2.2.2-1** for a conceptual representation.

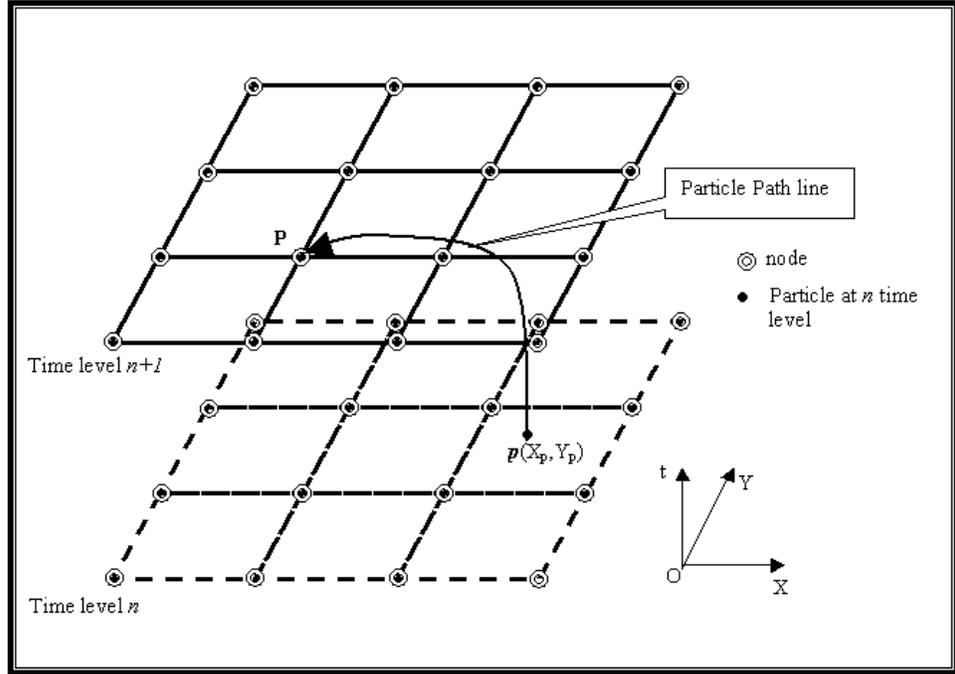


FIGURE 3.2.2.2-1
Conceptual representation of MMOC particle movement

Bilinear interpolation is used to calculate the concentration for the old time level. This technique is described in **Section 3.2.2.1**.

3.2.2.3 APPROXIMATION OF DIFFUSION TERM

The diffusion term approximation in the MMOC takes a similar form as that for the FIFD method (refer to **Section 3.2.1.2**). The approximating equation is

$$\frac{1}{n_e BR_d} \frac{\partial}{\partial X_i} \left(n_e BD_{ij} \frac{\partial C}{\partial X_j} \right) \Delta X_s \Delta Y_s =$$

$$a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N + a_S^{Diff} C_S + a_{NE}^{Diff} C_{NE} \quad (3.2.2.3-1)$$

$$+ a_{NW}^{Diff} C_{NW} + a_{SE}^{Diff} C_{SE} + a_{SW}^{Diff} C_{SW} - a_P^{Diff} C_P$$

The coefficients for the two different solver methods are presented below.

TRADITIONAL CV METHOD

The coefficients corresponding to the traditional CV method are defined as

$$a_E^{Diff} = \left(n_e B \frac{\Delta Y_s D_{xx}^e}{\Delta X} + n_e B \frac{D_{yx}^n - D_{yx}^s}{4} \right) \frac{1}{n_e BR_d}, \quad (3.2.2.3-2)$$

$$a_W^{Diff} = \left(n_e B \frac{\Delta Y_s D_{xx}^w}{\Delta X} - n_e B \frac{D_{yx}^n - D_{yx}^s}{4} \right) \frac{1}{n_e BR_d}, \quad (3.2.2.3-3)$$

$$a_N^{Diff} = \left(n_e B \frac{\Delta X_s D_{yy}^n}{\Delta Y} + n_e B \frac{D_{xy}^e - D_{xy}^w}{4} \right) \frac{1}{n_e BR_d}, \quad (3.2.2.3-4)$$

$$a_E^{Diff} = \left(n_e B \frac{\Delta X_s D_{yy}^x}{\Delta Y} - n_e B \frac{D_{xy}^e - D_{xy}^w}{4} \right) \frac{1}{n_e BR_d}, \quad (3.2.2.3-5)$$

$$a_{NE} = n_e B \frac{D_{xy}^e + D_{yx}^n}{4} \frac{1}{n_e BR_d}, \quad (3.2.2.3-6)$$

$$a_{NW} = -n_e B \frac{D_{xy}^w + D_{yx}^n}{4} \frac{1}{n_e BR_d}, \quad (3.2.2.3-7)$$

$$a_{SE} = -n_e B \frac{D_{xy}^e + D_{yx}^s}{4} \frac{1}{n_e BR_d}, \quad (3.2.2.3-8)$$

$$a_{SW} = n_e B \frac{D_{xy}^w + D_{yx}^s}{4} \frac{1}{n_e BR_d}, \quad (3.2.2.3-9)$$

and

$$a_P^{Diff} = a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_{NE} + a_{NW} + a_{SE} + a_{SW}. \quad (3.2.2.3-10)$$

RCVT

The coefficients corresponding to the RCVT method are defined as

$$a_E^{Diff} = \frac{1}{n_e BR_d} n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^e \sum_j a_E^j, \quad (3.2.2.3-11)$$

$$a_W^{Diff} = \frac{1}{n_e BR_d} n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^w \sum_j a_W^j, \quad (3.2.2.3-12)$$

$$a_N^{Diff} = \frac{1}{n_e BR_d} n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^n \sum_j a_N^j, \quad (3.2.2.3-13)$$

$$a_s^{Diff} = \frac{1}{n_e BR_d} n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D'_{yy}{}^s \sum_j a_s^j, \quad (3.2.2.3-14)$$

$$a_{NE} = \frac{1}{n_e BR_d} \left(\begin{aligned} & n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D'_{xx}{}^e \alpha_{NE}^E \\ & + n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D'_{xx}{}^w \alpha_{NE}^W \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D'_{yy}{}^n \alpha_{NE}^N \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D'_{yy}{}^s \alpha_{NE}^S \end{aligned} \right), \quad (3.2.2.3-15)$$

$$a_{NW} = \frac{1}{n_e BR_d} \left(\begin{aligned} & n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D'_{xx}{}^e \alpha_{NW}^E \\ & + n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D'_{xx}{}^w \alpha_{NW}^W \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D'_{yy}{}^n \alpha_{NW}^N \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D'_{yy}{}^s \alpha_{NW}^S \end{aligned} \right), \quad (3.2.2.3-16)$$

$$a_{SE} = \frac{1}{n_e BR_d} \left(\begin{aligned} & n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D'_{xx}{}^e \alpha_{SE}^E \\ & + n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D'_{xx}{}^w \alpha_{SE}^W \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D'_{yy}{}^n \alpha_{SE}^N \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D'_{yy}{}^s \alpha_{SE}^S \end{aligned} \right), \quad (3.2.2.3-17)$$

$$a_{SW} = \frac{1}{n_e BR_d} \left(\begin{aligned} & n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D'_{xx}{}^e \alpha_{SW}^E \\ & + n_e B \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D'_{xx}{}^w \alpha_{SW}^W \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D'_{yy}{}^n \alpha_{SW}^N \\ & + n_e B \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D'_{yy}{}^s \alpha_{SW}^S \end{aligned} \right), \quad (3.2.2.3-18)$$

and

$$a_p^{Diff} = a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_{NE} + a_{NW} + a_{SE} + a_{SW} \quad (3.2.2.3-19)$$

where $j = E, W, N, \text{ or } S$.

3.2.2.4 APPROXIMATION OF TIME DERIVATIVE TERM

The time derivative term is defined by

$$\Delta X_x \Delta Y_s \frac{DC}{Dt} = \frac{C_P^{n+1} - C_P^{n*}}{\Delta t} \Delta X_x \Delta Y_s = a_p^t C_P^{n+1} - S_f^t \quad (3.2.2.4-1)$$

In **Equation 3.2.2.4-1**, the superscript on C indicates the time level, Δt is the time step,

$$a_p^t = \Delta X_s \Delta Y_s \frac{1}{\Delta t}, \quad (3.2.2.4-2)$$

and

$$S_f^t = \Delta X_s \Delta Y_s \frac{1}{\Delta t} C_P^{n*} \quad (3.2.2.4-3)$$

When the RCVT is implemented, the substitution in **Equation 3.2.1.3-4** occurs.

3.2.2.5 APPROXIMATION OF SOURCE/SINK TERM

The source/sink term is approximated in a similar fashion as employed in the FIFD method. This term can be expressed as

$$\frac{1}{n_e BR_d} q_s C_s \Delta X_s \Delta Y_s = a_p^Q C_P + S_f^Q \quad (3.2.2.5-1)$$

where

$$a_p^Q = \Delta X_s \Delta Y_s \frac{1}{n_e BR_d} \max[-q_s, 0] \quad (3.2.2.5-2)$$

and

$$S_f^Q = \Delta X_s \Delta Y_s \frac{1}{n_e BR_d} \max[q_s, 0] C_s \quad (3.2.2.5-3)$$

When the RCVT is implemented, the substitution in **Equation 3.2.1.3-4** occurs.

3.2.2.6 APPROXIMATION OF DECAY TERM

There are three decay terms that can, together, be approximated by

$$\frac{-\lambda n_e C}{R_d} \Delta X_s \Delta Y_s = a_p^D C_p \quad (3.2.2.6-1)$$

where

$$a_p^D = \left(\frac{q_s + \lambda n B}{n_e B R_d} + \frac{R_d - 1}{B R_d} \frac{\partial B}{\partial t} \right) \Delta X_s \Delta Y_s \quad (3.2.2.6-2)$$

When the RCVT is implemented, the substitution in **Equation 3.2.1.3-4** occurs.

3.2.2.7 COEFFICIENT MATRIX ASSEMBLY AND SOLUTION

Equations 3.2.2.3-1, 3.2.2.4-1, 3.2.2.5-1 and 3.2.2.6-1 are substituted into **Equation 3.2.2-1** yielding

$$a_p C_p = a_E C_E + a_W C_W + a_N C_N + a_S C_S + a_{NE} C_{NE} + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} + S_f^Q + S_f^I \quad (3.2.2.7-1)$$

where

$$a_p = a_p^{ADV} + a_p^{Diff} + a_p^I + a_p^Q + a_p^D, \quad (3.2.1.7-2)$$

$$a_E = a_E^{ADV} + a_E^{Diff}, \quad (3.2.1.7-3)$$

$$a_W = a_W^{ADV} + a_W^{Diff}, \quad (3.2.1.7-4)$$

$$a_N = a_N^{ADV} + a_N^{Diff}, \quad (3.2.1.7-5)$$

and

$$a_S = a_S^{ADV} + a_S^{Diff}. \quad (3.2.1.7-6)$$

Note that these equations are equivalent to those presented for the fully implicit finite difference method (**3.2.1.6-1 – 3.2.1.6-6**) but the coefficients are not equivalent.

3.2.2.8 MATRIX SOLUTION

A matrix solver is required to obtain a solution to **Equation 3.2.2.7-1**. Refer to **Chapter 10** for a discussion of the available solver methods.

3.2.3 RANDOM WALK

The RW method treats the transport of solute mass by a large number of moving particles.

The particle tracking techniques discussed in **Section 3.3.1** have been used to approximate advection effects; an additional term is added in the RW formulation that takes into account dispersion effects by adding a random displacement to the particle after the advective motion has completed. The governing equations for particle movement are

$$X_P^{n+1} = X_P^n + \Delta X_{ADV} + \Delta X_{Diff} \quad (3.2.3-1)$$

and

$$Y_P^{n+1} = Y_P^n + \Delta Y_{ADV} + \Delta Y_{Diff} \quad (3.2.3-2)$$

where the X_P and Y_P terms indicate the particle coordinates and the superscripts on these terms indicate the time-step and the ΔX and ΔY terms indicate the displacements in one time-step and the subscripts indicate the mode of transport ('ADV' for advection and 'Diff' for dispersion). Refer to **Figure 3.2.3-1** for a conceptual representation.

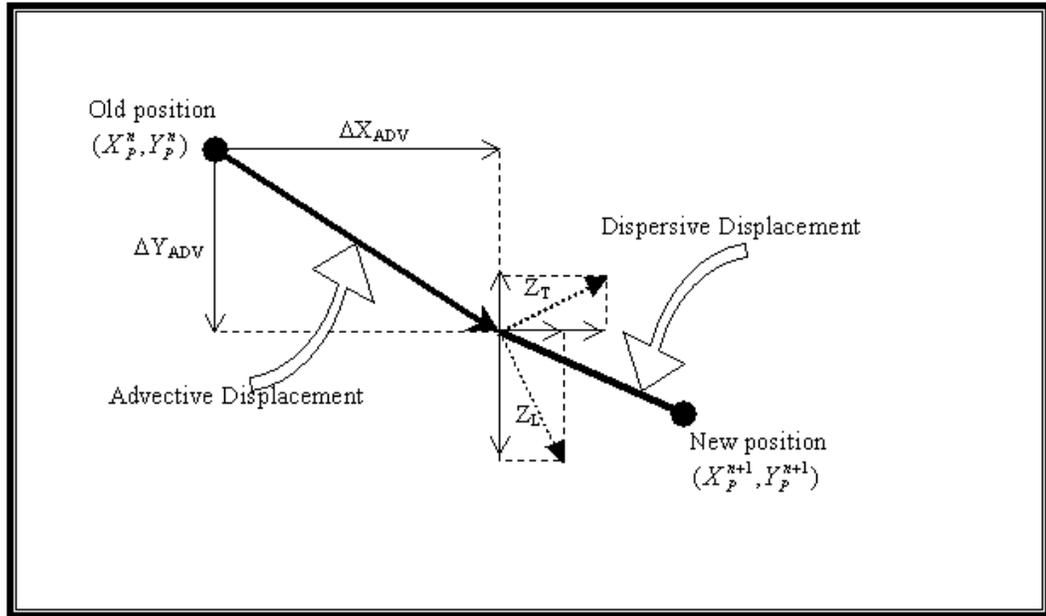


FIGURE 3.2.3-1
Conceptual visualization of the RW method

Note that the RW method was coded in subroutine TRACKFB_RW.

3.2.3.1 DETERMINATION OF ADVECTIVE DISPLACEMENT

The advective displacement is determined through a simple first-order Eulerian method that can be summarized through

$$\bar{u}_x = \frac{1}{R_d} (u_x + \Delta u_x) \quad (3.2.3.1-1)$$

and

$$\bar{u}_y = \frac{1}{R_d} (u_y + \Delta u_y) \quad (3.2.3.1-2)$$

where the \bar{u} terms are the total velocities in the direction indicated by the subscript, the u terms are the seepage velocities in the direction indicated by the subscript, and the Δu terms are the velocity correction factors.

The velocity correction factors are the spatial derivatives of the dispersion coefficients and are defined as

$$\Delta u_x = \frac{\partial D_{xx}}{\partial x} + \frac{\partial D_{xy}}{\partial y} \quad (3.2.3.1-3)$$

and

$$\Delta u_y = \frac{\partial D_{yx}}{\partial x} + \frac{\partial D_{yy}}{\partial y} \quad (3.2.3.1-4)$$

Substituting **Equation 3.2.3.1-3** and **3.2.3.1-4** into **3.2.3.1-1** and **3.2.3.1-2**, respectively, and the result is multiplied by Δt , the advective displacements can be defined as

$$\Delta X_{ADV} = \frac{1}{R_d} \left(u_x + \frac{\partial D_{xx}}{\partial x} + \frac{\partial D_{xy}}{\partial y} \right) \Delta t \quad (3.2.3.1-5)$$

and

$$\Delta Y_{ADV} = \frac{1}{R_d} \left(u_y + \frac{\partial D_{yx}}{\partial x} + \frac{\partial D_{yy}}{\partial y} \right) \Delta t \quad (3.2.3.1-6)$$

3.2.3.2 DETERMINATION OF RANDOM DISPLACEMENTS

The random dispersive displacement term is comprised of both a longitudinal and transverse component. These components are

$$Z_L = N_L(0, \sigma_L^2) + N_D(0, \sigma_D^2) = \sigma_L N_L(0,1) + \sigma_D N_D(0,1) \quad (3.2.3.2-1)$$

and

$$Z_T = N_T(0, \sigma_T^2) + N_D(0, \sigma_D^2) = \sigma_T N_T(0,1) + \sigma_D N_D(0,1) \quad (3.2.3.2-2)$$

where Z_L and Z_T are the random displacements in the longitudinal and transverse directions, respectively, $\sigma_L = \sqrt{2\alpha_L U \Delta t}$, $\sigma_T = \sqrt{2\alpha_T U \Delta t}$, $U = \sqrt{u_x^2 + u_y^2}$, $\sigma_D = \sqrt{2D^* \Delta t}$, and $N_L(0,1)$, $N_T(0,1)$ and $N_D(0,1)$ are normally distributed random numbers with a mean of zero and a standard deviation of one.

In terms of the x- and y-directions, the random displacements can be expressed as

$$\Delta X_{Diff} = Z_L \frac{\Delta X_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} + Z_T \frac{\Delta Y_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} \quad (3.2.3.2-3)$$

and

$$\Delta Y_{Diff} = Z_L \frac{\Delta Y_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} - Z_T \frac{\Delta X_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} \quad . \quad (3.2.3.2-4)$$

3.2.3.3 FINAL DISPLACEMENT

Substituting **Equations 3.2.3.1-5, 3.2.3.1-6, 3.2.3.2-3** and **3.2.3.2-4** into **Equations 3.2.3-1** and **3.2.3-2** the result is

$$\begin{aligned} X_P^{n+1} = X_P^n + \Delta X_{ADV} + Z_L \frac{\Delta X_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} \\ + Z_T \frac{\Delta Y_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} \end{aligned} \quad (3.2.3.3-1)$$

and

$$\begin{aligned} Y_P^{n+1} = Y_P^n + \Delta Y_{ADV} + Z_L \frac{\Delta Y_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} \\ - Z_T \frac{\Delta X_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} \end{aligned} \quad . \quad (3.2.3.3-2)$$

3.2.3.4 EVALUATION OF CONCENTRATION

The conversion of particle density to solute concentration is coded in the VB interface. This was done because the RW method can be visualized (which is done in the VB interface) as either a collection of particles or a concentration plume. The conversion is performed by evaluating the concentrations at nodal points based on

$$C_P^{n*} = \begin{cases} \sum_{i=1}^{NP_m} C_i^n & \text{if } NP_m > 0 \\ 0 & \text{if } NP_m = 0 \end{cases} \quad (3.2.3.4-1)$$

where NP_m = the number of particles within cell P , and
 C_i^n = the concentration of the i^{th} particle at time level n .

Refer to **Figure 3.2.3.4-1** for an illustration.

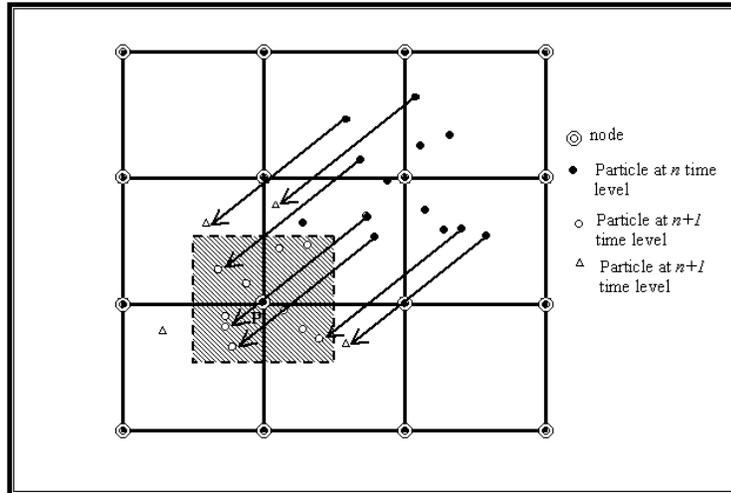


FIGURE 3.2.3.4-1

An illustration of particle movement and concentration evaluation in the RW method (note that this picture shows the overall particle path, not the component path as described above). The open circles represent those particles within the cell; the open triangles represent those outside of the cell.

3.3 ADDITIONAL INFORMATION

Additional information concerning the transport flow solver is presented in this section.

3.3.1 EULERIAN PARTICLE TRACKING

In terms of single, non-MMOC related particle tracking, the particle tracking method employed is the first-order Eulerian technique (not the Runge-Kutta technique described in the **Section 3.2.2.2**). This process is described by

$$X^{n+1} = \frac{\Delta t}{R_d} u_x(X^n, Y^n) \quad (3.3.1-1)$$

and

$$Y^{n+1} = \frac{\Delta t}{R_d} u_y(X^n, Y^n) \quad (3.3.1-2)$$

where the X and Y terms denote the particle coordinates at the time level indicated by the associated superscript and the u_x and u_y terms are the x- and y-direction velocities at X^n and Y^n , respectively. This method employs a uniform time step (Δt) for all particles and the bilinear interpolation method (see **Section 3.2.2.1**) for determining velocities that are non-nodal.

3.4 TRANSPORT SOLVER FLOW CHART

Figure 3.4-1 shows a flow chart that lists the steps involved in the IGW 3 transport solver procedure.

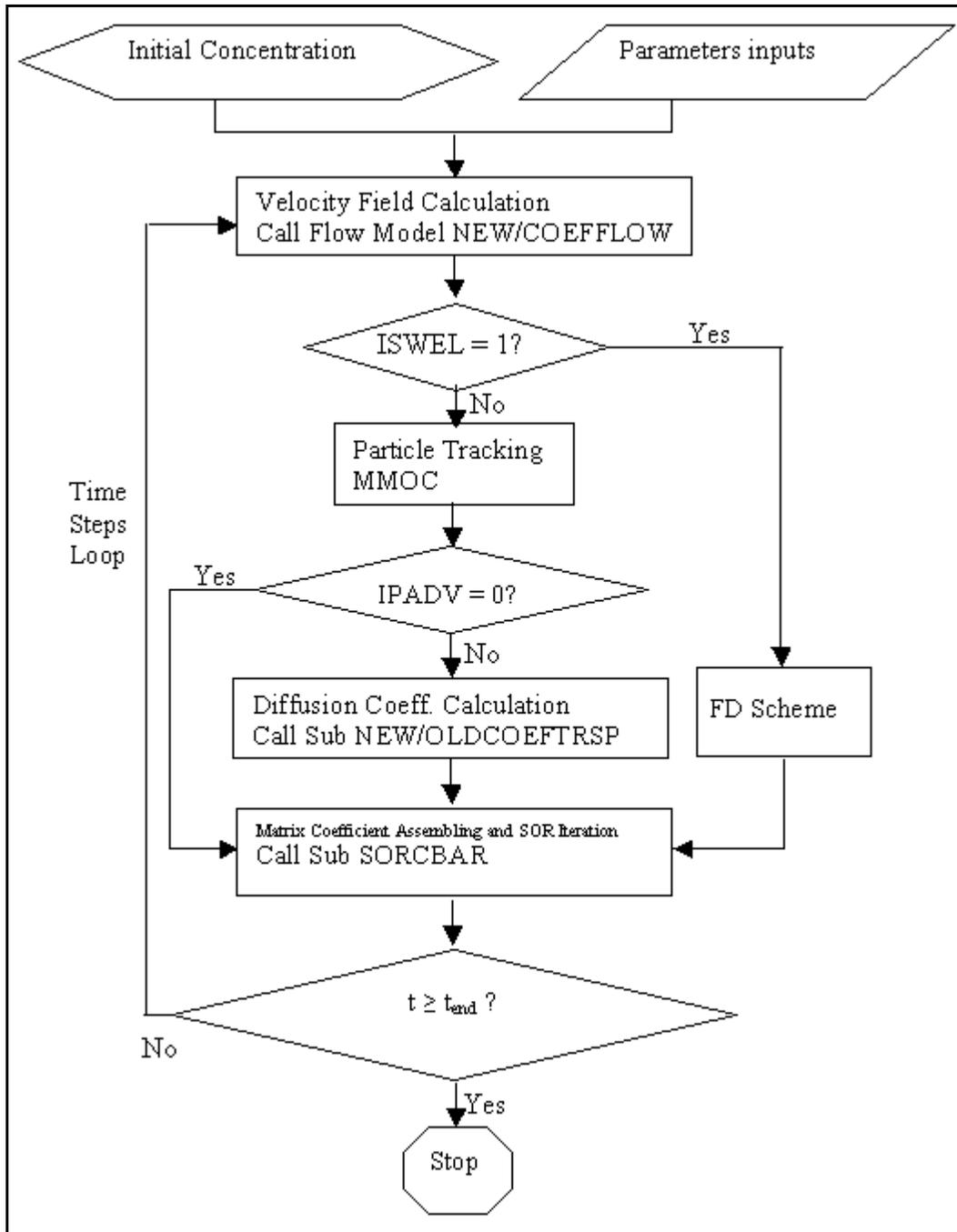


FIGURE 3.4-1
Flow chart for the IGW 3 transport solver procedure

CHAPTER 4: MONTE CARLO SIMULATION

A Monte Carlo (MC) simulation is one in which multiple statistically equivalent models are generated and solved and the results are statistically analyzed. It is used to evaluate the impacts of uncertainties associated with model inputs on the calculated results. The MC simulation scheme employed in IGW 3 includes 3 parts: 1) generating random fields of model input parameters, 2) solving head and solute transport equations based on the generated fields, and 3) recursively calculating the statistical distributions of the output parameters. The following sections describe these processes.

4.1 RANDOM FIELD GENERATOR

At the heart of the MC simulation method is the ability to generate multiple realizations of randomly variable input parameters. Each variable parameter follows certain probabilistic model that is specified through a spectral density function or covariance function.

IGW 3 employs the Fast Fourier Transform (FFT) technique to generate random fields. It is described in the following subsection.

4.1.1 FAST FOURIER TRANSFORM TECHNIQUE

Given a zero-mean 2-D stochastic process, $h(t_1, t_2)$, with a stationary covariance function

$$R(\tau_1, \tau_2) = E[h(t_1 + \tau_1, t_2 + \tau_2)h^*(t_1, t_2)] \quad (4.1.1-1)$$

where $h^*(t_1, t_2)$ is the conjugate function of $h(t_1, t_2)$, the spectrum of that stochastic process, $S(f_1, f_2)$, can be expressed as the FFT of the covariance function where

$$S(f_1, f_2) = \iint R(\tau_1, \tau_2) e^{2\pi i(f_1 \tau_1 + f_2 \tau_2)} d\tau_1 d\tau_2 \quad (4.1.1-2)$$

and

$$R(\tau_1, \tau_2) = \iint S(f_1, f_2) e^{-2\pi i(f_1 \tau_1 + f_2 \tau_2)} df_1 df_2 \quad (4.1.1-3)$$

and where the corresponding discrete FFT are

$$S_{n_1, n_2} = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} R_{k_1, k_2} e^{2\pi i \left(\frac{k_1 n_1}{N_1} + \frac{k_2 n_2}{N_2} \right)} \quad (4.1.1-4)$$

and

$$R_{k_1, k_2} = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} S_{n_1, n_2} e^{-2\pi i (k_1 \Delta_1 f_{n_1} + k_2 \Delta_2 f_{n_2})} \quad (4.1.1-5)$$

where N_1 and N_2 = the 2-D domain dimensions [L],

f = the frequency [L^{-1}],

τ = the spatial quantity [L], and

Δ = the grid spacing [L].

If h_k is a stochastic process and its discrete FFT has the form

$$h_{k_1, k_2} = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} H_{n_1, n_2} e^{-2\pi i(k_1 \Delta_1 f_{n_1} + k_2 \Delta_2 f_{n_2})} \quad (4.1.1-6)$$

then H_{n_1, n_2} are random. Assuming

$$E[H_{n_1, n_2}] = 0, \quad E[H_{n, n} H_{m, m}^*] = 0, \quad \text{and } n_i \neq m_i$$

where H^* is the conjugate function of H , one obtains from **Equation 4.1.1-6**

$$E[h_{k_1, k_2}] = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} E[H_{n_1, n_2}] e^{-2\pi i(k_1 \Delta_1 f_{n_1} + k_2 \Delta_2 f_{n_2})} = 0 \quad (4.1.1-7)$$

and

$$\begin{aligned} R_{j_1, j_2} &= E[h_{k_1 + j_1, k_2 + j_2} h_{k_1, k_2}^*] \\ &= \frac{1}{N_1^2 N_2^2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} E[H_{n_1, n_2} H_{n_1, n_2}^*] e^{2\pi i(j_1 \Delta_1 f_{n_1} + j_2 \Delta_2 f_{n_2})}. \end{aligned} \quad (4.1.1-8)$$

Comparing **Equation 4.1.1-8** and **4.1.1-5** yields

$$E[H_{n_1, n_2} H_{n_1, n_2}^*] = N_1 N_2 S_{n_1, n_2} \quad (4.1.1-9)$$

Thus, generating a zero-mean stochastic process h_k with a specific covariance structure is equivalent to the generating of a different stochastic process \underline{H}_{n_1, n_2} in a frequency domain under the conditions

$$E[H_{n_1, n_2}] = 0, \quad E[H_{n, n} H_{m, m}^*] = 0, \quad n_i \neq m_i, \quad \text{and } E[H_{n_1, n_2} H_{n_1, n_2}^*] = N_1 N_2 S_{n_1, n_2}.$$

If

$$H_{n_1, n_2} = \sqrt{N_1 N_2 S_{n_1, n_2}} e^{i\theta_{n_1, n_2}} \quad (4.1.1-10)$$

and θ_{n_1, n_2} are independent random variables uniformly distributed over $[0, 2\pi]$, then the conditions outlined above are met. For the process to be real

$$H_{N_1 - n_1, N_2 - n_2} = H_{n_1, n_2}^* \quad (4.1.1-11)$$

which implies that

$$\theta_{N_1 - n_1, N_2 - n_2} = \theta_{n_1, n_2}. \quad (4.1.1-12)$$

Thus, the major steps in generating a realization of a zero-mean stochastic process can be summarized as:

- 1) $H_{0,0} = 0 + 0i$
- 2) $H_{\frac{N_1}{2}, n_2} = 0 + 0i$, for all n_2 ; and $H_{n_1, \frac{N_2}{2}} = 0 + 0i$ for all n_1
- 3) $H_{n_1, n_2} = \sqrt{N_1 N_2 S_{n_1 n_2}} e^{i\theta_{n_1 n_2}}$ for $1 \leq n_1 \leq N_1 / 2 - 1$ and $N_1 / 2 + 1 \leq n_1 \leq N_1 - 1$, and $1 \leq n_2 \leq N_2 / 2 - 1$, where θ_{n_1, n_2} are independent random variables, uniformly distributed over $[0, 2\pi]$
- 4) $H_{n_1, n_2} = H_{N_1 - n_1, N_2 - n_2}^*$, for $1 \leq n_1 \leq N_1 / 2 - 1$ and $N_1 / 2 + 1 \leq n_1 \leq N_1 - 1$, and $N_2 + 1 \leq n_2 \leq N_2 - 1$
- 5) $H_{n_1, n_2} = \sqrt{N_1 N_2 S_{n_1 n_2}} e^{i\theta_{n_1 n_2}}$ for $n_2 = 0$ and $1 \leq n_1 \leq N_1 / 2 - 1$, and for $n_1 = 0$ and $1 \leq n_2 \leq N_2 / 2 - 1$
- 6) $H_{n_1, n_2} = H_{N_1 - n_1, n_2}^*$, for $n_2 = 0$ and $N_1 / 2 + 1 \leq n_1 \leq N_1 - 1$
- 7) $H_{n_1, n_2} = H_{n_1, N_2 - n_2}^*$, for $n_1 = 0$ and $N_2 / 2 + 1 \leq n_2 \leq N_2 - 1$
- 8) Generate h_{k_1, k_2} by using the inverse FFT.

The polygon based random field generator is contained in the source code in subroutine RANDOM_FIELD.

4.2 SOLUTIONS

The MC simulation process can be applied to any of the flow or transport solver methods discussed in the previous chapters except for the RW method of transport solvers.

4.3 CALCULATING STATISTICAL DISTRIBUTIONS

There are two main types of statistical distributions that are calculated by the IGW 3 MC simulator. The first type is the point-based output statistics and includes: 1) the probability density function (PDF), 2) the cumulative density function (CDF), 3) the mean, 4) the median, 5) the mode, 6) the standard deviation, 7) the average deviation, 8) the skewness, and 9) the kurtosis. The second type is the recursively updated field-based output statistics and includes a variety of means, variances, and covariances.

Note that IGW 3 currently supports the calculation of these statistics for conductivity, head, concentration, and polyline flux only.

The two main types of statistical distributions are discussed in the following subsections.

4.3.1 POINT BASED STATISTICS

The point-based statistics are based upon a set of values at a specific MC simulation. For purposes of describing the point-based statistics, let the set be denoted $x_1, x_2, \dots, x_i, \dots, x_N$.

For the point-based statistics, the set of values is stored in array and then the statistics are calculated. This process is coded in subroutine CALPDFCDF.

The individual statistical parameters are discussed further.

PDF

The following steps outline the process IGW 3 uses to calculate the PDF:

- 1) Define X_{max} to be equal to $\text{MAX}(x_1, x_2, \dots, x_i, \dots, x_N)$ where the MAX function extracts the maximum value in the associated set;
- 2) Define X_{min} to be equal to $\text{MIN}(x_1, x_2, \dots, x_i, \dots, x_N)$ where the MIN function extracts the minimum value in the associated set;
- 3) Calculate Δx as

$$\Delta x = \frac{X_{\max} - X_{\min}}{M} \quad (4.3.1-1)$$

where M is the number of intervals (user specified);

- 4) Sum the total number of data in each interval $[x_j, x_{j+1}]$ where $j=1, 2, \dots, M$. These values are assigned to n_j . Note that

$$\sum_{j=1}^M n_j = N \quad (4.3.1-2)$$

- 5) Calculate the PDF as

$$p(x_j) = \frac{n_j}{N} \quad (4.3.1-3)$$

CDF

The CDF is derived from the PDF through

$$c(x) = \int_{X_{\min}}^x p(x) dx \quad (4.3.1-4)$$

which is calculated in IGW 3 as

$$c(x_j) \approx \Delta x \sum_{i=1}^j p(x_i) \quad (4.3.1-5)$$

MEAN

The mean is calculated as

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N x_i \quad . \quad (4.3.1-6)$$

MEDIAN

The median of a probability distribution function $p(x)$ is defined as

$$\int_{-\infty}^{X_{med}} p(x)dx = 0.5 = \int_{X_{med}}^{\infty} p(x)dx \quad . \quad (4.3.1-7)$$

This implicit equation is solved iteratively to find X_{med} , the value for which larger and smaller values of x are equally probable.

MODE

The mode of a probability distribution function $p(x)$ is defined as

$$p(x) \Big|_{x_{mod}} = p_{MAX} \quad . \quad (4.3.1-8)$$

This implicit equation is solved by finding the maximum value of $p(x_j)$ which gives $X_{mod} = x_j$. X_{mod} is the value of x where the probability distribution is at its maximum value.

STANDARD DEVIATION

The standard deviation is calculated by

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{X})^2} \quad . \quad (4.3.1-9)$$

AVERAGE DEVIATION

The average deviation is calculated by

$$ADev = \frac{1}{N} \sum_{i=1}^N |x_i - \bar{X}| \quad . \quad (4.3.1-10)$$

SKEWNESS

The skewness is calculated by

$$Skew = \frac{1}{N} \sum_{i=1}^N \frac{(x_i - \bar{X})^3}{\sigma} \quad . \quad (4.3.1-11)$$

KURTOSIS

The kurtosis is calculated by

$$Kurt = \left\{ \frac{1}{N} \sum_{i=1}^N \frac{(x_i - \bar{X})^4}{\sigma} \right\} - 3 \quad . \quad (4.3.1-12)$$

4.3.2 FIELD BASED STATISTICS

The field-based statistics are based upon a random field. For purposes of describing the field-based statistics, let the random field be denoted $f(x,y)$.

For the field-based statistics, $\overline{f'(x, y)g'(x_0, y_0)}^{(k)}$ (defined below) is stored in the COV() array. The process of statistical calculations is coded in subroutine COVHC.

The individual statistical parameters are discussed further.

RECURSIVE MEAN

The recursive mean is calculated by

$$\overline{F}^{(k)}(x, y) = \frac{\overline{F}^{(k-1)}(x, y)(k-1) + f(x, y)}{k}, \quad k = 1, 2, \dots, N_k \quad (4.3.1-13)$$

where k is the realization index number. The total number of realizations, N_k , is set by the user and can theoretically be infinity.

RECURSIVE VARIANCE

The recursive variance is calculated by

$$\sigma^{(k)}(x, y) = \frac{\sigma^{(k-1)}(x, y)(k-1) + [f(x, y) - \overline{F}^{(k)}(x, y)]^2}{k}, \quad k = 1, 2, \dots, N_k \quad (4.3.1-14)$$

RECURSIVE COVARIANCE

Considering a second random field, $g(x, y)$, the covariance between point $P_f(x, y)$ and $P_g(x_0, y_0)$ is expressed as

$$\begin{aligned} \overline{f'(x, y)g'(x_0, y_0)}^{(k)} &= \frac{\overline{f'(x, y)g'(x_0, y_0)}^{(k-1)}(k-1)}{k} \\ &+ \frac{[f(x, y) - \overline{F}^{(k)}(x, y)][g(x_0, y_0) - \overline{G}^{(k)}(x_0, y_0)]}{k} \end{aligned} \quad (4.3.1-15)$$

$$k = 1, 2, \dots, N_k$$

If a point involved in the covariance calculation is non-nodal, then the bilinear interpolation scheme (refer to **Section 3.2.2.1**) is used to obtain $f(x, y)$, $\overline{F}(x, y)$ or $g(x_0, y_0)$, $\overline{G}(x_0, y_0)$.

4.4 ADDITIONAL MONTE CARLO SIMULATION INFORMATION

Some additional information about MC simulations is presented in the following subsections.

4.4.1 COV ARRAY

The COV array, which stores the 25 pair of covariances (based on $\log K$, h , v_x , v_y , and C), is allocated as a dynamic array in the source code (these are only calculated when explicitly instructed to through the user input). Storing values to or retrieving them from the array involves a very precise procedure that is implemented to save memory in part by reducing the interaction, or number of calls, to the array (in the VF code) by the VB code, and from the array (in the VF code) to the VB code.

4.4.2 LIMITATION

Currently, hydraulic conductivity is the only variable that can be considered random in terms of the MC simulation. All other input parameters are considered deterministic.

4.5 MONTE CARLO SIMULATION FLOW CHART

Figure 4.5-1 shows a flow chart that lists the steps involved in the IGW 3 MC simulation procedure.

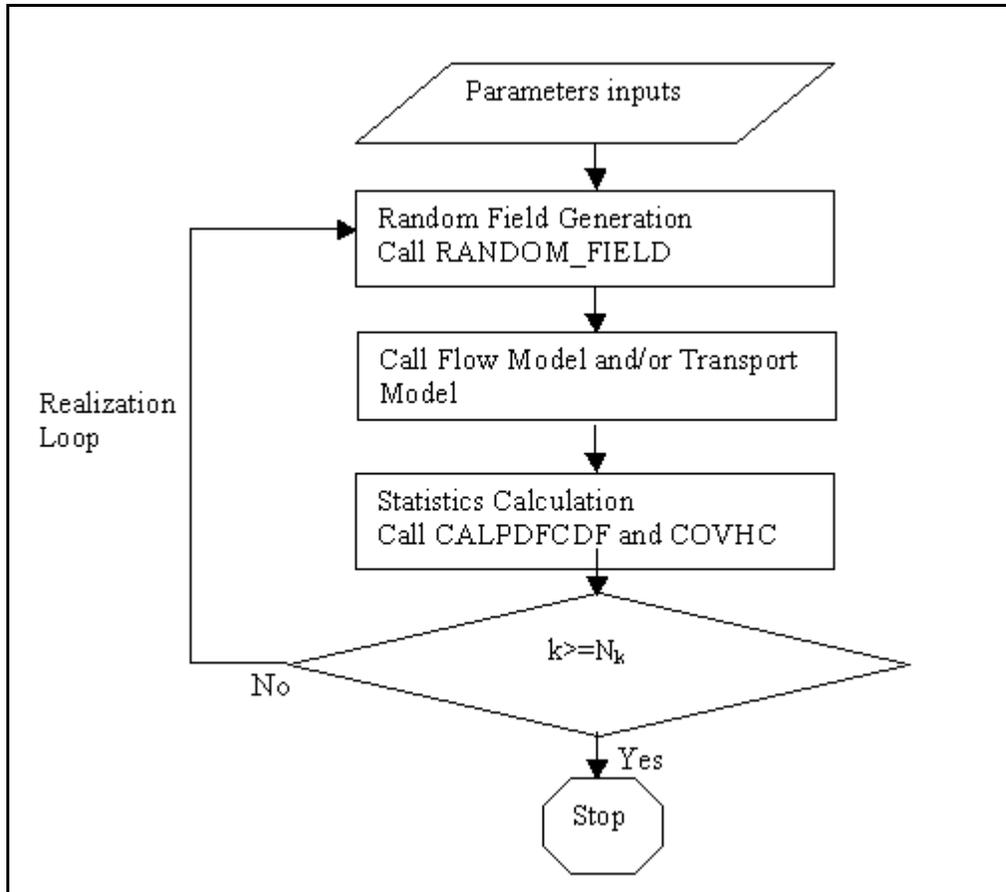


FIGURE 4.5-1
Flow chart for the IGW 3 MC simulation procedure

CHAPTER 5: PROFILE MODEL

The profile model employed in IGW 3 is more robust than most other profile models. The model is referred to as the PV2-D model and incorporates both the plane and profile models in one. **Figure 5-1** shows the conceptual differences between full 3-D, PV2-D, and the traditional plane and profile model.

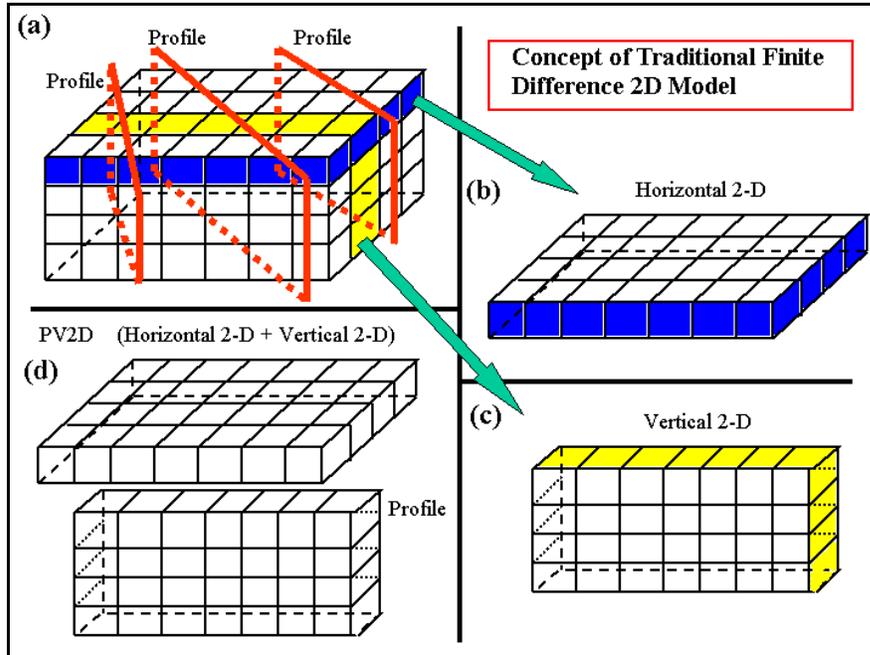


FIGURE 5-1
Conceptual representation of the full 3-D, PV2-D and traditional 2-D models – need to fix picture via David

In terms of modeling in 3-D, the PV2-D model achieves better results than either the traditional areal or profile models could separately². The new techniques allow the profile models to present more 3-D-like results and free the profile models to be drawn across flow lines without being completely inaccurate.

Formulation of an IGW 3 profile model is based upon the following: 1) all of the information such as location, hydraulic features, and boundary conditions comes from the areal 2-D model (and user input, where appropriate), and 2) because the areal 2-D model can provide information for only the surface node in the profile model, where this information cannot be calculated by the profile model it will be assumed to be the same for all nodes below the surface (-z direction) as it is for the surface node.

5.1 GOVERNING EQUATION

Take the slice shown in **Figure 5.1-1** as the reference for discussion. Note that **Figure 5.1-2** is a conceptualization of the areal 2-D model from which the slice was generated (and shows the location of the slice).

² This does not mean that the PV2-D model will duplicate 3-D model results. It means only that the results are generally better and more representative of 3-D situations than the results obtained through traditional modeling techniques. **Section 11.3** presents a number of comparative cases that illustrate the effectiveness of the PV2-D model and its limitations.

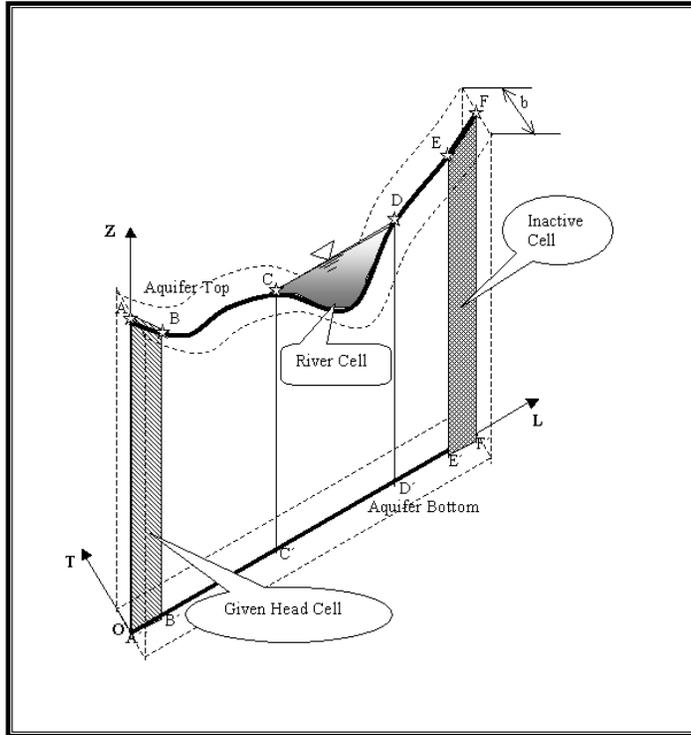


FIGURE 5.1-1
A sample vertical slice

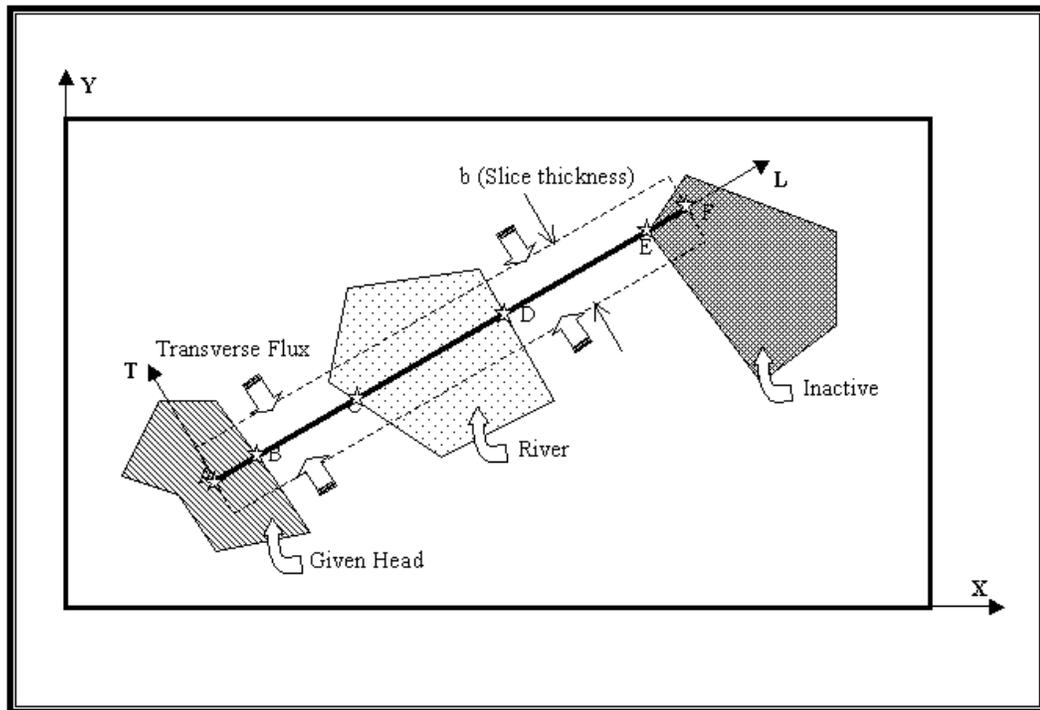


FIGURE 5.1-2
Conceptualization of the areal model in which the slice from Figure 5.1-1 was made

Considering the L-T-Z coordinate system shown in **Figure 5.1-1**, the governing equation for groundwater flow (**Equation 2.2-1**) can be written as

$$\begin{aligned}
S_s \frac{\partial h}{\partial t} = & \frac{\partial}{\partial X_L} (K_{LL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_L} (K_{LZ} \frac{\partial h}{\partial X_Z}) + \frac{\partial}{\partial X_L} (K_{LT} \frac{\partial h}{\partial X_T}) \\
& + \frac{\partial}{\partial X_Z} (K_{ZL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_Z} (K_{ZZ} \frac{\partial h}{\partial X_Z}) + \frac{\partial}{\partial X_Z} (K_{ZT} \frac{\partial h}{\partial X_T}) \quad . \quad (5.1-1) \\
& + \frac{\partial}{\partial X_T} (K_{TL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_T} (K_{TZ} \frac{\partial h}{\partial X_Z}) + \frac{\partial}{\partial X_T} (K_{TT} \frac{\partial h}{\partial X_T}) + q_s
\end{aligned}$$

If **Equation 5.1-1** is integrated from $X_T = -b/2$ to $X_T = b/2$ in the transverse direction, the result is

$$\begin{aligned}
S \frac{\partial h}{\partial t} = & \frac{\partial}{\partial X_L} (T_{LL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_L} (T_{LZ} \frac{\partial h}{\partial X_Z}) \\
& + \frac{\partial}{\partial X_Z} (T_{ZL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_Z} (T_{ZZ} \frac{\partial h}{\partial X_Z}) + Q_{Tr} + Q_s \quad (5.1-2)
\end{aligned}$$

where

$$Q_{Tr} = \int_{-b/2}^{b/2} \left[\frac{\partial}{\partial X_T} (K_{TT} \frac{\partial h}{\partial X_T}) \right] dX_T \quad . \quad (5.1-3)$$

Q_{Tr} represents the net transverse direction flux entering and leaving the slice (calculated from the areal 2-D model). **Equation 5.1-2** is based on the following assumptions

$$\frac{\partial}{\partial X_L} (K_{LT} \frac{\partial h}{\partial X_T}) = 0 \quad , \quad (5.1-4)$$

$$\frac{\partial}{\partial X_T} (K_{TL} \frac{\partial h}{\partial X_L}) = 0 \quad , \quad (5.1-5)$$

$$\frac{\partial}{\partial X_Z} (K_{ZT} \frac{\partial h}{\partial X_T}) = 0 \quad , \quad (5.1-6)$$

and

$$\frac{\partial}{\partial X_T} (K_{TZ} \frac{\partial h}{\partial X_Z}) = 0 \quad . \quad (5.1-7)$$

The governing equation for the profile model (**Equation 5.1-2**) is solved by the methods presented in **Chapter 2**. Calculation of Q_{Tr} is implemented in subroutine FLUXFORVMD in the source code. The procedure used to solve **Equation 5.1-2** is coded in subroutine TSHEAD.

5.2 DISCRETIZING CONCEPTUAL FEATURES

When a portion of the profile model crosses the conceptual features defined in the areal 2-D model, those features are imported into the profile model. The importing process is done based on the numerical parameters evaluated at the nodal points.

There are two procedures used to import the data into the profile model. One deals with point-related features and the other with line related features. These two procedures are discussed in the following subsections.

5.2.1 APPROXIMATION OF POINT RELATED FEATURES

Figure 5.2.1-1 illustrates an example of mapping a point related feature to the nodes.

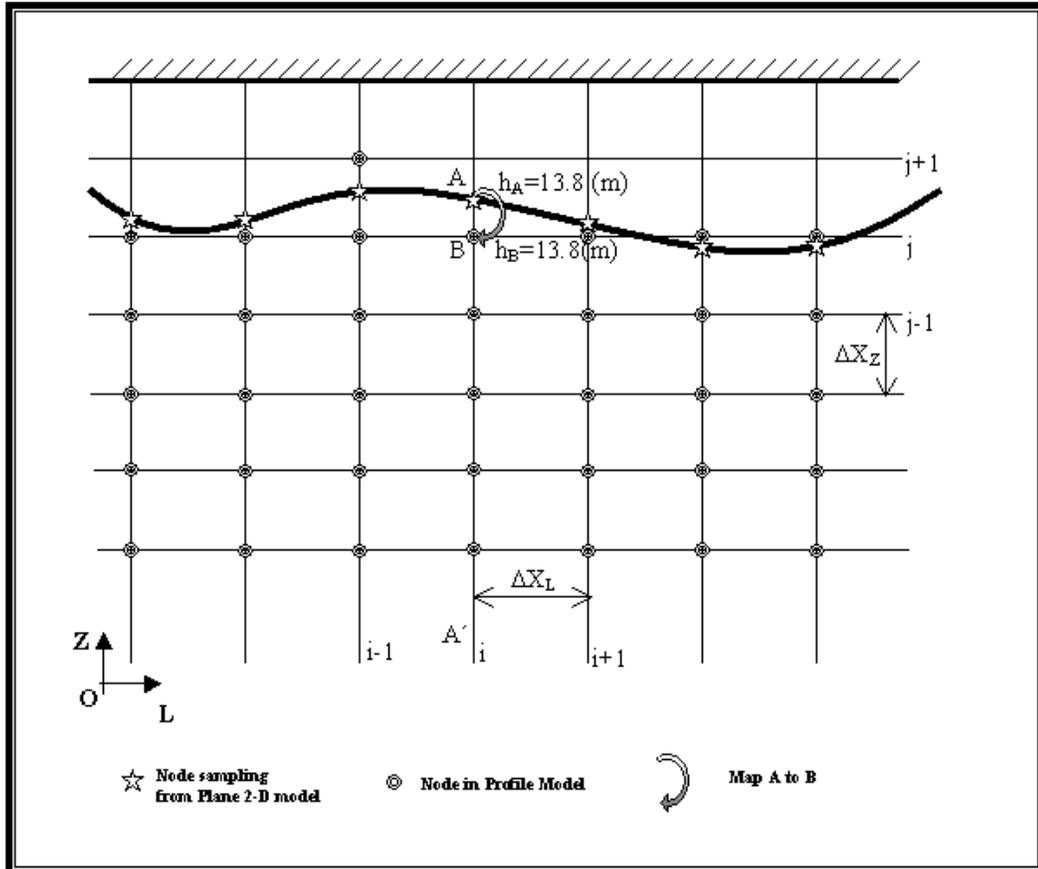


FIGURE 5.2.1-1
An example of mapping a point related feature to nodes

This mapping process involves two steps:

- 1) Determine the location of the node nearest to the sampling node A (that lies on the i,j indexed nodal grid) based on the coordinates of A (X_L^A, X_Z^A) and the grid size ($\Delta X_L, \Delta X_Z$). This process is accomplished through

$$i = \text{int} \left(\frac{X_L - X_L^0}{\Delta X_L} \right) + 1 \quad (5.2.1-1)$$

and

$$j = \text{int} \left(\frac{X_Z - X_Z^0}{\Delta X_Z} \right) + 1 \quad (5.2.1-2)$$

where (X_L^0, X_Z^0) are the coordinates of the origin in the profile model computational domain.

In this case the nearest node is B.

- 2) Assign the parameter value, p_a , at node A to node B and all other nodes on line A-A'. This process is accomplished through

$$p_B = p_A \quad (5.2.1-3)$$

and

$$p_{il} = p_A, \quad l = j, \quad j-1, \dots, j_{A'} \quad (5.2.1-4)$$

5.2.2 APPROXIMATION OF POLYLINE RELATED CONCEPTUAL FEATURES

Figure 5.2.2-1 illustrates an example of mapping a polyline related feature to the nodes. Polyline related features include: aquifer top, aquifer bottom, water table, river stage, river bed elevation, river leakance, drain elevation, and drain leakances. Note that although wells are point-related features, their associated screen lengths are treated in a manner similar to that of a polyline-related feature.

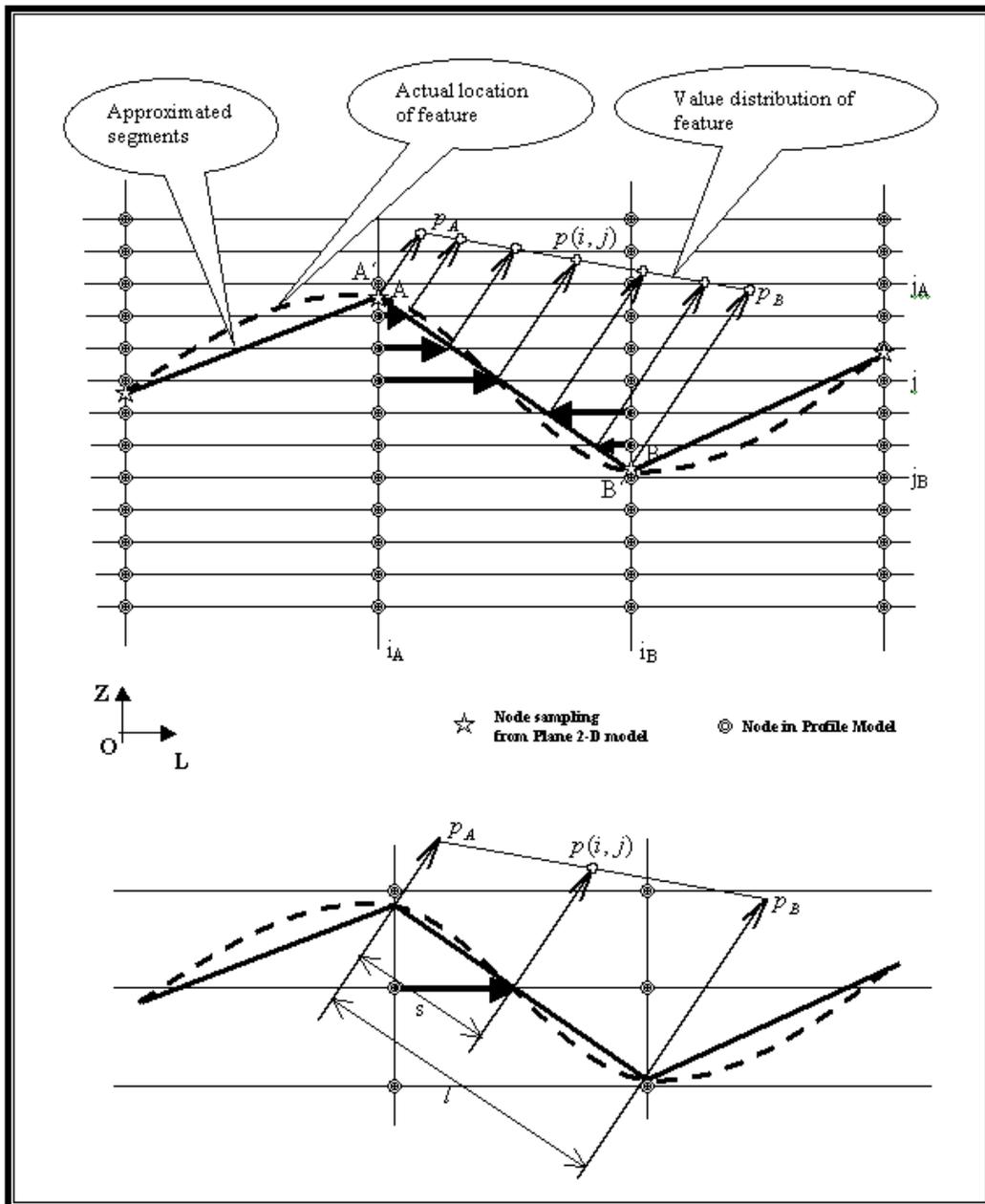


FIGURE 5.2.2-1
An example of mapping a polyline related feature to nodes

This mapping process involves four steps:

- 1) The drawn polyline is divided into N_p segments with the endpoints set to fall on the vertical grid lines (Z-direction). The distance between the vertical grid lines is determined by ΔX_L (which is determined from the size of the domain and the user-specified grid resolution).

This process is done in the VB interface as the line is drawn.

- 2) For each segment, the endpoints are mapped to the nearest nodes.

Consider the segment AB. The endpoints are mapped to A' and B' through

$$i_{A'} = \text{int} \left(\frac{X_L^A - X_L^0}{\Delta X_L} \right) + 1 \quad , \quad (5.2.2-1)$$

$$j_{A'} = \text{int} \left(\frac{X_Z^A - X_Z^0}{\Delta X_Z} \right) + 1 \quad , \quad (5.2.2-2)$$

$$i_{B'} = \text{int} \left(\frac{X_L^B - X_L^0}{\Delta X_L} \right) + 1 \quad , \quad (5.2.2-3)$$

and

$$j_{B'} = \text{int} \left(\frac{X_Z^B - X_Z^0}{\Delta X_Z} \right) + 1 \quad . \quad (5.2.2-4)$$

3) The values at the endpoints A and B are assigned to A' and B' through

$$p(i_{A'}, j_{A'}) = p_A \quad (5.2.2-5)$$

and

$$p(i_{B'}, j_{B'}) = p_B \quad . \quad (5.2.2-6)$$

4) If the segment in question crosses multiple horizontal (L-direction) grid lines, then

$$ABS(j_{A'} - j_{B'}) \geq 2 \quad (5.2.2-7)$$

is true and the nodes in the j index between A' and B' are determined using the linear interpolation scheme where

$$p(i, j) = \left(1 - \frac{s}{l}\right) p_A + \frac{s}{l} p_B \quad , \quad \begin{cases} i = i_{A'}, i_{B'} \\ j = j_{A'} + 1, \dots, i_{B'} - 1 \end{cases} \quad . \quad (5.2.2-8)$$

For screen thickness (Z_{sT} and Z_{sB}) interpolation, the flow rate of the well (Q_{well}) is uniformly distributed over the nodes according to

$$Q_j = \frac{Q_{well}}{ABS(j_{B'} - j_{A'}) + 1} \quad , \quad j = j_{A'}, \dots, j_{B'} \quad (5.2.2-9)$$

5.3 DETERMINATION OF COMPUTATIONAL DOMAIN

The computational domain in the profile model depends on a combination of aquifer top (T), aquifer bottom (B), bed elevation of river or drains (E), and the water table (W). These features in

relation to a sample domain are shown in **Figure 5.3-1** with the features denoted by the corresponding letter above.

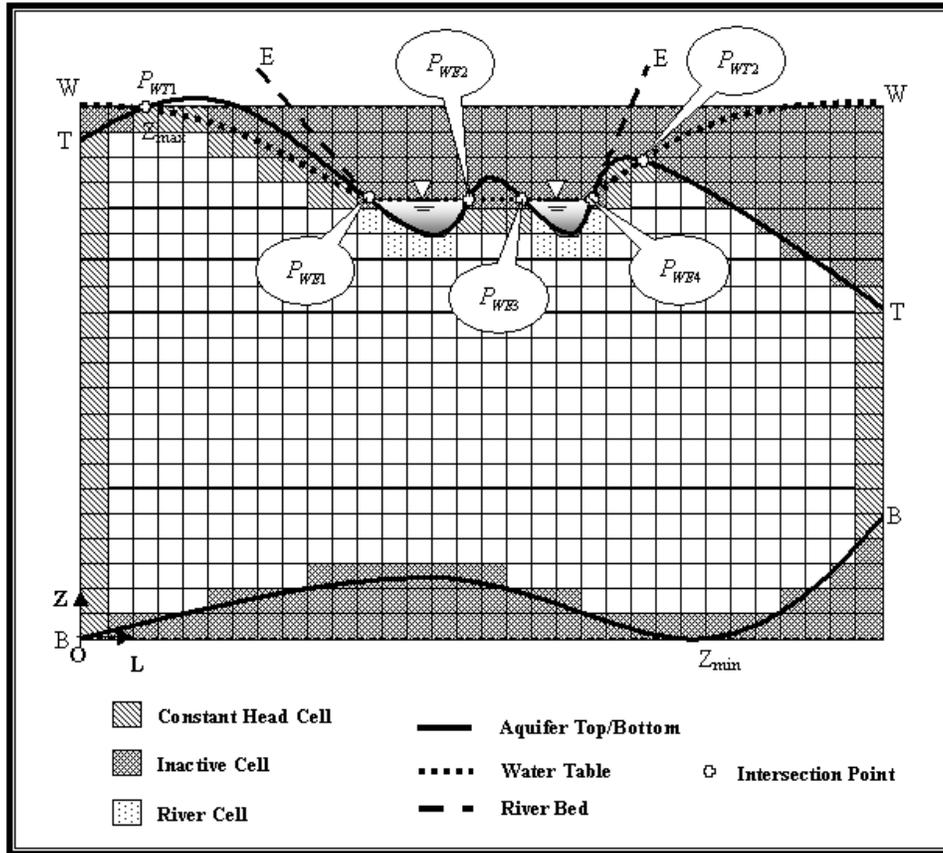


FIGURE 5.3-1
A sample profile model computational domain with boundary features denoted

The methods used to determine the computational boundaries from the bounding features are described in the following subsections.

5.3.1 UPPER BOUNDARY

The upper boundary of the computational domain is set by T, E, and W. For any point, the boundary is set by the lowest of these. Thus, where the T, E, and W lines cross, the boundary-controlling feature may change. These crossing, or intersection, points are determined by the software. Intersection points corresponding to **Figure 5.3-1** are given as $\{P_{WT1}, P_{WT2}\} = \overline{WW} \cap \overline{TT}$ and $\{P_{WE1}, P_{WE2}, P_{WE3}, P_{WE4}\} = \overline{WW} \cap \overline{EE}$. The upper boundary can be described as T- P_{WT1} - P_{WE1} - P_{WE2} - P_{WE3} - P_{WE4} - P_{WT2} -T. The topmost portion of the boundary is denoted Z_{max} and corresponds to P_{WT1} in this case.

5.3.2 LOWER BOUNDARY

The lower boundary is typically set by the aquifer bottom, B. In the special case of a river that extends below the aquifer bottom, the bottom is then set in a similar fashion as described in **Section 5.3.1**. The bottommost portion of the boundary is denoted Z_{min} .

5.3.3 DISPLAYED BOUNDARY

The displayed computational boundary is determined by the vertical left-hand and right-hand boundaries (corresponding to the user-defined endpoints of the profile model), and two lines equal to Z_{min} and Z_{max} . The grid is then laid out over this entire domain.

5.3.4 INACTIVE CELLS

All cells that lie outside of upper and lower boundaries are set to inactive by the software.

5.3.5 BOUNDARY CONDITIONS

The boundary conditions on the left- and right-hand sides are either constant head or no flow. The lower boundary is always set to a no flow condition. Because the upper boundary may consist of a number of conceptual features it will take on different boundary conditions over certain portions of it. For example, in terms of **Figure 5.3-1**, the features and corresponding boundary conditions presented in **Table 5.3-1** are applicable.

TABLE 5.3-1
Conceptual features and boundary conditions corresponding to the upper boundary in Figure 5.3-1

Section	Feature	Boundary Condition
T-P _{WT1}	Aquifer Top	No Flow
P _{WT1} -P _{PWE1}	Water Table	Constant Head
P _{PWE1} -P _{PWE2}	River Bed	River Cell
P _{PWE2} -P _{PWE3}	Water Table	Constant Head
P _{PWE3} -P _{PWE4}	River Bed	River Cell
P _{PWE4} -P _{PWT2}	Water Table	Constant Head
P _{PWT2} -T	Aquifer Top	No Flow

5.4 ADDITIONAL INFORMATION

Some additional information about IGW 3 profile models is presented in the following subsections.

5.4.1 SOURCE CODE

All of the techniques described in this chapter are coded in subroutine NEWCELLCNCPT.

5.4.2 ANISOTROPY RATIO

In order to have the most numerically well behaved solution to the profile model, the ratio of grid spacing the longitudinal and vertical directions, $\Delta X_L/\Delta X_Z$, is set by

$$\frac{\Delta X_L}{\Delta X_Z} = \sqrt{R_{anisf}} \quad (5.4.2-1)$$

where

$$R_{anisf} = \frac{K'_{xx}}{K'_{zz}} \quad (5.4.2-2)$$

This parameter may be adjusted through user input.

5.4.3 RIVER CELL TREATMENTS

River cells are assigned to the cells that are adjacent to the riverbed cells. If the case is that the riverbed is larger than the river stage, as illustrated in **Figure 5.4.3-1**, then some of the river cells are turned into drain cells.

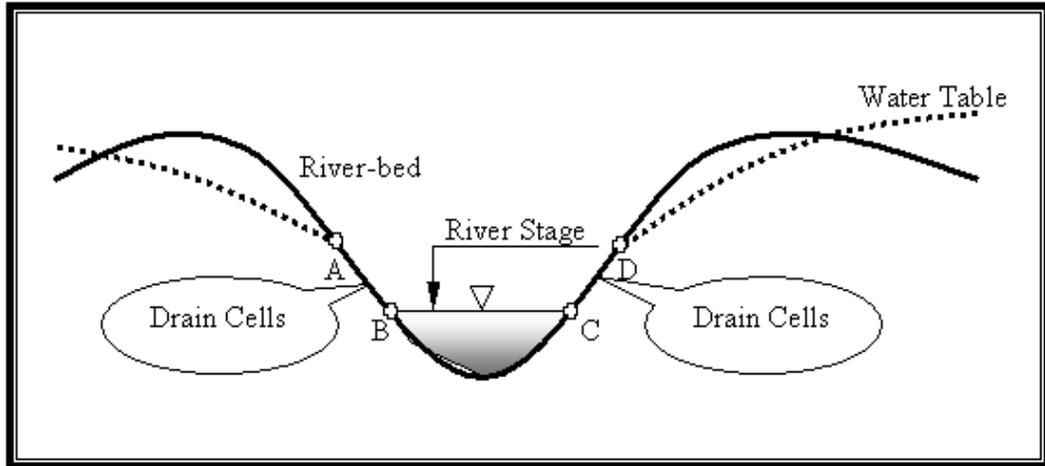


FIGURE 5.4.3-1
Example of riverbed larger than river stage

5.4.4 TRANSIENT MODEL

The IGW 3 model can be classified as quasi-transient. This is to say the profile model will update at each time step if the areal 2-D model is using transient state, but it will essentially be a new steady-state model at each time step. Each time the areal 2-D model updates, the profile model will be solved based on the new parameters provided by the areal model and no information from the previous profile model state is used in the next profile model solution. This is done due to the fact that a transient water table requires a computational domain that is constantly fluctuating and presents a challenge in terms of software coding for a truly transient representation.

5.5 PROFILE MODEL SOLVER FLOW CHART

Figure 5.5-1 shows a flow chart that lists the steps involved in the IGW 3 profile model solution procedure.

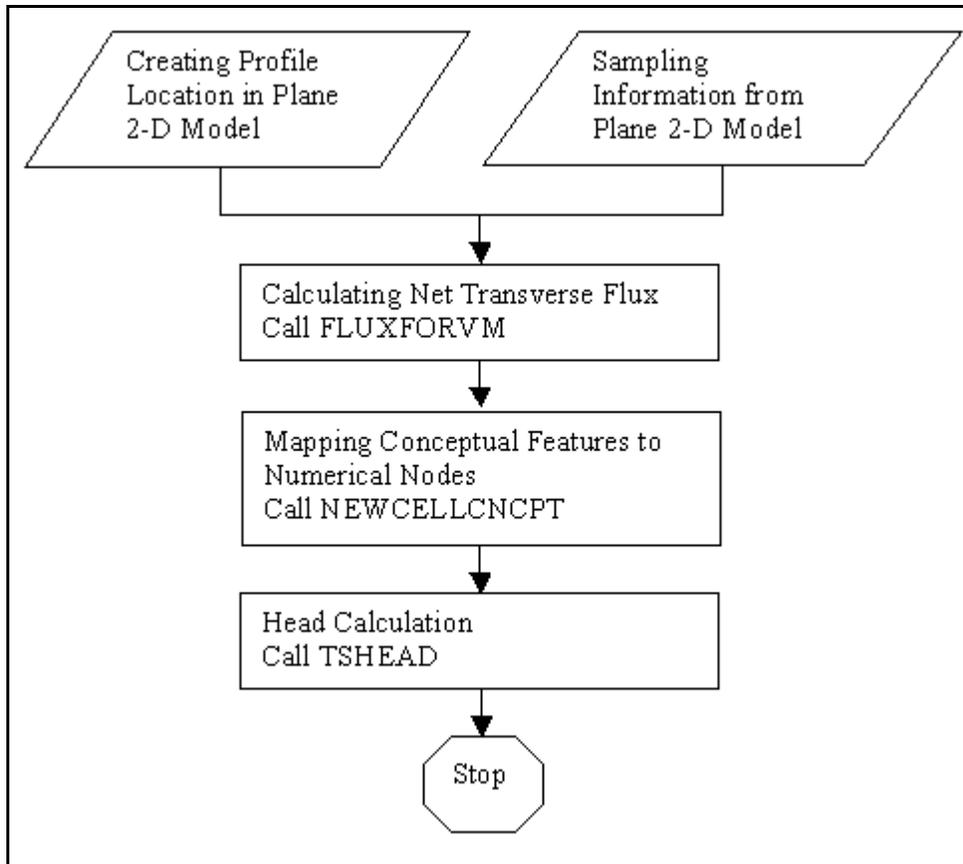


FIGURE 5.5-1
Flow chart for the IGW 3 profile model solution procedure

CHAPTER 6: SCATTER POINT STATISTICS

This chapter describes the methods available to IGW 3 users when extrapolating/interpolating scatter point data into variable fields.

6.1 INVERSE DISTANCE METHOD MATHEMATICS

The inverse distance weighting (IDW) method is based on the premise that a given point for which a value is to be determined should be influenced most by points that are closest to it.

IGW 3 employs ‘Shepard’s method’ which uses

$$F(x, y) = \sum_{i=1}^n w_i f_i \quad (6.1-1)$$

as the controlling equation. In the equation, n is the number of scatter points in the set, i is the index value for each scatter point, f_i is the value at each scatter point, and w_i is the weight factor for each point.

The weight factor is defined as

$$w_i = \frac{h_i^{-p}}{\sum_{j=1}^n h_j^{-p}} \quad (6.1-2)$$

where p is the inverse distance exponent, h_i is the distance from the scatter point to the interpolation point, and the summation term the sum of these distances.

6.2 KRIGING METHOD MATHEMATICSⁱ

The kriging method is basically a set of linear regressions. Based on a user defined variogram model, the kriging method aims to minimize estimation variance. It is based on the premise that points that are close together have a high degree of spatial correlation but those far apart are statistically independent.

There are two general types of kriging: ordinary and universal. Ordinary kriging is explicitly programmed into IGW 3 as it is also the backbone of the universal kriging method. The universal kriging method is the application of ordinary kriging to previously detrended data and can be activated in IGW 3 by implementing both ‘Regression’ and kriging to a set of scatter points.

ORDINARY KRIGING

The equation used in kriging is

$$F(x, y) = \sum_{i=1}^n w_i f_i \quad (6.2-1)$$

where n is the number of scatter points in the set, i is the scatter point index, f is the value for each individual point in the set, and w is the weighting factors for each individual point in the set.

The weighting factors at a particular point are determined through the solution of a set of linear equations. The general form of the equation (for a point p) is

$$\sum_{j=1}^n w_j S(d_{i,j}) = S(d_{i,p}), i = 1..n \quad (6.2-2)$$

Thus for each interpolated point there exists a set of n equations that each have n unknowns. For **Equation 6.2-2**, the S terms are the values of the variogram model evaluated at a distance equal to the distance between the two subscript points.

The set of linear equations includes

$$\sum_i^n w_i = 1 . \quad (6.2-3)$$

Equation 6.2-2 is modified with a slack variable to make the number of equations and unknowns equivalent and becomes

$$\sum_{j=1}^n w_j S(d_{i,j}) + Slack = S(d_{i,p}), i = 1..n \quad (6.2-4)$$

The equations are solved simultaneously to find the w values. Then the interpolated value, f_p , is determined by

$$f_p = \sum_{i=1}^n w_i f_i . \quad (6.2-5)$$

6.3 UNCONDITIONAL SIMULATION MATHEMATICS

The unconditional simulation option generates a random field based upon the scatter point statistics only. Thus the resulting field may not have values that exactly correspond to the scatter point values at the particular scatter point locations.

The user may implement the ‘Spectral Algorithm’, the ‘Sequential Gaussian Simulation’, or the ‘Turning Bands Algorithm’ as the simulation method. These methods are discussed in **Sections 7.1, 7.2 and 7.3**, respectively.

In terms of the statistics, the user may specify them directly using the ‘Random Field Options’ window (refer to **Appendix F-III** in the *IGW 3 User’s Manual*) or automatically generate and manually tweak them using the ‘Variogram’ window (refer to **Chapter 8** of this document and **Appendix F-II** in the *IGW 3 User’s Manual*).

6.4 CONDITIONAL SIMULATION MATHEMATICS

The conditional simulation option generates a random field based upon the scatter point statistics and the actual scatter point values. A random field is generated which has values that exactly correspond to the scatter point values at the particular scatter point locations.

The user may implement the ‘Spectral Algorithm’, the ‘Sequential Gaussian Simulation’, or the ‘Turning Bands Algorithm’ as the simulation method. These methods are discussed in **Sections 7.1, 7.2 and 7.3**, respectively.

In terms of the statistics, the user may specify them directly using the ‘Random Field Options’ window (refer to **Appendix F-III** in the *IGW 3 User’s Manual*) or automatically generate and manually tweak them using the ‘Variogram’ window (refer to **Chapter 8** of this document and **Appendix F-II** in the *IGW 3 User’s Manual*).

CHAPTER 7: RANDOM FIELDS

A number of algorithms may be implemented in the random field generation process. The desired algorithm is selected (and the associated parameters defined) using one of two different windows in the software: 1) the ‘Option of Unconditional Random Field (Attr.)’ window, and 2) the ‘Random Parameters’ window. The first is used when defining an unconditional random field directly for a zone (refer to **Appendix B-I** in the *IGW 3 User’s Manual* for a description of the window/interface). The second is used when defining an unconditional or conditional random field for a set of scatter points within a zone (refer to **Appendix F-III** in the *IGW 3 User’s Manual* for a description of the window/interface).

There are a number of parameters that are involved in most of the algorithms. These are defined in **Table 7-1** (in terms of their IGW 3 interface names).

TABLE 7-1
Common random field parameters

Parameter	Definition
LambdaX	x
LambdaY	x
Seed	x
(Theoretical) Variance	x
Angle(X-North Clockwise)	x
Nugget	x

The specifics of each algorithm are discussed in the appropriate section.

7.1 SPECTRAL ALGORITHM

The spectral algorithm used in generating random fields is that which is implemented in the Monte Carlo simulation process, the Fast Fourier Transform Technique. This technique is discussed in **Section 4.1.1**.

Table 7.1-1 lists the covariance functions that may be used in the spectral algorithm.

TABLE 7.1-1
Covariance functions available for use in the spectral algorithm

Function Type	Function	Form
Anisotropic	Bell	x
	Exponential	x
Isotropic	whittle	x
	Mizzel-A	x
	Mizzel-B	x

block average – allowed for scatter points

conditional vs unconditional

7.2 SEQUENTIAL GAUSSIAN ALGORITHM

The sequential gaussian algorithm involves the following steps:

- 1) Transform the data into a normal distribution;
- 2) Develop a variogram model for the data;
- 3) Select a node at random and kriging the value at that node (this gives the kriged variance);
- 4) Draw a random number from a gaussian distribution that has a variance equivalent to the kriged variance and a mean equivalent to the kriged value (this is the simulated number for the node);
- 5) Repeat steps 3-5 (for the kriging portion, include all previously simulated nodes to preserve the spatial variability as modeled in the variogram);
- 6) When all nodes have been simulated, back transform to the original distribution (this is the first realization); and
- 7) Repeat using a different random number sequence to generate multiple realizations.

need the mathematics that go along with each step...
conditional vs. unconditional...

Table 7.2-1 lists the covariance functions that may be used in the sequential gaussian algorithm.

TABLE 7.2-1
Covariance functions available for use in the sequential gaussian algorithm

Function Type	Function	Form
Anisotropic	Gaussian	x
	Exponential	x
	Spherical	x
	Hole-Exponential	x
	bombing model	x

7.3 TURNING BAND ALGORITHM

unconditional only on its own
xxxxx – steps and mathematics

Table 7.3-1 lists the variogram model that may be used in the turning bands algorithm.

TABLE 7.3-1
Covariance functions available for use in the turning bands algorithm

Variogram Model	Form
Spherical	x
Exponential	x

made conditional through ordinary kriging

7.4 SIMULATED ANNEALING

xxxxx – steps and mathematics –GSLIB says this is conditional only... always honors data values... how is it then implemented in the ‘Unconditional Random Field (Attr.)’ window????

Table 7.4-1 lists the covariance functions that may be used in the simulated annealing process.

TABLE 7.4-1

Covariance functions available for use in the simulated annealing

Function Type	Function	Form
Anisotropic	Gaussian	x
	Exponential	x
	Spherical	x
	Hole-Exponential	x
	Hole-Gaussian	x

not available in scatter point simulation ...

need to add acronym references from this section to the beginning

use correct and notation

italics

dimensions

eq, tables, figs, sections

n-ne

CHAPTER 8: VARIOGRAM^{i,ii,3}

The IGW 3 variogram building process that is essential to scatter point kriging and simulation is quite powerful in that it provides for both automatic optimization and manual model adjustment.

There are three implicit options for defining spatial statistics in IGW 3: 1) fully user specified, 2) user adjusted automatic optimization, and 3) fully automatic optimization.

The fully user specified option uses the ‘Input Parameters’ interface or ‘Random Parameters’ interface (refer to *IGW 3 User’s Manual*) for kriging or simulation, respectively. This option essentially skips the formation of the experimental framework (see **Section 8.1**) from which to base the theoretical model. Thus these interfaces do not provide for any visualization of the variogram. They do however provide some additional theoretical models (as discussed in **Section 8.2**).

The other two options make use of the ‘Variogram’ interface (discussed further in the IGW 3 User’s Manual). By default, ‘Automatic Optimization’ is selected and IGW 3 automatically generates all of the appropriate models and parameters. This is the fully automatic optimization option. The user adjusted automatic optimization option is implemented by selecting ‘Manual/Trial and error’ and adjusting the software determined parameters.

The paradigm of building a variogram in IGW 3 is discussed throughout the sections of this chapter.

8.1 EXPERIMENTAL VARIOGRAM

The first step in the variogram building process is specifying an experimental framework from which to fit a theoretical model. This is done through the following steps (refer to **Figure 8.1-1** for a graphical representation of the discussed parameters):

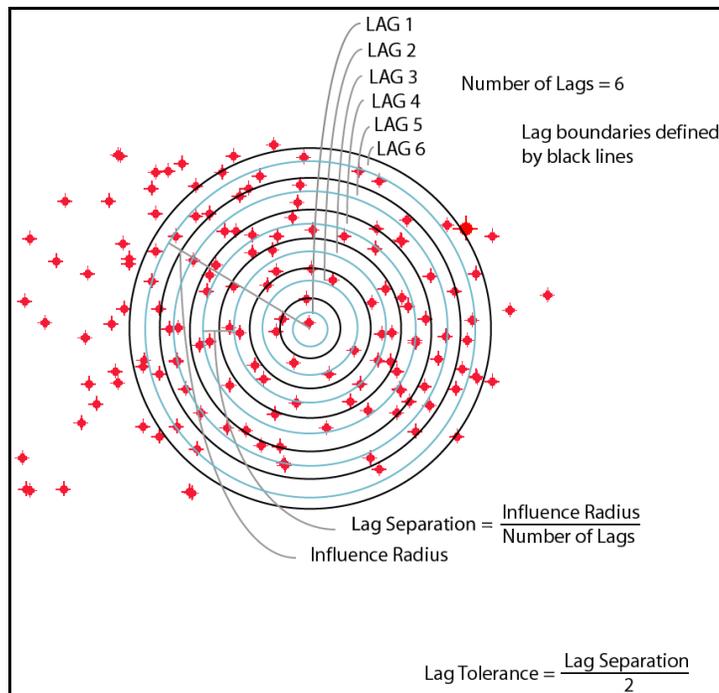


FIGURE 8.1-1
Lag parameter visualization (the red dots with crosshairs indicate scatter points)

³ The indicated references were utilized a number of times throughout this chapter.

- 1) An influence radius is specified which provides the proximity cutoff limit for considering a pair of scatter points in the analysis. The influence radius is represented by the outermost blue line in the figure.
- 2) The number of lags is specified. Each lag is centered about a lag line (represented by the blue lines in the figure). The spacing of the lag lines is determined from the influence radius and number of lags by

$$x_{Lag} = \frac{IR}{n_{Lag} - 0.5} \quad (8.1-1)$$

where x_{Lag} = the radial lag separation between each lag line [L],
 IR = the influence radius [L], and
 n_{Lag} = the number of lags.

Note that the innermost lag line is set at a radius of $\frac{x_{Lag}}{2}$ around the central point.

Each lag extends in and out (radially) a certain distance from the lag line. This distance is known as the lag tolerance. In this case, the lag tolerance for each lag is equal to $\frac{x_{Lag}}{2}$.

For example, assume that the influence radius is equal to 550 m. Six lags are defined, so from **Equation 8.1-1**, the x_{Lag} is 100 m. The lag line for Lag 6 (the influence radius) is at 550 m. Accordingly, Lag 6 covers $\frac{x_{Lag}}{2}$, 50 m, in and out from the lag line, so it covers from 500 m to 600 m. Thus, each point that is at a radial distance of between 500 m (inclusive) and 600 m (not inclusive) from the central point is considered to be in Lag 6.

- 3) The value for each lag is computed based on the variogram type. In IGW 3, the only available variogram type is ‘Semi-variogram’. This type is defined by

$$\gamma(h) = \frac{1}{2N} \sum_{i=1}^N (f_{i,radial} - f_{i,central})^2 \quad (8.1-2)$$

where N is the number of pairs of scatter points represented by the lag, i is the index of the scatter point pairs, and the f variables represent the values at the two points (where the central indicates the central point and radial indicates the radial point – refer to **Figure 8.1-2**).

The user may choose to alter the radius of influence and the number of lags and may also choose between an isotropic or anisotropic variant of the experimental model (discussed in the following subsection).

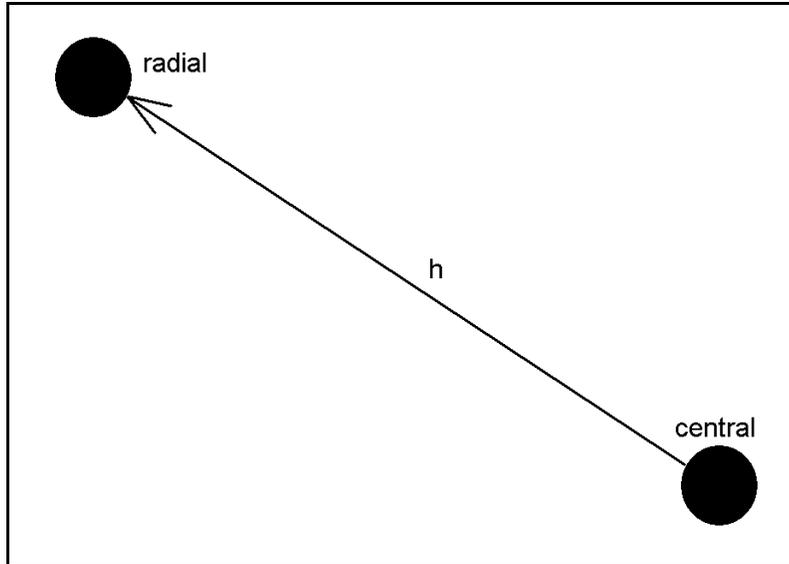


FIGURE 8.1-2
Point naming convention

8.1.1 ANISOTROPIC OPTION

If an anisotropic model is selected, an additional 3 parameters are available for the experimental framework: 1) principal angle, 2) bandwidth, and 3) angle tolerance.

These parameters are visualized in **Figure 8.1.1-1**.

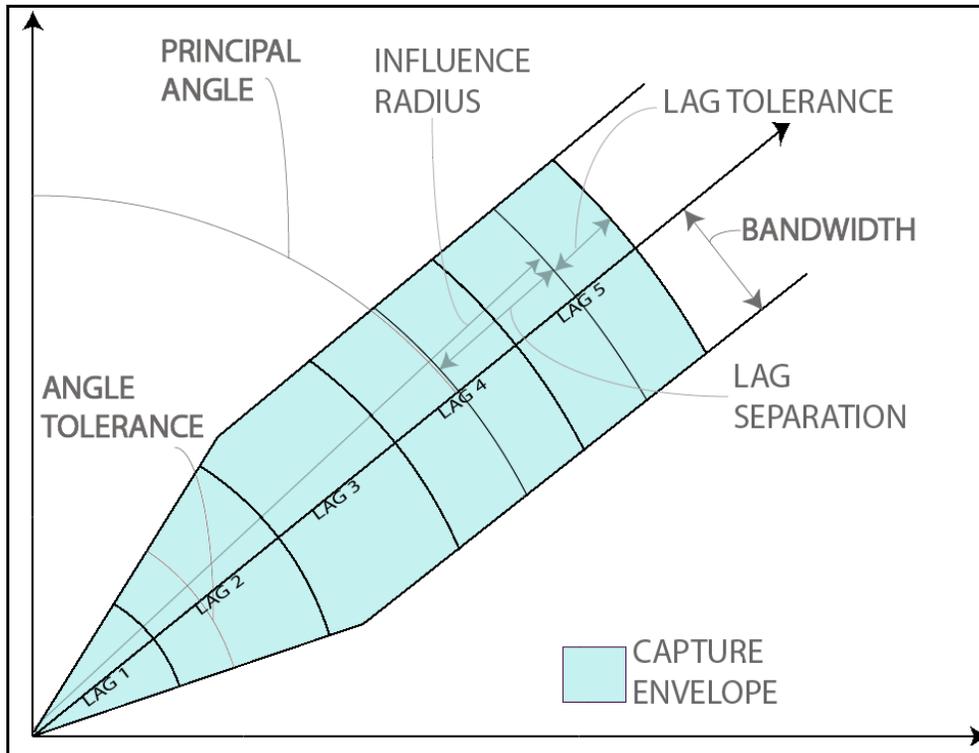


FIGURE 8.1.1-1
Anisotropic parameters involved in the experimental framework

The scatter points considered in developing the experimental framework when the ‘Anisotropic’ option is selected are limited to those falling within the capture envelope.

8.2 THEORETICAL MODELS

The second step in the variogram building process is fitting a theoretical model to the experimental framework. The model function is selected and the parameters of the function are adjusted until the best fit is achieved. This process may be automated or the model and parameters may be adjusted by the user.

The models and their associated parameters are discussed in the following subsections.

8.2.1 GENERAL PARAMETERS

There are a number of parameters that are common to a number of the models. They are 1) the nugget, 2) the sill (denoted as ‘variance’ in the ‘Variogram’ window), and 3) the range.

The nugget, a , is the minimum variance. The sill, b , is the average variance of points at a distance from the point in question that there is no correlation between the points. The range, c , is the distance from a point at which there is no correlation between the point and any others. Refer to **Figure 8.2-1** for a visual representation of these parameters.

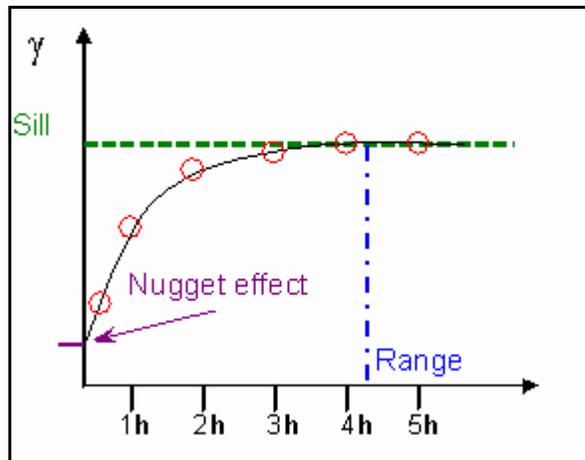


FIGURE 8.2-1
Model parameters

Note that when ‘Anisotropic’ is selected, there will be a second set of these parameters available for the second direction.

8.2.2 SPHERICAL

The spherical model is available in both the ‘Variogram’ interface and the ‘Input Parameters’ interface.

It is defined by

$$\gamma(h) = b \left(1.5 \frac{h}{c} - 0.5 \left(\frac{h}{c} \right)^3 \right) + a, \quad \text{if } \frac{h}{c} \leq c$$

$$\gamma(h) = b + a, \quad \text{if } \frac{h}{c} > c$$
(8.2.2-1)

8.2.3 EXPONENTIAL

The exponential model is available in both the ‘Variogram’ interface and the ‘Input Parameters’ interface.

It is defined by

$$\gamma(h) = b \left(1 - \exp \left(- \frac{3h}{c} \right) \right) + a$$
(8.2.3-1)

8.2.4 GAUSSIAN

The gaussian model is available in both the ‘Variogram’ interface and the ‘Input Parameters’ interface.

It is defined by

$$\gamma(h) = b \left(1 - \exp \left(- \frac{3h^2}{c^2} \right) \right) + a$$
(8.2.4-1)

8.2.5 POWER

The power model is available only in the ‘Input Parameters’ interface.

It is defined by

$$\gamma(h) = bh^c + a$$
(8.2.5-1)

where $0 < c < 2$.

8.2.6 HOLE-EXP

The hole-exp model is available only in the ‘Input Parameters’ interface.

It is defined by

$$\gamma(h) = b \left(1 - \cos \left(\frac{h}{c} \pi \right) \right) + a$$
(8.2.6-1)

8.2.7 HOLE-GAUSS

The hole-gauss model is available only in the 'Input Parameters' interface.

It is defined by

$$\gamma(h) = b \left(1 - \cos \left(\left(\frac{h}{c} \right)^2 \pi \right) \right) + a . \quad (8.2.7-1)$$

8.3 VARIOGRAM PLOT

The variogram plot is discussed in **Appendix F-II-1** of the *IGW 3 User's Manual*.

Note that the experimental variogram data are given as points in the plot. Each point represents one lag. The theoretical variogram model is displayed as a line in the plot.

CHAPTER 9: DATA ANALYSIS

Off line, exploratory analysis of scatter point data may be performed in the ‘Exploratory Data Analysis’ window. This window can be accessed through the Alternate RHP (see **Section 7.8.4.1** of the *IGW 3 User’s Manual*) and is discussed in **Appendix E** of the *IGW 3 User’s Manual*.

The following sections address some details that were not fully explained in the *IGW 3 User’s Manual*.

9.1 CONTROLLING PARAMETERS

The parameters that control the analysis (by specifying how the data are grouped) are:

- 1) the ‘Number of Intervals’,
- 2) the ‘Scatterplot Lag h’ value, and
- 3) the ‘Scatterplot Tolerance’.

As such, the data are grouped in such a way that is explained in **Section 8.1**. However, the steps are somewhat different and are given as (refer to **Figure 8.1-1** for a graphical representation of the discussed parameters and note the term ‘lag’ and ‘interval’ are equivalent):

- 1) The number of intervals is specified (‘Number of Intervals’). This is the same as the number of lag lines.
- 2) The spacing of the lag lines is specified (‘Scatterplot Lag h’). The first lag line is set at a radius of $\frac{x_{Lag}}{2}$ from the central point. Each subsequent lag line is set a distance of x_{Lag} from the previous one.
- 3) The tolerance for the lag lines is specified (‘Scatterplot Tolerance’). This specifies the distance in and out (radially) from the lag line that the corresponding lag will extend.

For example, if a lag line exists at 550 m and the scatterplot tolerance is set to 50 m, the corresponding lag will cover from 500 m to 600 m.

CHAPTER 10: SOLVER METHODSⁱⁱⁱ

The three solver methods available in the IGW 3 code are: 1) the Jacobi method; 2) the Gauss-Seidel method; and 3) the Successive Over Relaxation (SOR) method. These methods are discussed in the following sections.

10.1 JACOBI METHOD

The Jacobi method is an iterative solver in which the solution to the FD equation at a particular node is based on values from both the previous iteration level. Thus, the set of FD equations for the entire modeling domain are all solved for a particular iteration level before the next iteration level is considered.

Using **Equation 2.3.1.4-1** as an example, the resulting iterative equation based on the Jacobi method is

$$h_p^{k+1} = \frac{1}{(a_p + a_p^t + a_p^Q)} \left(\begin{array}{l} a_E h_E^k + a_W h_W^k + a_N h_N^k + a_S h_S^k + a_{NE} h_{NE}^k \\ + a_{NW} h_{NW}^k + a_{SE} h_{SE}^k + a_{SW} h_{SW}^k + S_f^Q + S_f^t \end{array} \right) \quad (10.1-1)$$

where the k superscripts correspond to the iteration level.

The software default, or user-specified, initial values are used in the first iteration. The iterations proceed using the values obtained from the previous iteration until the solution converges at each node. IGW 3 considers the solution converged when the value between an iteration and the subsequent iteration are different by less than some error tolerance. The solver can also be instructed to stop when a certain number of iterations are performed, regardless of the error criteria. The convergence criteria in IGW 3 are set in the ‘Solver’ window (refer to **Chapter 13** in the *IGW 3 User’s Manual*).

10.2 GAUSS-SEIDEL METHOD

The Gauss-Seidel method is an iterative solver in which the solution to the FD equation at a particular node is based on values from both the previous iteration level and the current iteration level. Thus, the set of FD equations for the entire modeling domain are solved in an orderly, predetermined way that allows the solutions to be applied in the same iteration level and ultimately achieve faster convergence of the overall solution.

Using **Equation 2.3.1.4-1** as an example, the resulting iterative equation based on the Gauss-Seidel method is

$$h_p^{k+1} = \frac{1}{(a_p + a_p^t + a_p^Q)} \left(\begin{array}{l} a_E h_E^k + a_W h_W^{k+1} + a_N h_N^k + a_S h_S^{k+1} + a_{NE} h_{NE}^k \\ + a_{NW} h_{NW}^k + a_{SE} h_{SE}^{k+1} + a_{SW} h_{SW}^{k+1} + S_f^Q + S_f^t \end{array} \right) \quad (10.2-1)$$

where the k superscripts correspond to the iteration level.

The software default, or user-specified, initial values are used in the first iteration. The iterations proceed using the values obtained from the previous and present iteration (as shown in **Equation 10.2-1**) until the solution converges at each node. IGW 3 considers the solution converged when

the value between an iteration and the subsequent iteration are different by less than some error tolerance. The solver can also be instructed to stop when a certain number of iterations are performed, regardless of the error criteria. The convergence criteria in IGW 3 are set in the ‘Solver’ window (refer to **Chapter 13** in the *IGW 3 User’s Manual*).

10.3 SUCCESSIVE OVER-RELAXATION METHOD

The SOR method is an iterative solver in which the solution to the FD equation at a particular node is based on values from both the previous iteration level and the current iteration level, and an additional parameter.

The SOR builds on the Gauss-Seidel method through the use of a relaxation factor, α (the additional parameter mentioned above). We define the difference between two Gauss-Seidel iterations as c , the residual. In equation form, using head (h) as the parameter of interest,

$$c = h_p^{k+1} - h_p^k \quad (10.3-1)$$

where k superscripts correspond to the iteration level.

In the SOR, the user-defined relaxation factor (1.85 by default) is used in determining the value to be used in the next iteration through

$$h_p^{k+1} = h_p^k + \omega c \quad (10.3-2)$$

Thus, the set of FD equations for the entire modeling domain are solved in an orderly, predetermined way that allows individual equation solutions to be applied in the same iteration level and using a relaxation factor to ‘inflate’ the value used in the next iteration level. These two premises are implemented to achieve faster convergence of the overall solution.

Using **Equation 2.3.1.4-1** as an example, the resulting iterative equation based on the SOR method is

$$\begin{aligned} h_p^{k+1} = h_p^k + \frac{\alpha}{(a_p + a_p^t + a_p^Q)} [& a_E h_E^k + a_W h_W^{k+1} \\ & + a_N h_N^k + a_S h_S^{k+1} + a_{NE} h_{NE}^k + a_{NW} h_{NW}^k \\ & + a_{SE} h_{SE}^{k+1} + a_{SW} h_{SW}^{k+1} + S_f^Q + S_f^t - (a_p + a_p^t + a_p^Q) h_p^k] \end{aligned} \quad (10.3-3)$$

The software default, or user-specified, initial values are used in the first iteration. The iterations proceed using the values (augmented by the relaxation factor) obtained from the previous and present iteration (as shown in **Equation 10.3-1**) until the solution converges at each node. IGW 3 considers the solution converged when the value between an iteration and the subsequent iteration are different by less than some error tolerance. The solver can also be instructed to stop when a certain number of iterations are performed, regardless of the error criteria. The convergence criteria in IGW 3 are set in the ‘Solver’ window (refer to **Chapter 13** in the *IGW 3 User’s Manual*).

The matrix assembly and solution are carried out by the source code subroutines SORHEADT and SORCBAR for head and concentration calculations, respectively. For concentration calculations, **Equation 10.3-3** becomes

$$\begin{aligned}
C_P^{k+1} = C_P^k + \frac{\alpha}{a_P} & (a_E C_E^k + a_W C_W^{k+1} + a_N C_N^k + a_S C_S^{k+1} \\
& + a_{NE} C_{NE}^k + a_{NW} C_{NW}^k + a_{SE} C_{SE}^{k+1} + a_{SW} C_{SW}^{k+1} \cdot \quad (10.3-4) \\
& + S_f^Q + S_f^t - a_P C_P^k)
\end{aligned}$$

In terms of the source code, the subroutines involve a coefficient matrix and a vector. The main diagonal elements are stored in S00(I,J)+CST2(I,J)%SP [or S00(I,J)+CST1(I,J)%SP] and the other elements in those listed in **Table 2.3.1-1** [or **Table 3.2.1.2-1**]. The vector is stored in the SUM23(I,J) variable.

CHAPTER 11: MODEL TESTING AND VERIFICATION

This chapter contains a number of IGW 3 verification exercises and results. The first section documents those related to the flow solver, the second section documents those related to the transport solver.

11.1 FLOW SOLVER VERIFICATION EXERCISES

The exercises presented in the following subsections test various aspects of the IGW 3 flow solver.

11.1.1 PUMPING IN AN INFINITE CONFINED AQUIFER - THEIS

Theis^{iv} presented an exact analytical solution for the transient drawdown in an infinite uniform confined aquifer. See **Figure 11.1.1-1**.

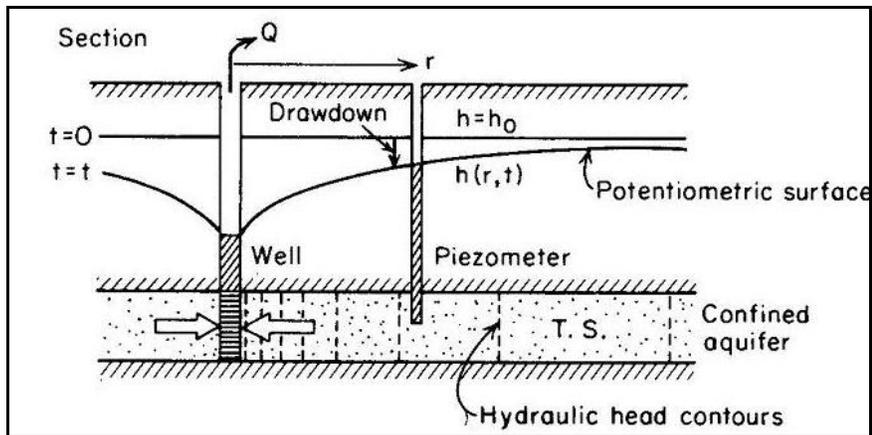


FIGURE 11.1.1-1'
Radial flow to a well in a horizontal confined aquifer

ANALYTICAL SOLUTION

The analytical solution of the drawdown as a function of time and distance could be expressed by

$$\frac{\partial^2 h}{\partial r^2} + \frac{1}{r} \frac{\partial h}{\partial r} = \frac{S}{T} \frac{\partial h}{\partial t}$$

Initial condition: $h(r,0) = h_0$ for $t \ll 0$

Boundary conditions:

$$\left. \begin{array}{l} h \rightarrow h_0 \text{ as } r \rightarrow \infty \\ \lim_{r \rightarrow 0} \left(r \frac{\partial h}{\partial r} \right) = \frac{Q}{2\pi T} \end{array} \right\} \text{for } t > 0 \text{ (Davis \& De Wiest, 1991)}$$

$$s(r,t) = h_0 - h(r,t) = -\frac{Q}{4\pi T} W(u)$$

$$u = \frac{Sr^2}{4Tt} \text{ and } W(u) = -Ei(-u) = \int_u^\infty \frac{e^{-x}}{x} dx$$

$$W(u) \approx -0.5772 - \ln u + \sum_{i=1}^n \frac{(-1)^{i+1} u^i}{i!i} \text{ (series approximation)}$$

where $s(r,t)$: predicted drawdown at a given radial distance for a time interval, [L]

h_0 : initial head in the aquifer, [L]

$h(r,t)$: predicted head at a given radial distance for a time interval, [L]

r : radial distance from the pumping well, [L]

Q : constant pumping discharge, [L³/T]

$W(u)$: well function (exponential integral)

S : storage coefficient, [-]

T : transmissivity, [L²/T]

t : time, [T]

Another approximation to the well function is given by Swamee and Ojha (1990) as;

$$A = \ln \left[\left(\frac{0.56146}{u} + 0.65 \right) (1+u) \right]$$

$$B = u^4 e^{7.7u} (2+u)^{3.7}$$

$$W(u) = (A^{-7.7} + B)^{-0.13}$$

(11.1.1-1)

and where

$$u = \frac{(x^2 + y^2)S}{4Tt} \tag{11.1.1-2}$$

and the well function, $W(u)$, is defined by

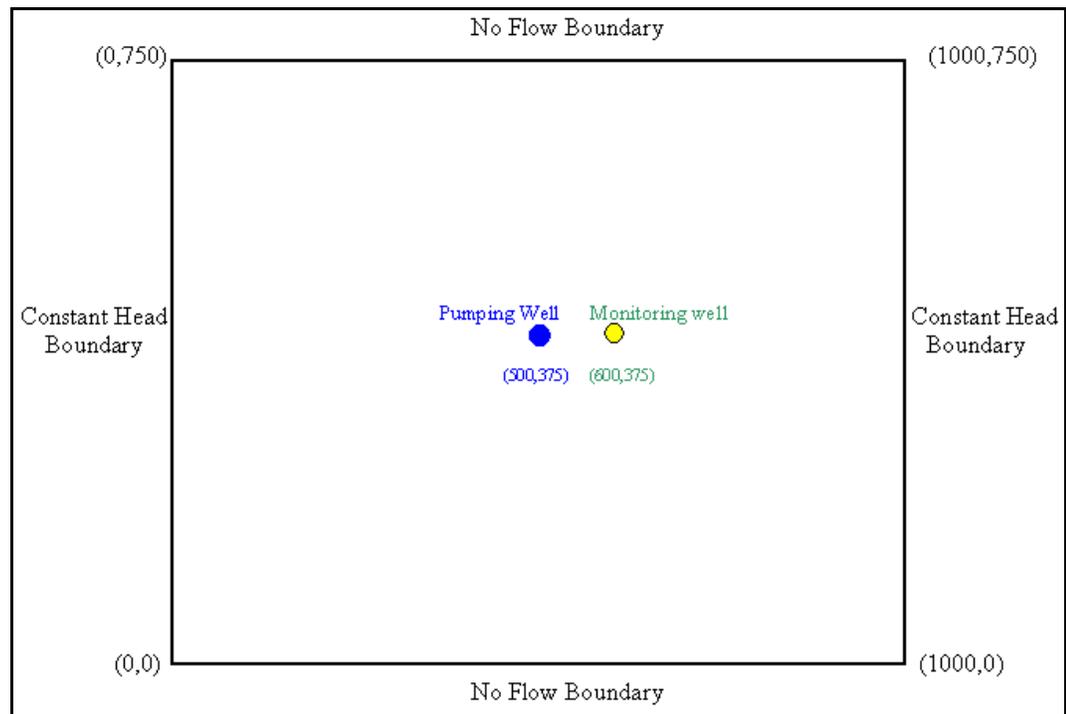
IGW NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.1.1-1**. A plan view of the IGW model set up is presented in **Figure 11.1.1-2**.

TABLE 11.1.1-1

Assumptions related to IGW numerical solution

Physical Parameters					Numerical Parameters		
Q	h_0	S	t	T	Δx	Δy	Δt
1000 m ³ /day	25 m	0.0002	0.12 day	1000 m ² /day	34.48 m	34.48 m	0.004 s

**FIGURE 11.1.1-2**

Plan view of IGW model set up

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Figures 11.1.1-3 and **11.1.1-4** show comparisons between the Theis solution and the IGW solution.

Figure 11.1.1-5 shows a 3-D comparison of the depression cones from the two solutions.

The numerical solution is graphically indistinguishable from the exact solution until the drawdown influence begins to reach the boundaries.

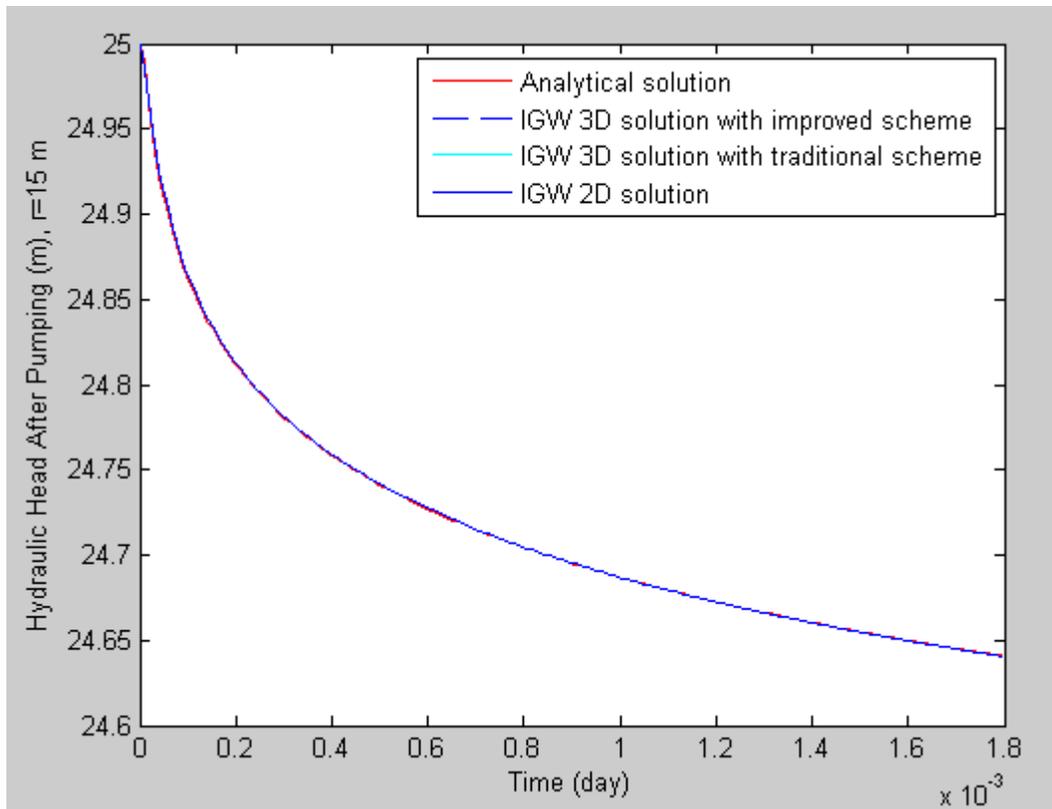


FIGURE 11.1.1-3
Comparison of the exact solution and the IGW predicted solution at a location 15 m from the well

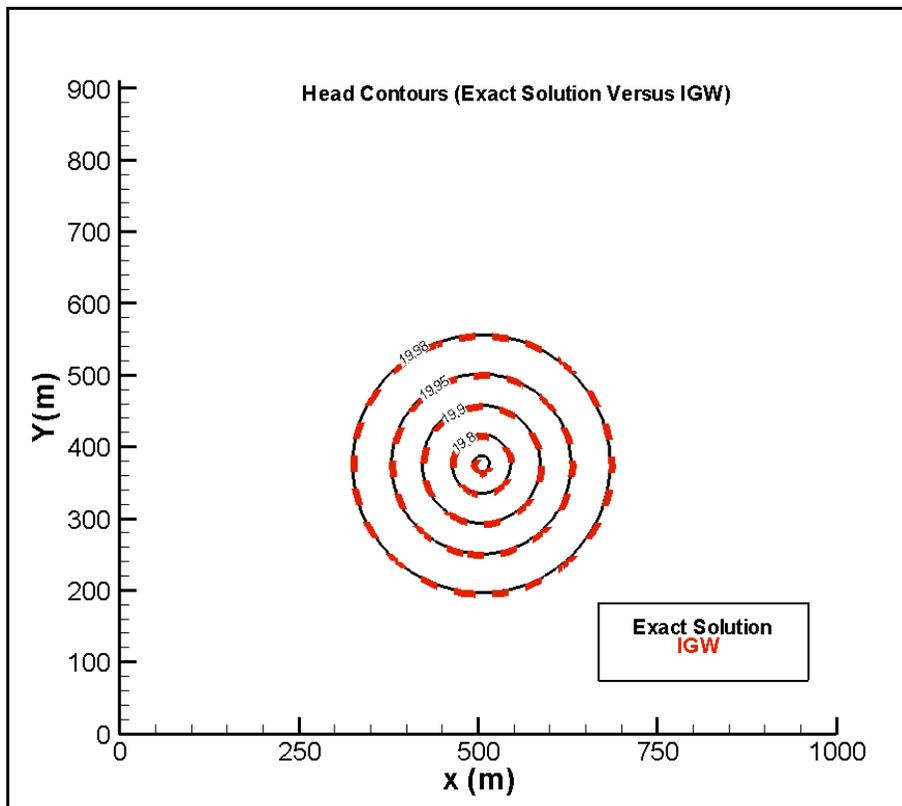


FIGURE 11.1.1-4

Comparison of the exact solution and the IGW predicted solution at 151.37 seconds

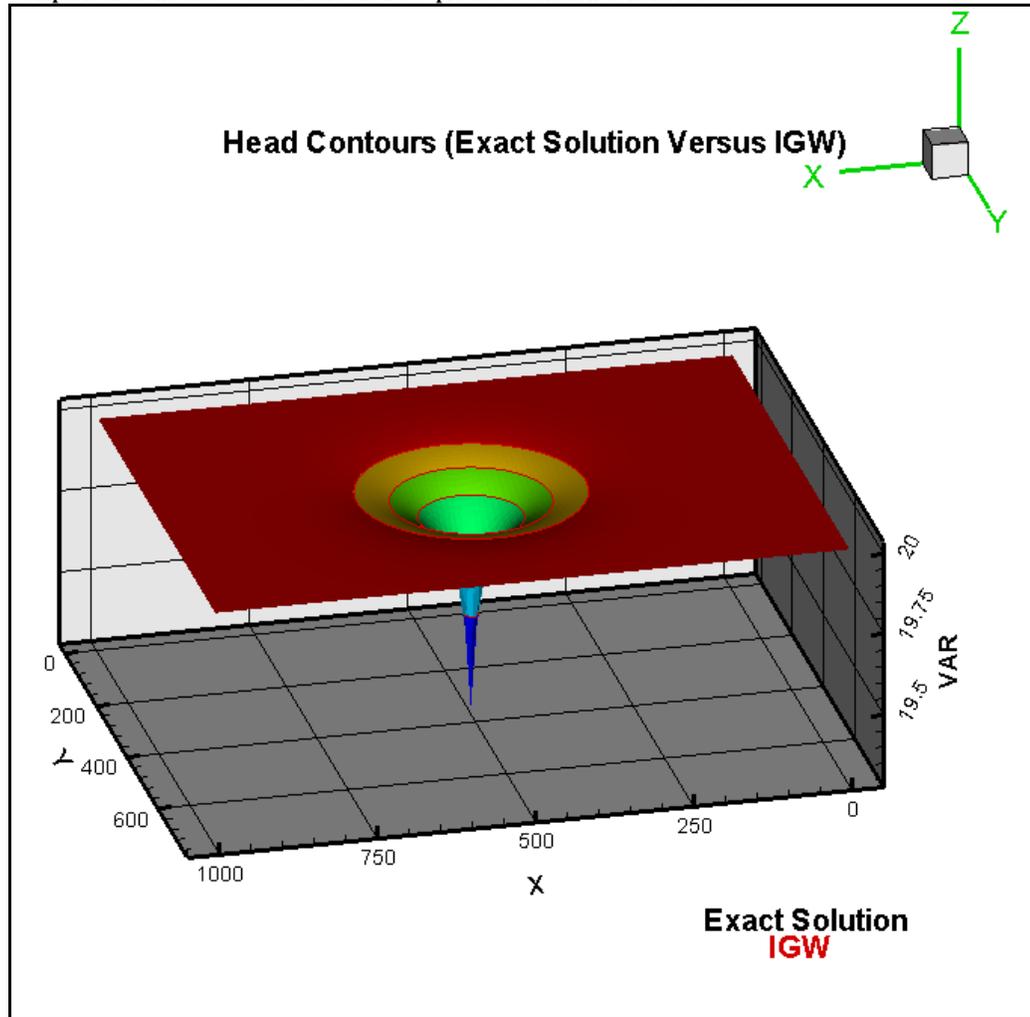


FIGURE 11.1.1-5

Depression cone comparison

11.1.2 PUMPING IN AN INFINITE CONFINED ANISOTROPIC AQUIFER

The exact solution to flow in an infinite, anisotropic, and confined aquifer under constant pumping is:

$$h_0 - h(x, y, t) = \frac{Q\sqrt{p}}{4\pi T} W(u) \quad (11.1.2-1)^{vi}$$

where x, y = the rectilinear coordinates relative to the pumping well [L],

$p = K_x/K_y$ = the ratio of anisotropy [-],

S = the storage coefficient [-],

T = the transmissivity [L^2/T],

t = the time [T],

h_0 = the initial head [L],

Q = the constant flow rate abstracted from the well [L^3/T],

and where

$$u = \frac{(x^2 + py^2)S}{4Tt} \quad (11.1.2-2)$$

and the well function $W(u)$ is defined in **Equation 11.1.1-3**.

IGW NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.1.2-1**. Refer to **Figure 11.1.1-2** for a plan view of the IGW model set up. Note the K_x is oriented left to right and K_y is oriented top to bottom (with respect to the plan layout on the page).

TABLE 11.1.2-1

Assumptions related to IGW numerical solution

Physical Parameters						Numerical Parameters		
Q	h_0	S	t	T	p	Δx	Δy	Δt
1000 m ³ /day	20 m	0.0002	151.37 s	1000 m ² /day	10	10 m	10 m	1.036 s

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Figures 11.1.2-2 and **11.1.2-3** show comparisons between the Theis solution and the IGW solution.

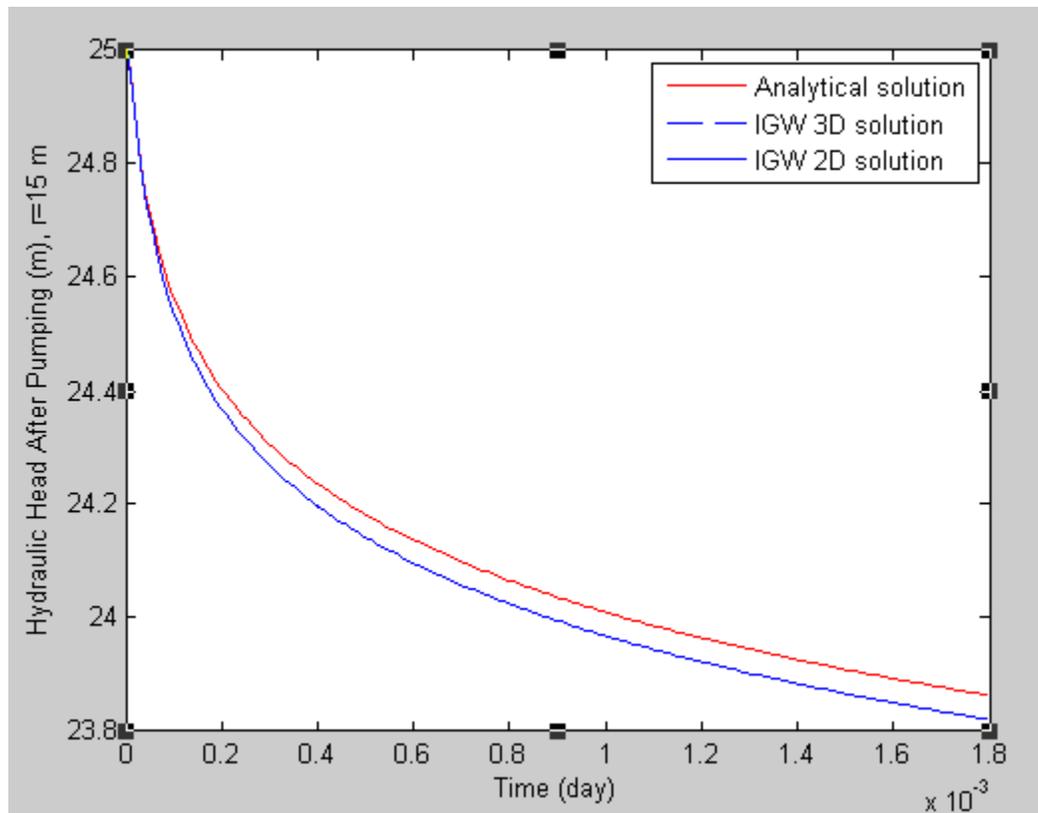


FIGURE 11.1.2-2

Comparison of the analytical solution and the IGW solution at a location 100 meters from the well

Figure 11.1.2-4 shows a 3-D comparison of the depression cones from the two solutions.

The numerical solution is graphically indistinguishable from the exact solution until the drawdown influence begins to reach the boundaries.

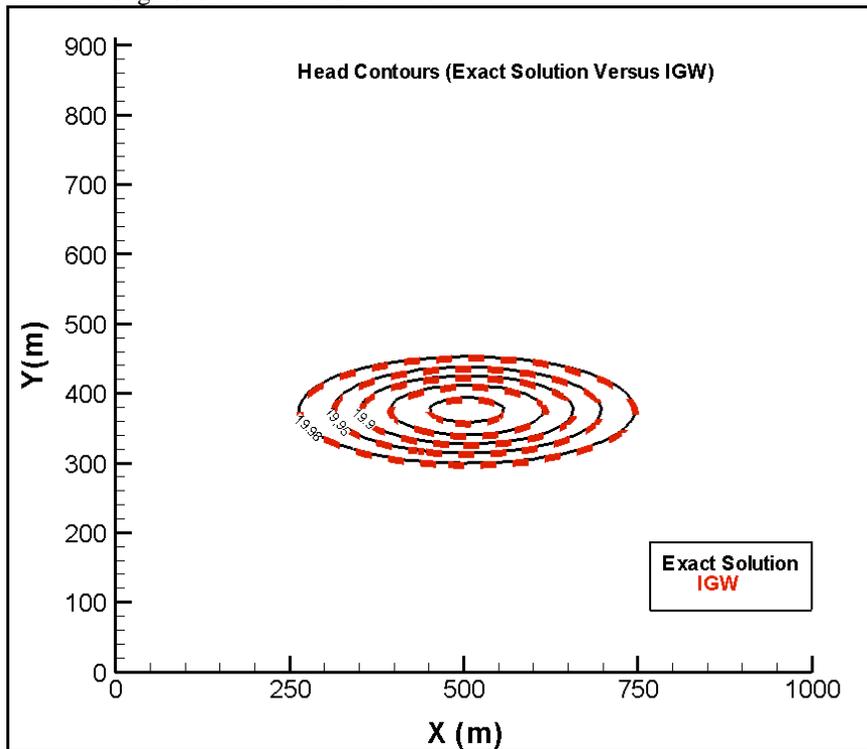


FIGURE 11.1.2-3
Comparison of the analytical solution with the IGW solution at 151.37 seconds

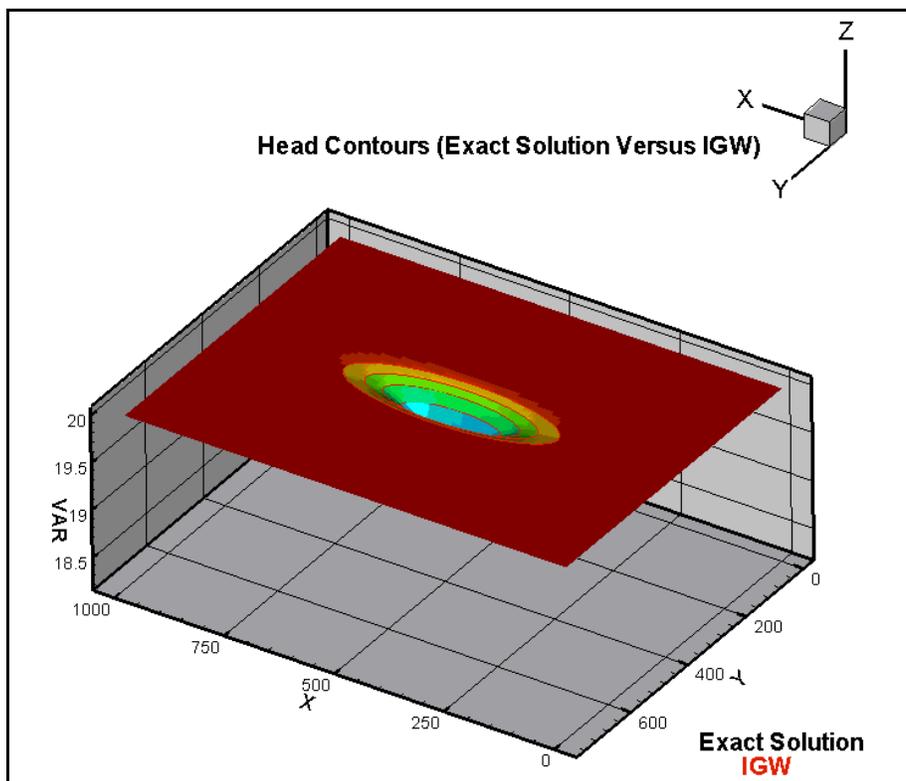


FIGURE 11.1.2-4
Depression cone comparison

11.1.3 PUMPING NEAR AN IMPERVIOUS BODY

When a confined aquifer is bounded on one side by a straight-line impermeable boundary, drawdown due to pumping will be greater near the boundary (see **Figure 11.1.3-1**).

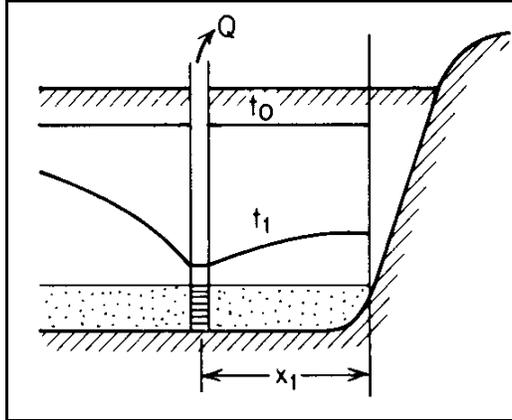


FIGURE 11.1.3-1
Drawdown near an impervious boundaryⁱⁱ

ANALYTICAL SOLUTION

The analytical solution for drawdown near an impervious boundary as a function of time and space is given by

$$h_0 - h(x, y, t) = \frac{Q}{4\pi T} [W(u_r) + W(u_i)]$$

where;

$$W(u) = (A^{-7.7} + B)^{-0.13} \quad (11.1.3-1)^{vii}$$

$$A = \ln \left[\left(\frac{0.56146}{u} + 0.65 \right) (1 + u) \right]$$

$$B = u^4 e^{7.7u} (2 + u)^{3.7}$$

where x, y = rectilinear coordinates relative to the pumping well [L],

a = the distance of pumping well from constant head boundary [L],

S = the aquifer storage coefficient [-],

T = the aquifer transmissivity [L^2/T],

t = the time [T],

h_0 = the initial head in the boundaries before pumping [L],

Q = the constant flow rate abstracted from the well [L^3/T],

and where

$$u_r = \frac{[(x - a)^2 + y^2]S}{4Tt}, \quad (11.1.3-2)$$

$$u_i = \frac{[(x + a)^2 + y^2]S}{4Tt} \quad (11.1.3-3)$$

and the well function $W(u)$ is defined in **Equation 11.1.1-3**.

IGW NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.1.3-1**. A plan view of the IGW model set up is given in **Figure 11.1.3-2**.

TABLE 11.1.3-1

Assumptions related to IGW numerical solution

Physical Parameters						Numerical Parameters		
Q	h_0	S	t	T	a	Δx	Δy	Δt
1000 m ³ /day	20 m	0.0002	151.37 s	1000 m ² /d	50 m	5.5 m	5.5 m	1.0368 s

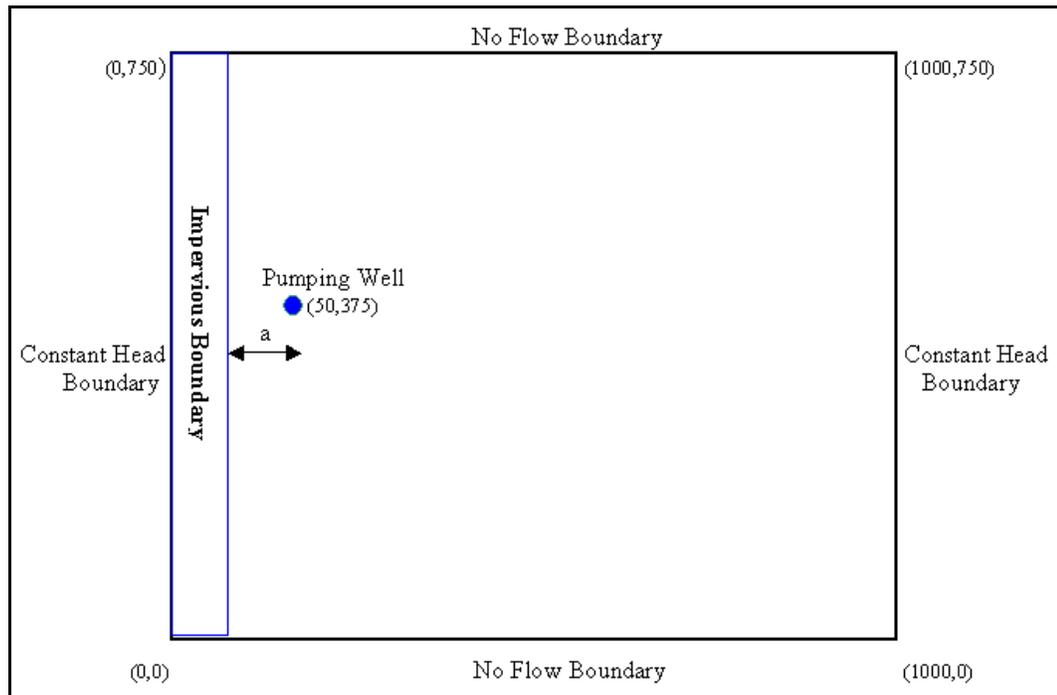


FIGURE 11.1.3-2
Plan view of IGW model setup

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Figures 11.1.3-3 and **11.1.3-4** show comparisons between the Theis solution and the IGW solution.

Figure 11.1.3-5 shows a 3-D comparison of the depression cones from the two solutions.

The numerical solution is graphically indistinguishable from the exact solution until the drawdown influence begins to reach the boundaries.

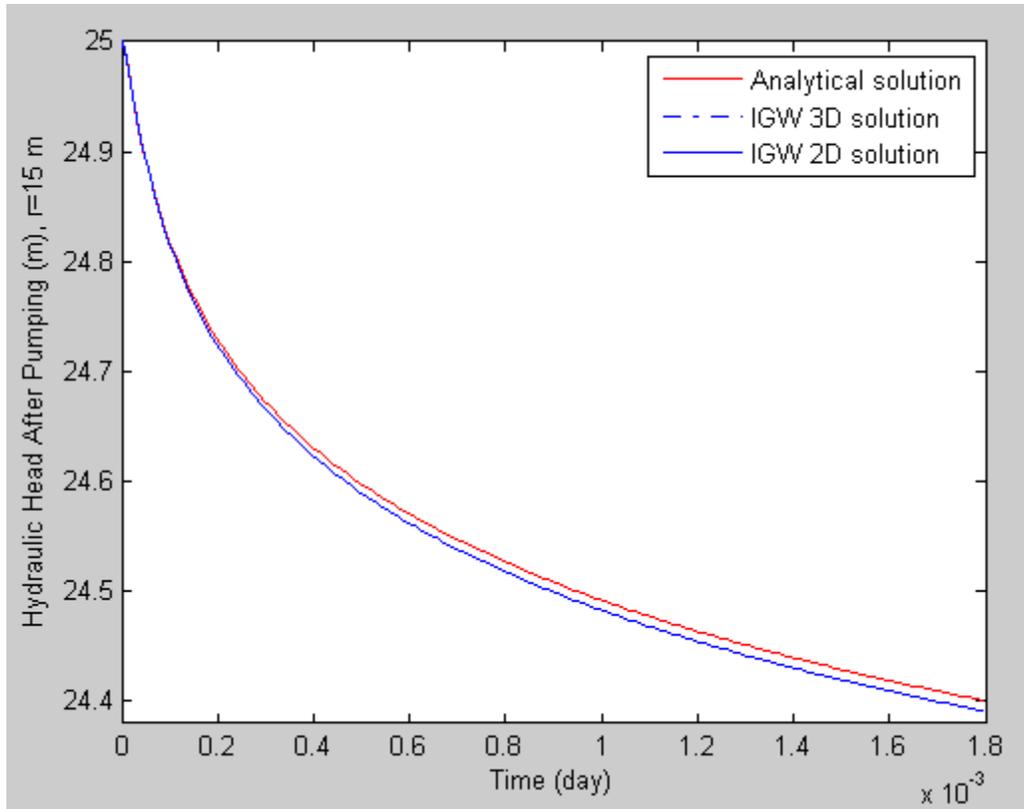


FIGURE 11.1.3-3
Transient comparison between the analytical solution and the IGW solution at 50 meters from the pumping well

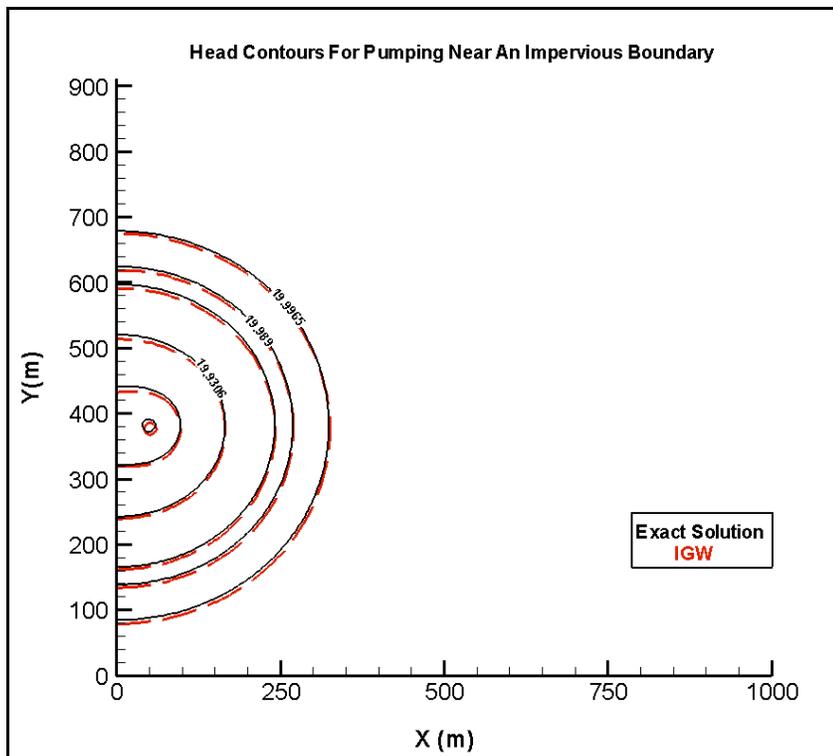


FIGURE 11.1.3-4

Comparison of the analytical solution with the IGW solution and 151.37 seconds

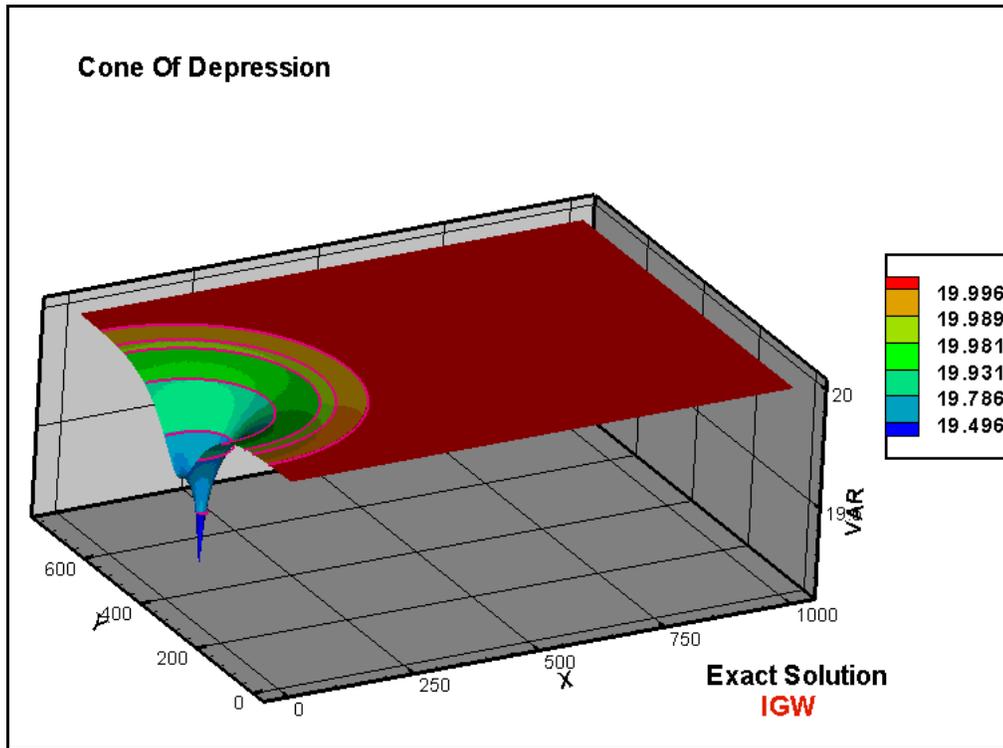


FIGURE 11.1.3-5
Depression cone comparison

11.1.4 PUMPING NEAR A CONSTANT HEAD BOUNDARY

When a confined aquifer is bounded on one side by a straight constant head boundary, drawdown due to pumping will be less near the boundary (see **Figure 11.1.4-1**).

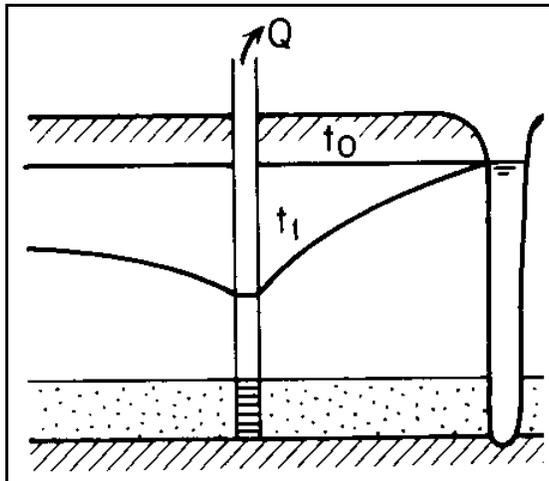


FIGURE 11.1.4-1
Drawdown near a constant head boundaryⁱⁱ

ANALYTICAL SOLUTION

The analytical solution for drawdown near a constant head boundary as a function of time and space is given by^{viii}

$$h_0 - h(x, y, t) = \frac{Q}{4\pi T} [W(u_r) - W(u_i)] \quad (11.1.4-1)$$

where x, y = rectilinear coordinates relative to the pumping well [L],

a = the distance of pumping well from constant head boundary [L],

S = the aquifer storage coefficient [-],

T = the aquifer transmissivity [L²/T],

t = the time [T],

h_0 = the initial head in the boundaries before pumping [L],

Q = the constant flow rate abstracted from the well [L³/T],

and where

$$u_r = \frac{[(x - a)^2 + y^2]S}{4Tt}, \quad (11.1.4-2)$$

$$u_i = \frac{[(x + a)^2 + y^2]S}{4Tt}, \quad (11.1.4-3)$$

and the well function, $W(u)$ is given in **Equation 11.1.1-3**. Note that the subscripts i and r stand for real and imaginary, respectively.

IGW NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.1.4-1**. Refer to **Figure 11.1.3-2** for a plan view of the IGW model set up (and note that the impervious boundary is replaced with a river that has a constant defined head).

TABLE 11.1.4-1

Assumptions related to IGW numerical solution

Physical Parameters						Numerical Parameters		
Q	h_0	S	t	T	a	Δx	Δy	Δt
1000 m ³ /day	20 m	0.0002	151.37 s	1000 m ² /d	50 m	10 m	10 m	1.036 s

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Figures 11.1.4-2 and **11.1.4-3** show comparisons between the analytical solution and the IGW solution.

Figure 11.1.4-4 shows a 3-D comparison of the depression cones from the two solutions.

The numerical solution is graphically indistinguishable from the exact solution until the drawdown influence begins to reach the boundaries.

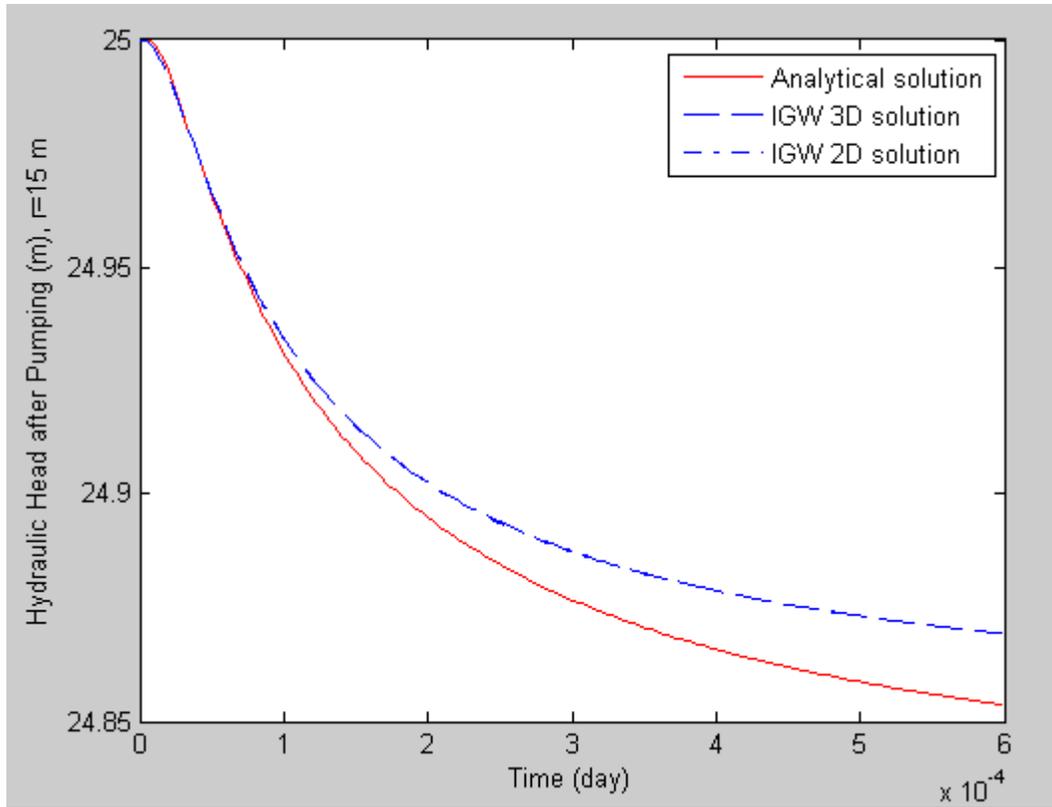


FIGURE 11.1.4-2
Transient comparison between the analytical solution and the IGW solution at 50 meters from the well

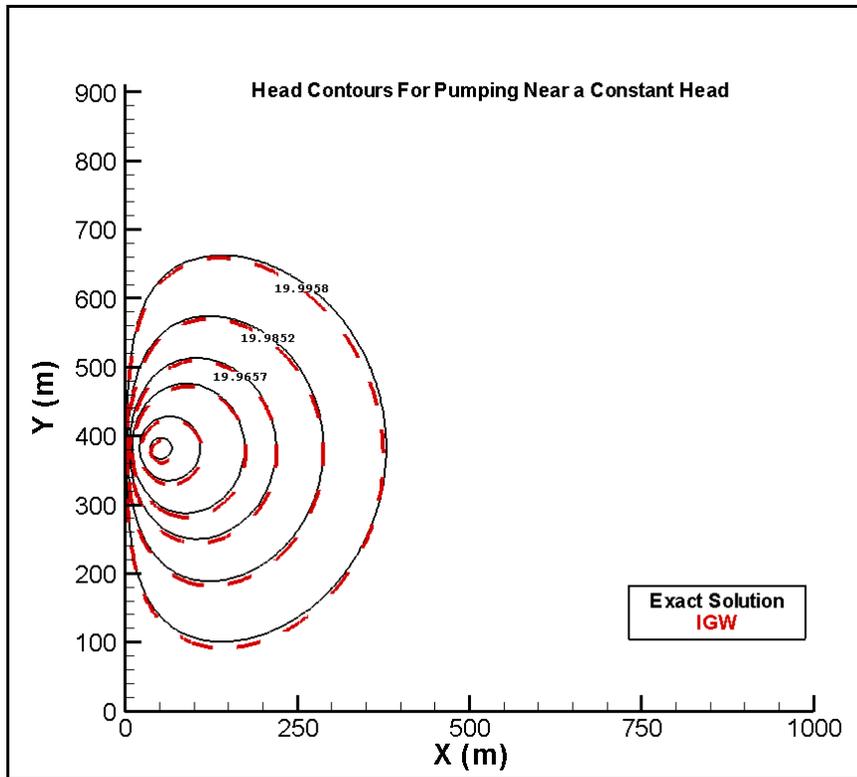
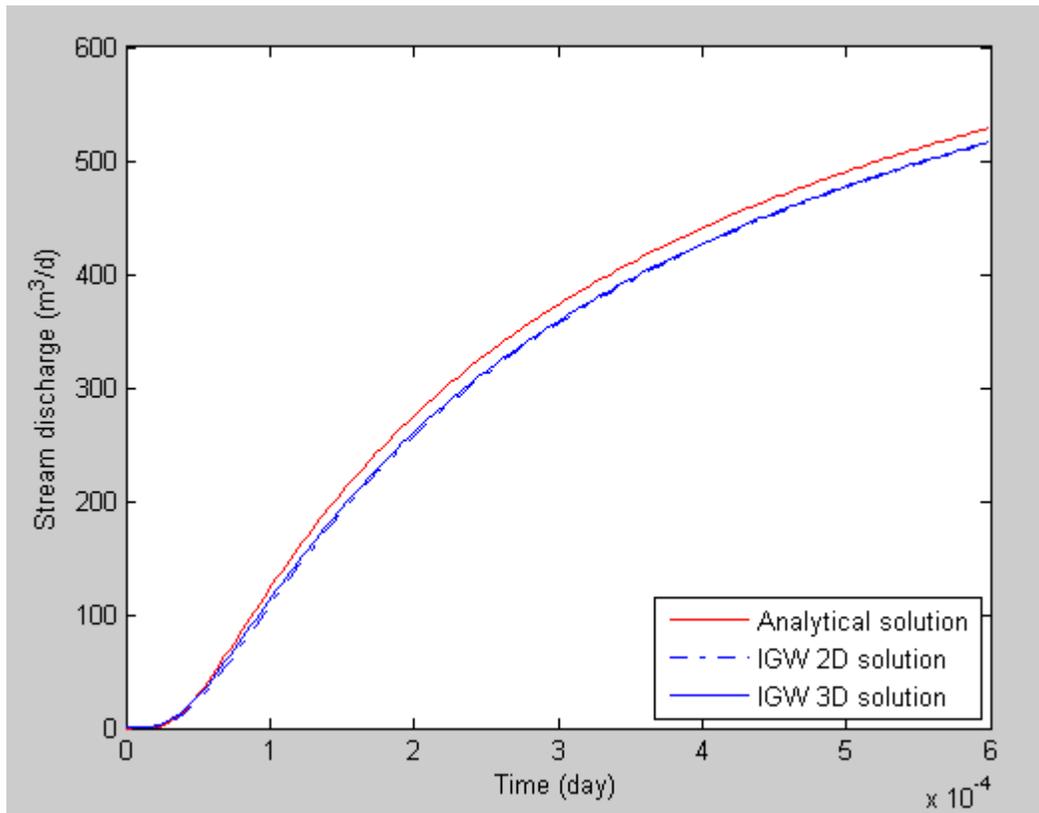


FIGURE 11.1.4-3

Comparison of the analytical solution with the IGW solution at 151.37 seconds



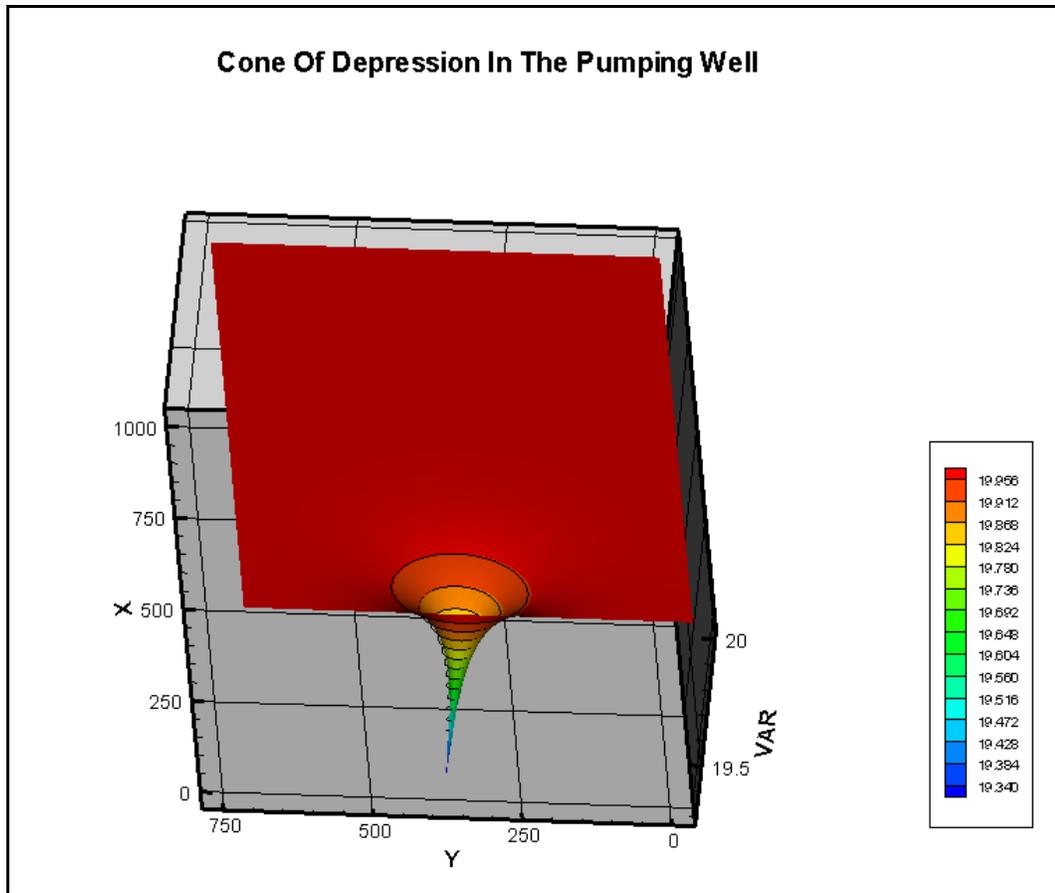


FIGURE 11.1.4-4
Depression cone comparison

11.1.5 TRANSIENT AQUIFER DYNAMICS – TIDAL RIVER RESPONSE

Waves in surface water bodies will propagate into aquifers that are in direct communication with them (see **Figure 11.1.5-1**).

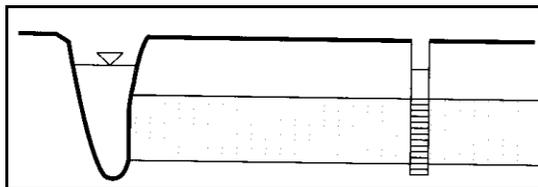


FIGURE 11.1.5-1
Cross-section of an aquifer in direct communication with a surface water body

ANALYTICAL SOLUTION

The problem has been investigated by Ferris^{ix}, who considered sinusoidal stage changes in the surface water body. If s is the stage change in the aquifer (departure from the equilibrium piezometric head value), then

$$s(x, t) = s_r \exp\left(-\sqrt{\frac{\omega S x^2}{2T}}\right) \sin\left(\omega t - \sqrt{\frac{\omega S x^2}{2T}}\right) \quad (11.1.5-1)$$

where s_r = the amplitude or half-range of the stage change in the surface water body [L],

$\omega = 2\pi / \tau$ = the frequency [T^{-1}],

S = the aquifer storage coefficient [-],

T = the aquifer transmissivity [L^2/T],

t = the time [T], and

x = the distance from the surface water body [L].

IGW NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.1.5-1**. A plan view of the IGW model set up is given in **Figure 11.1.5-2**.

TABLE 11.1.5-1

Assumptions related to IGW numerical solution

Physical Parameters							Numerical Parameters		
x	h_0	S	t	T	s_r	ω	Δx	Δy	Δt
82.61 and 203.452 m	0 m	0.0002	1120 d	0.1 m ² /d	2 m	1 d ⁻¹	8.5 m	8.5 m	2 d



FIGURE 11.1.5-2

Plan view of the IGW model setup

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Figures 11.1.5-3 and **11.1.5-4** show comparisons between the Theis solution and the IGW solution.

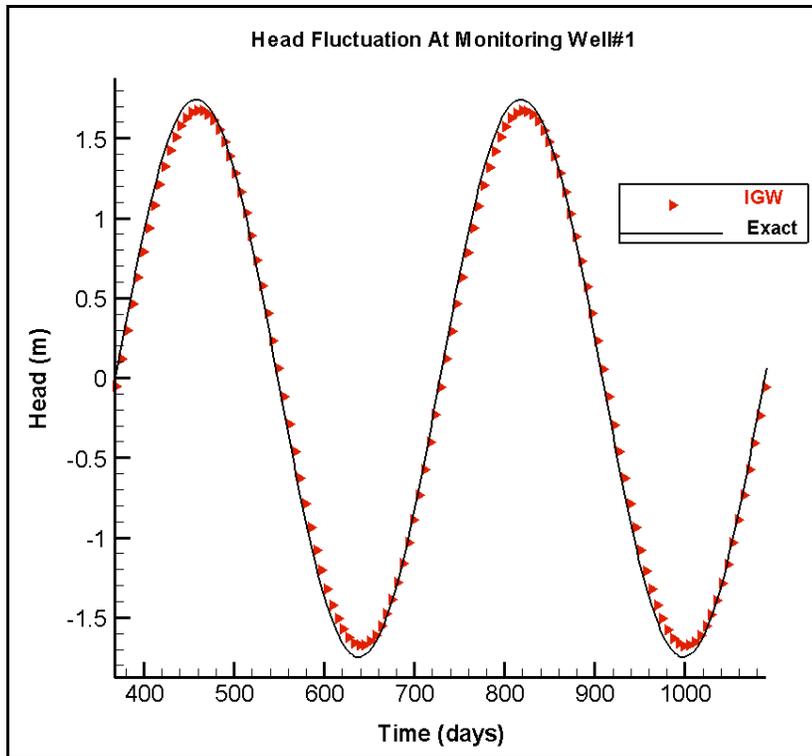


FIGURE 11.1.5-3
 Transient aquifer dynamics in response to tidal surface bodies at monitoring well #1

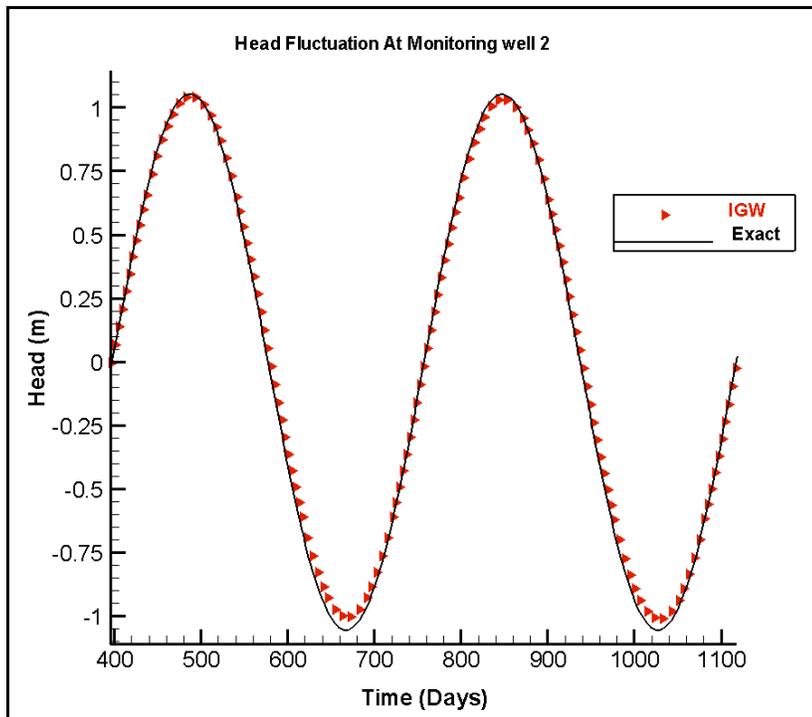
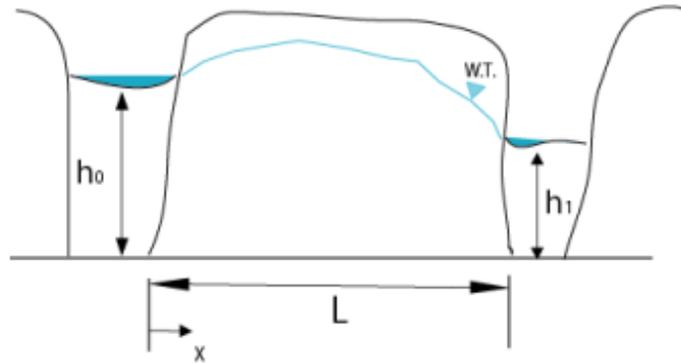


FIGURE 11.1.5-4
 Transient aquifer dynamics in response to tidal surface bodies at monitoring well #2

The numerical solution is graphically consistent with the exact solution.

11.1.6 1D STEADY-STATE FLOW IN UNCONFINED AQUIFER (NO RECHARGE)

Unconfined aquifer with constant head boundaries on left and right sides are given below.



Unconfined aquifer with two constant head boundaries

Analytical solution

For steady state flow conditions;

$$\frac{dq}{dx} = 0 \Rightarrow \frac{d}{dx} \left(kh \frac{dh}{dx} \right) = 0, \frac{d^2(h^2)}{dx^2} = 0$$

$$x = 0: h = h_0$$

$$x = L: h = h_1$$

$$h(x) = \sqrt{h_0^2 - \frac{(h_0^2 - h_1^2)x}{L}}$$

where $h(x)$: predicted head at a given distance, [L]

h_0 : constant head at $x = 0$, [L]

h_1 : constant head at $x = L$, [L]

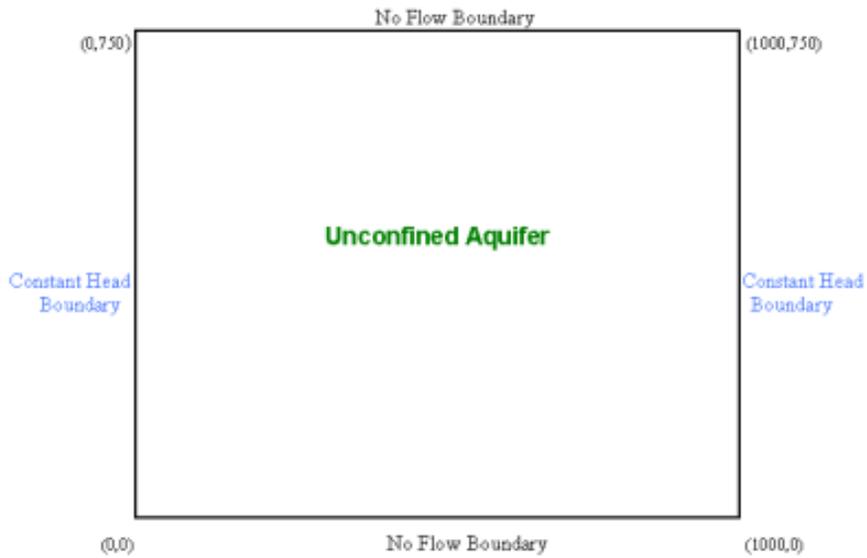
x : distance from the left boundary, [L]

L : length of the domain, [L]

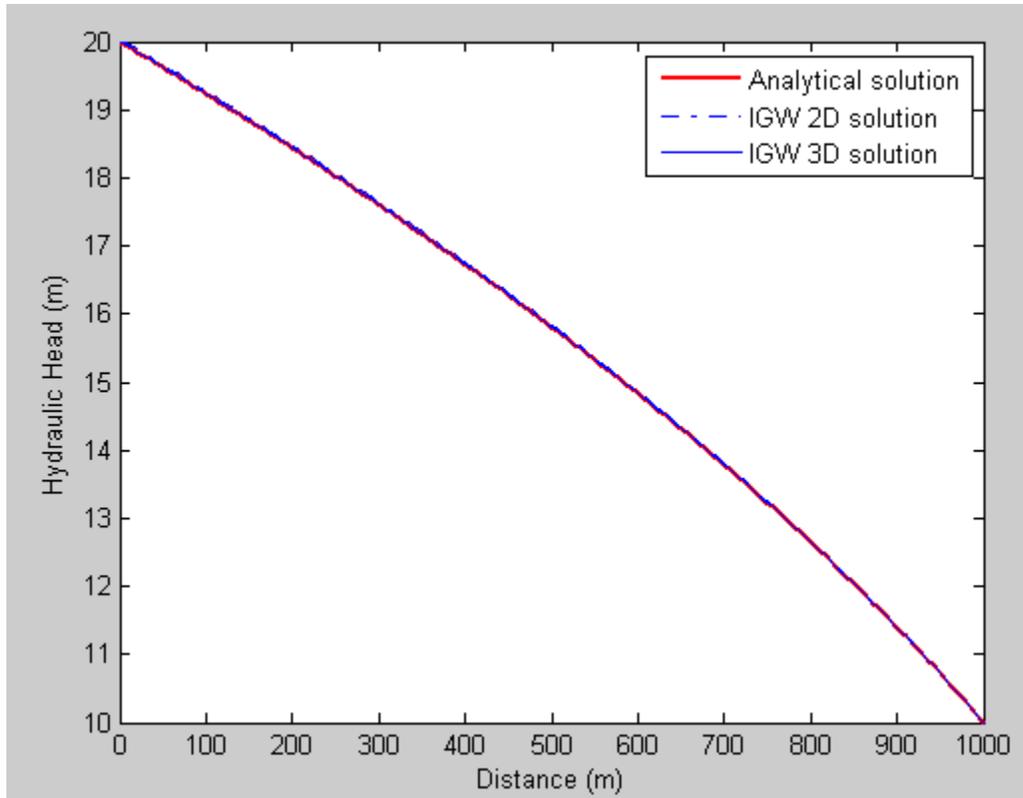
Physical Parameters				
Aquifer Top/Bottom	h_0	h_1	K	L
30 m and 0 m	20 m	10 m	50 m/d	1000 m

Numerical setup

Using given physical and numerical parameters, IGW is applied to solve the one-dimensional groundwater flow problem.



IGW Numerical Parameters			
Δx	<i>Matrix solver</i>	<i>Maximum outer iterations</i>	<i>Relative tolerance</i>
4.92 m	Algebraic multi grid	10	0.0001 %



Comparison of analytical solution and IGW solution (no recharge)

11.1.6 1D STEADY-STATE FLOW IN UNCONFINED AQUIFER (WITH RECHARGE)

In this case an unconfined aquifer is adjacent to two constant head boundaries and constant recharge is applied to top of the aquifer (see **Figure 11.1.6-1**).

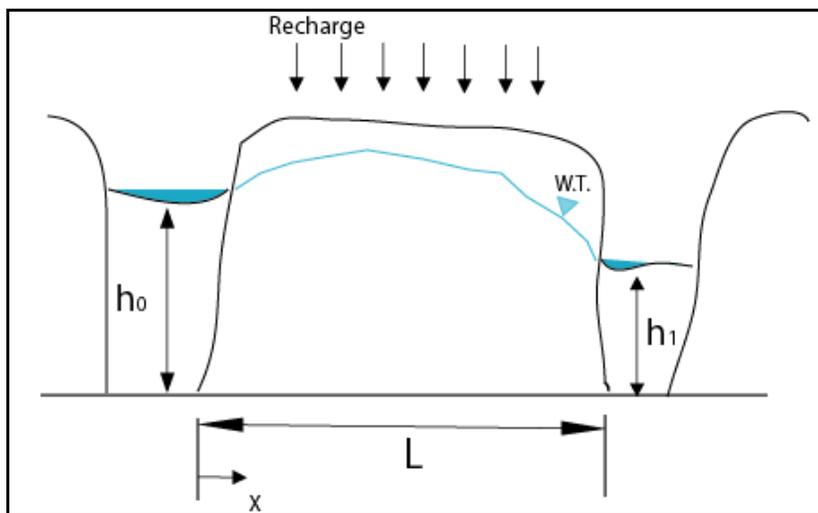


FIGURE 11.1.6-1
Unconfined aquifer with two constant head boundaries

ANALYTICAL SOLUTION

The analytical solution for steady state flow for head prediction along the aquifer's length (1D) is given by (where it is assumed that aquifer is homogenous)

For steady state flow conditions;

$$\frac{dq}{dx} + N = 0 \Rightarrow k \frac{d}{dx} \left(h \frac{dh}{dx} \right) + N = 0$$

$$x = 0 : h = h_0$$

$$x = L : h = h_1$$

$$h(x) = \sqrt{h_0^2 - \frac{(h_0^2 - h_1^2)x}{L} + \frac{N}{k}(L-x)x}$$

(11.1.6-1)

where $h(x)$: predicted head at a given distance, [L]

h_0 : constant head at $x = 0$, [L]

h_1 : constant head at $x = L$, [L]

x : distance from the left boundary, [L]

L : length of the domain, [L]

N : recharge, [L/T]

k : hydraulic conductivity, [L/T]

IGW NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.1.6-1**. A plan view of the IGW model set up is given in **Figure 11.1.6-2**.

TABLE 11.1.6-1

Assumptions related to IGW numerical solution

Physical Parameters					
Aquifer Top / Bottom	h_0	h_1	K	r	L
30 m and 0 m	20 m	10 m	50 m/d	0.1 m/d	1000 m

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Figure 11.1.6-3 shows a comparison between the analytical solution and the IGW solution.

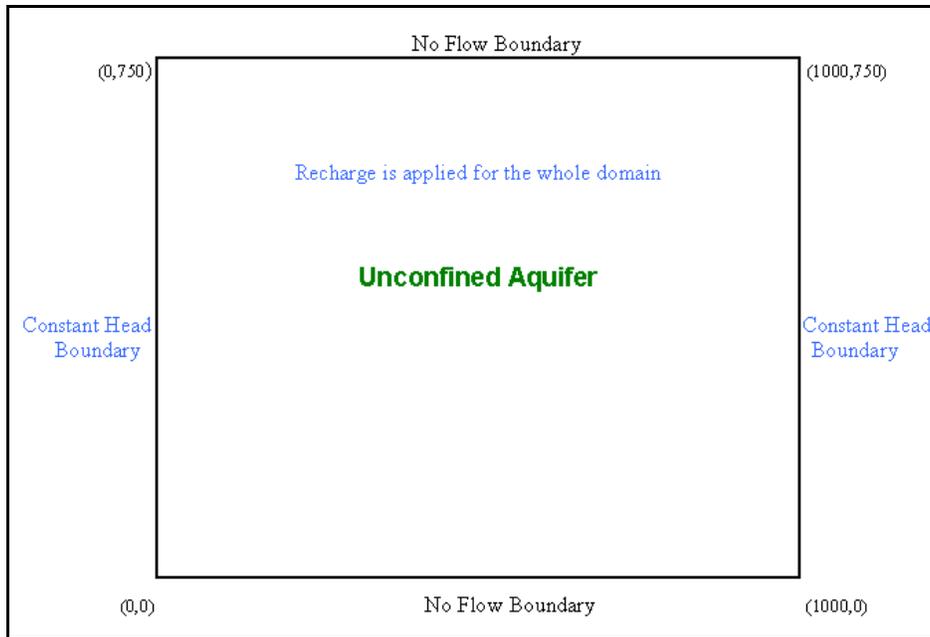


FIGURE 11.1.6-2
Plan view of IGW model setup

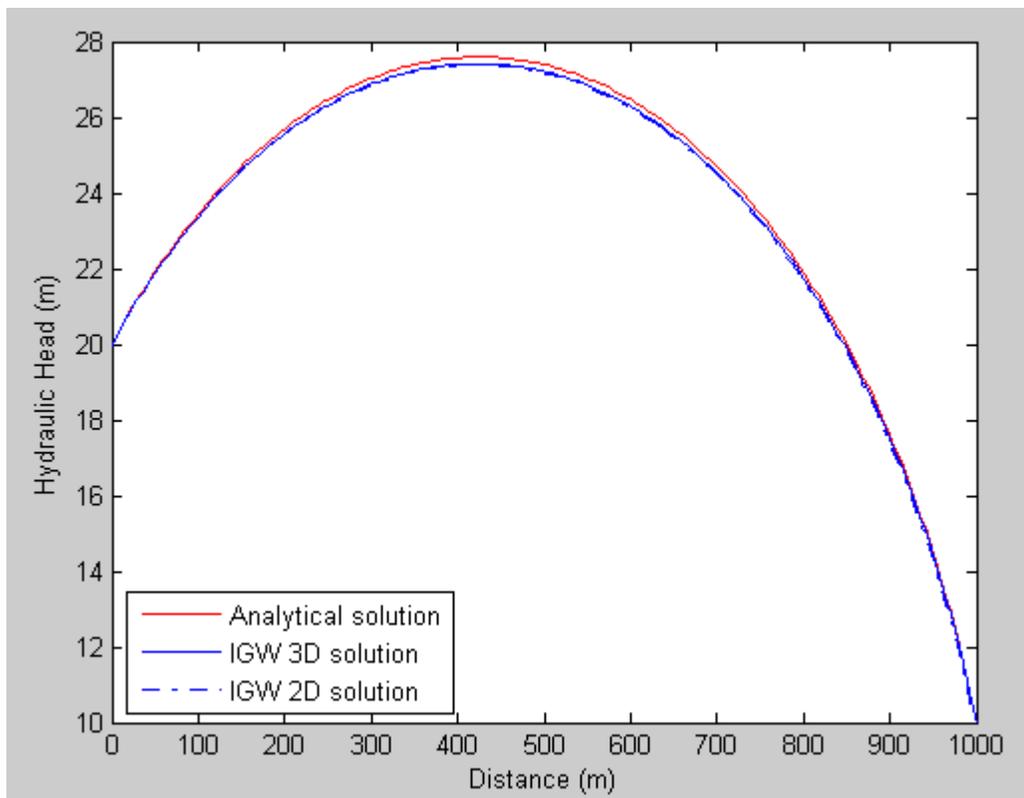


FIGURE 11.1.6-3
Comparison of the analytical solution and the IGW solution

IGW Numerical Parameters			
Δx	Matrix solver	Maximum outer iterations	Relative tolerance
4.92 m	Algebraic multi grid	10	0.0001 %

The numerical solution is graphically indistinguishable from the exact solution.

11.1.7 ISLAND AQUIFER

In the island aquifer problem, IGW was compared to the analytical solution of the steady state head distribution due to groundwater pumping in a homogeneous aquifer. Hydraulic head is calculated for a cylindrical island with a constant abstraction rate in a well in the center of the island. Constant recharge is added on top of the aquifer (see **Figure 11.1.7-1**).

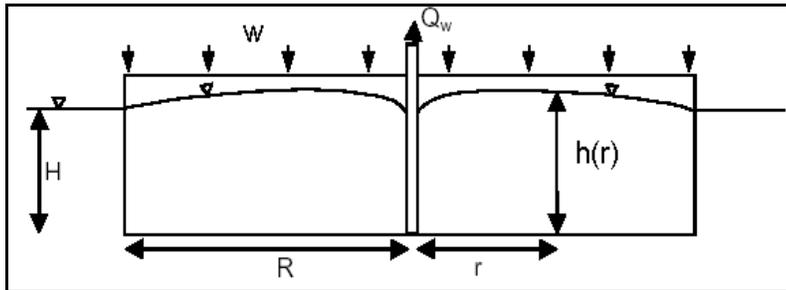


FIGURE 11.1.7-1
The island aquifer

ANALYTICAL SOLUTION

The steady-state solution of the hydraulic head (before pumping), h , as a function of the radial distance from the well, r , is given by

For steady state flow conditions;

For steady state flow conditions;

$$\frac{d^2(h^2)}{dr^2} + \frac{1}{r} \frac{d(h^2)}{dr} + N = 0$$

$$r = R: h = H, h_w \leq h \leq H$$

$$Q_w = 2\pi rhq_r = 2\pi rhk \frac{dh}{dr} = 2\pi rk \frac{d(h^2)}{dr}, h_w \leq h \leq H, r_w \leq r \leq R$$

$$h(r) = \sqrt{H^2 - \frac{Q_w}{\pi k} \ln \frac{R}{r} + \frac{N}{2k} (R^2 - r^2)}$$

where $h(r)$: predicted head at a given radial distance, [L]

H : initial water level in the aquifer, [L]

Q_w : constant pumping discharge, [L^3/T]

k : hydraulic conductivity, [L/T]

r : radial distance, [L]

R : distance from the left boundary, [L]

L : length of the domain, [L]

N : recharge, [L/T]

(11.1.7-1)

NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.1.7-1**. A plan view of the IGW model set up is given in **Figure 11.1.7-2** (note that the pumping well is located at (507.5,438.5)).

TABLE 11.1.7-1

Assumptions related to IGW numerical solution

Physical Parameters							Numerical Parameters	
w	Aquifer Top / Bottom	Q_w	H	S_y	K	R	$\Delta x / \# \text{ cells}$	$\Delta y / \# \text{ cells}$
0.1 m/day	50 m and 0 m	3000 m ³ /day	20 m	0.1	50 m/day	323 m	5.5 m / 200	5.5 m / 150

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Comparison between the analytical solution and IGW is presented for two cases. The first case (**Figure 11.1.7-3**) occurs before pumping when only recharge is added to the island aquifer and the second (**Figure 11.1.7-4**) occurs after pumping. Both cases are in steady state condition.

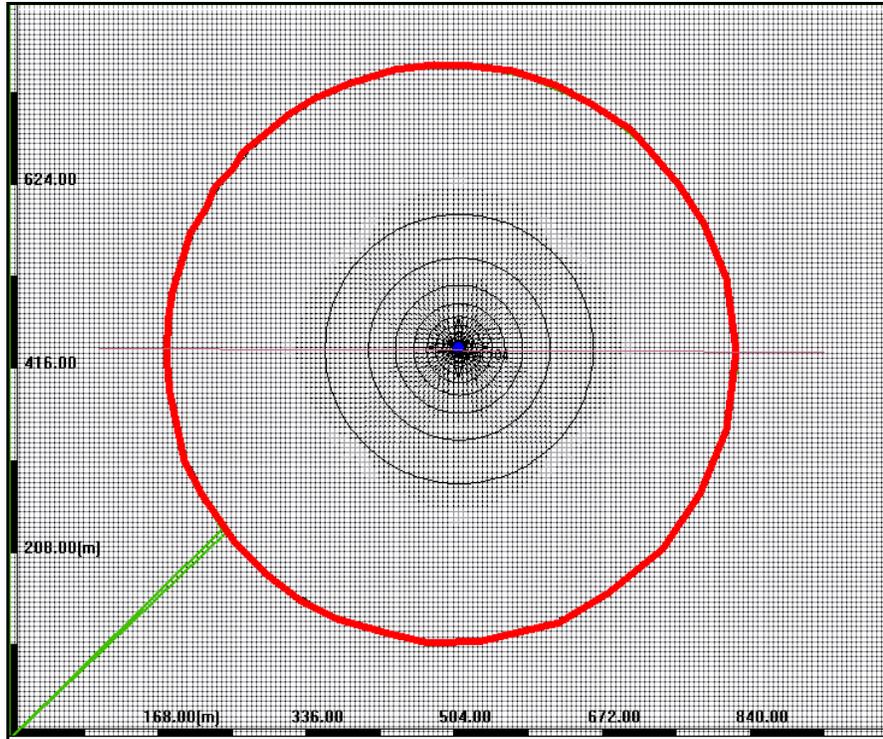


FIGURE 11.1.7-2
IGW model layout for island aquifer

For steady state flow conditions;

$$\frac{dq}{dx} + N = 0 \Rightarrow k \frac{d}{dx} \left(h \frac{dh}{dx} \right) + N = 0$$

$$x = 0: h_0 = H$$

$$x = L: h_1 = H$$

$$h(x) = \sqrt{h_0^2 - \frac{(h_0^2 - h_1^2)x}{L} + \frac{N}{k}(L-x)x}$$

where $h(x)$: predicted head at a given distance, [L]

h_0 : constant head at $x = 0$, [L]

h_1 : constant head at $x = L$, [L]

x : distance from the left boundary, [L]

L : length of the domain, [L]

N : recharge, [L/T]

k : hydraulic conductivity, [L/T]

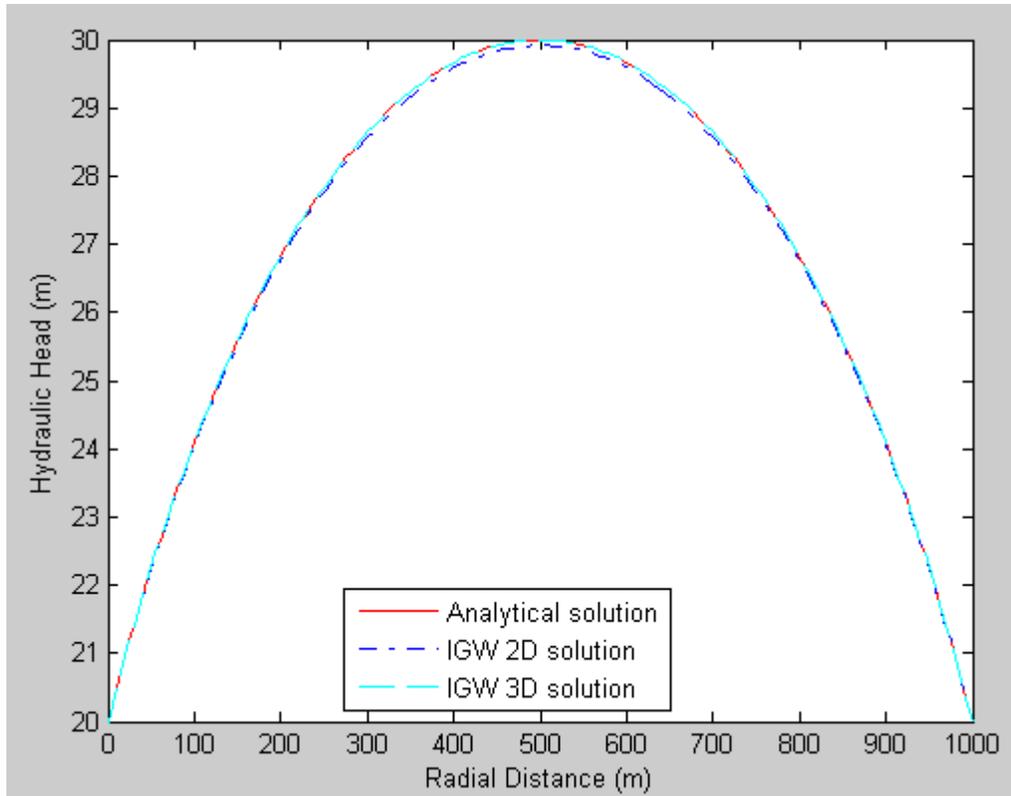


FIGURE 11.1.7-3
The island aquifer analytical solution compared to IGW – before pumping

IGW Numerical Parameters			
Δx	<i>Matrix solver</i>	<i>Maximum outer iterations</i>	<i>Relative tolerance</i>
3.18 m	Algebraic multi grid	10	0.0001 %

FIGURE 11.1.7-4
The island aquifer analytical solution compared to IGW – after pumping

Note that the numerical solution is graphically indistinguishable from the exact solution.

11.1.8 1D STEADY-STATE FLOW IN CONFINED AQUIFER (NO RECHARGE)

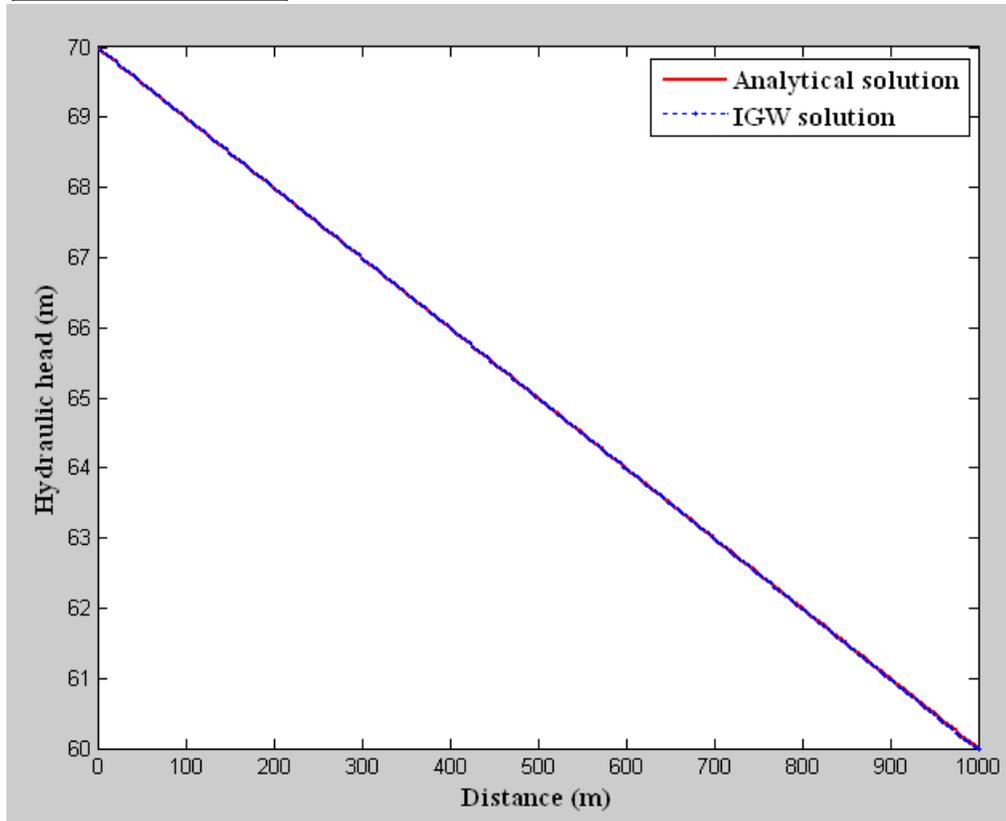
For steady state flow conditions;

$$\frac{dq}{dx} = 0 \Rightarrow kb \frac{d}{dx} \left(\frac{dh}{dx} \right) = 0, \frac{d^2h}{dx^2} = 0$$

$$x = 0 : h = h_0$$

$$x = L : h = h_1$$

$$h(x) = h_0 + \left(\frac{h_1 - h_0}{L} \right) x$$



IGW Numerical Parameters			
Δx	Matrix solver	Maximum inner iterations	Tolerance
4.92 m	Algebraic multi grid	4000	0.000001

11.1.8 STEADY-STATE FLOW WITH RECHARGE - CONFINED

Consider a confined aquifer which is adjacent to two constant head boundaries and a constant recharge infiltrates at the top of the aquifer (see **Figure 11.1.8-1**).

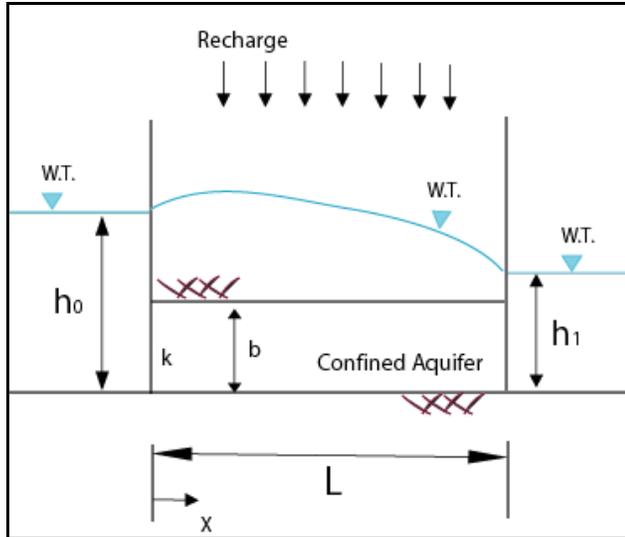


FIGURE 11.1.8-1
Confined aquifer with two constant head boundaries

ANALYTICAL SOLUTION

The analytical solution for steady state flow for head prediction along the aquifer's length (1D) is given by (assuming aquifer homogeneity)

For steady state flow conditions;

$$\frac{dq}{dx} + N = 0 \Rightarrow kb \frac{d}{dx} \left(\frac{dh}{dx} \right) + N = 0, \frac{d^2 h}{dx^2} + \frac{N}{kb} = 0$$

$$x = 0: h = h_0$$

$$x = L: h = h_1$$

$$h(x) = h_0 + \left(\frac{(h_1 - h_0)}{L} + \frac{NL}{2T} \right) x - \frac{N}{T} \frac{x^2}{2}$$

where $h(x)$: predicted head at a given distance, [L]

h_0 : constant head at $x = 0$, [L]

h_1 : constant head at $x = L$, [L]

x : distance from the left boundary, [L]

L : length of the domain, [L]

N : recharge, [L/T]

k : hydraulic conductivity, [L/T]

b : aquifer thickness, [L]

T : transmissivity, [L²/T]

(11.1.8-1)

IGW NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.1.8-1**. Refer to **Figure 11.1.6-2** for a plan view of the IGW model set up (note that unconfined is replaced with confined).

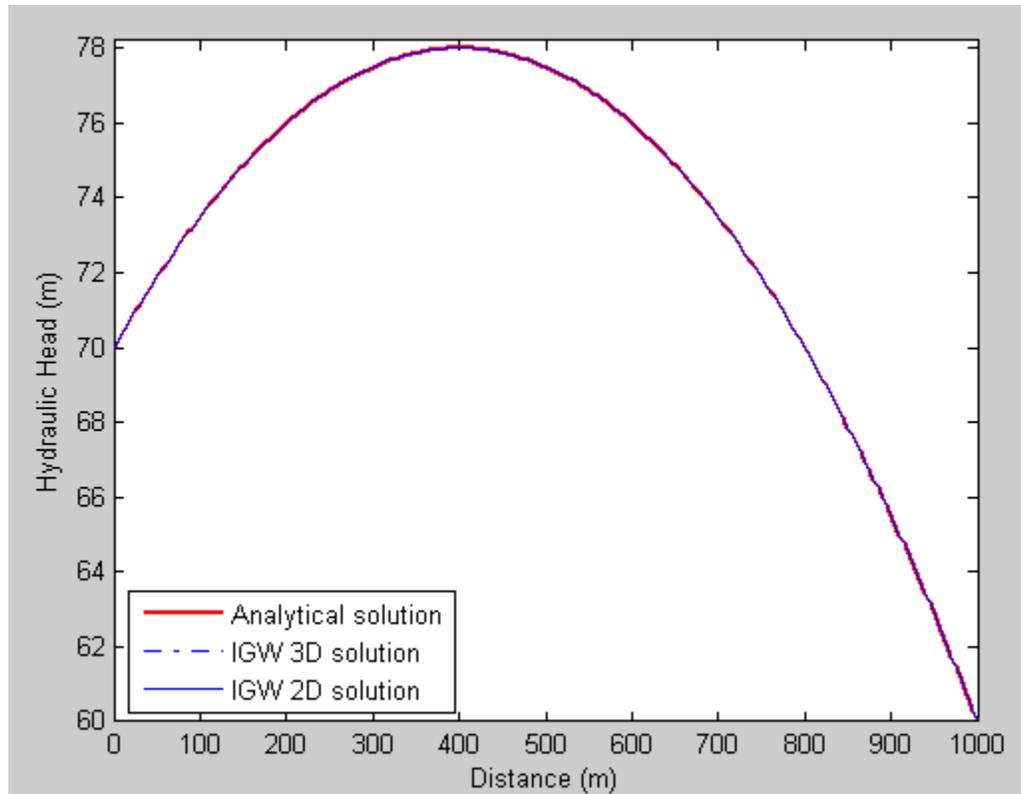
TABLE 11.1.8-1

Assumptions related to IGW numerical solution

Physical Parameters					Numerical Parameters	
<i>Aquifer Top / Bottom</i>	h_0	h_l	r	L	Δx	Δy
-10 m and -50 m	20 m	10 m	-0.2 m/day	1000 m	5.5 m	5.5 m

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Figure 11.1.8-2 shows a comparison between the analytical solution and the IGW solution.

**FIGURE 11.1.8-2**

Comparison of the analytical solution and IGW

IGW Numerical Parameters			
Δx	<i>Matrix solver</i>	<i>Maximum inner iterations</i>	<i>Tolerance</i>
3.18 m	Algebraic multi grid	4000	0.000001

Note that the numerical solution is graphically indistinguishable from the exact solution.

11.1.9 STEADY-STATE FLOW IN CONFINED AQUIFER WITH VARIABLE THICKNESS

$$\frac{d}{dx} \left[T(x) \frac{dh}{dx} \right] = 0$$

$$T(x) = kt(x)$$

where $k = \text{hydraulic conductivity}$,

$t(x) = ax + b$, general thickness function (aquifer thickness is changing linearly)

$$T(x) \frac{dh}{dx} = C_1$$

$$kt(x) \frac{dh}{dx} = C_1, \frac{dh}{dx} = \frac{C_1}{k(ax+b)}, \boxed{h = \frac{C_1}{k} \frac{\ln(ax+b)}{a} + C_2} \text{ (General solution)}$$

$$\text{At } x = 0, t(0) = 30 \text{ m}$$

$$\text{At } x = 1000, t(1000) = 20 \text{ m}$$

Thickness function, $t(x) = 30 - 10 \frac{x}{L}$, where $L = 1000 \text{ m}$

$$\boxed{h = -\frac{1}{10} \frac{C_1 L}{k} \ln(30L - 10x) + C_2}$$

$$x = 0: h = 60 \text{ m}$$

$$x = 1000: h = 50 \text{ m}$$

$$k = 50 \text{ m/day}$$

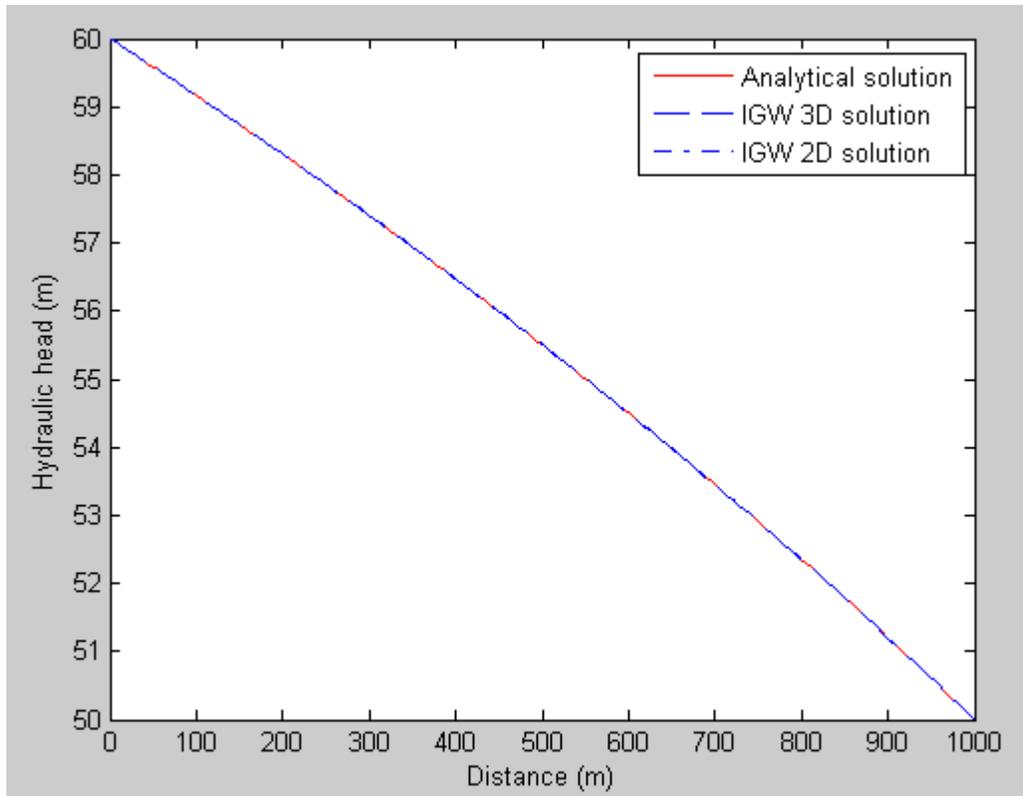
$$60 = -\frac{1000C_1}{500} \ln(30000) + C_2, 60 = -20.6179C_1 + C_2$$

$$50 = -\frac{1000C_1}{500} \ln(20000) + C_2, 50 = -19.8070C_1 + C_2$$

$$\boxed{C_1 = -12.3320}$$

$$\boxed{C_2 = -194.2599}$$

$$\boxed{h = \frac{1.2332L}{k} \ln(30L - 10x) - 194.2599}$$



CONFINED AQUIFER WITH A VARIABLE THICKNESS OF EXPONENTIAL FUNCTION

$$\frac{d}{dx} \left[kh(x) \frac{dh}{dx} \right] = 0$$

$$T(x) = kt(x)$$

where k = hydraulic conductivity,

$$t(x) = b_0 e^{-ax}, \text{ general thickness function}$$

$$Kb_0(-a)e^{-ax} \frac{dh}{dx} + Kb_0 e^{-ax} \frac{d^2h}{dx^2} = 0$$

$$\frac{d^2h}{dx^2} - a \frac{dh}{dx} = 0$$

$$h(x) = c_1 + c_2 e^{ax} \text{ (General solution)}$$

$$x = 0: h = H_1, \quad H_1 = c_1 + c_2$$

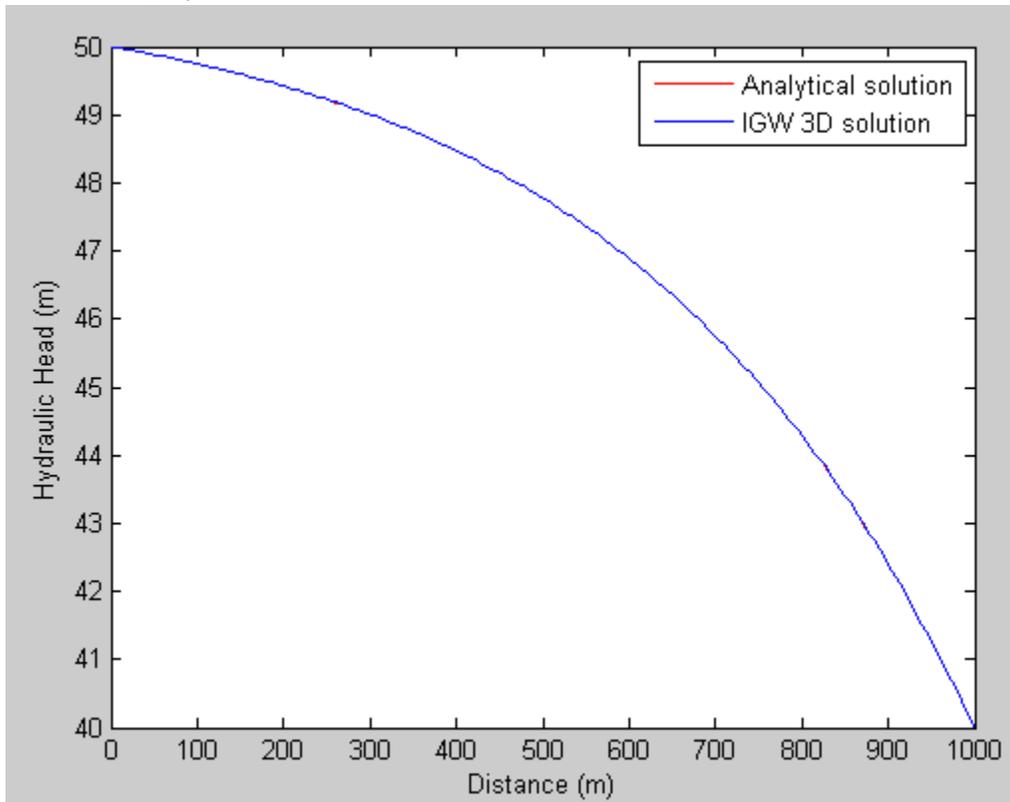
$$x = L: h = H_2, \quad H_2 = c_1 + c_2 e^{aL}$$

$$c_1 = H_2 - c_2$$

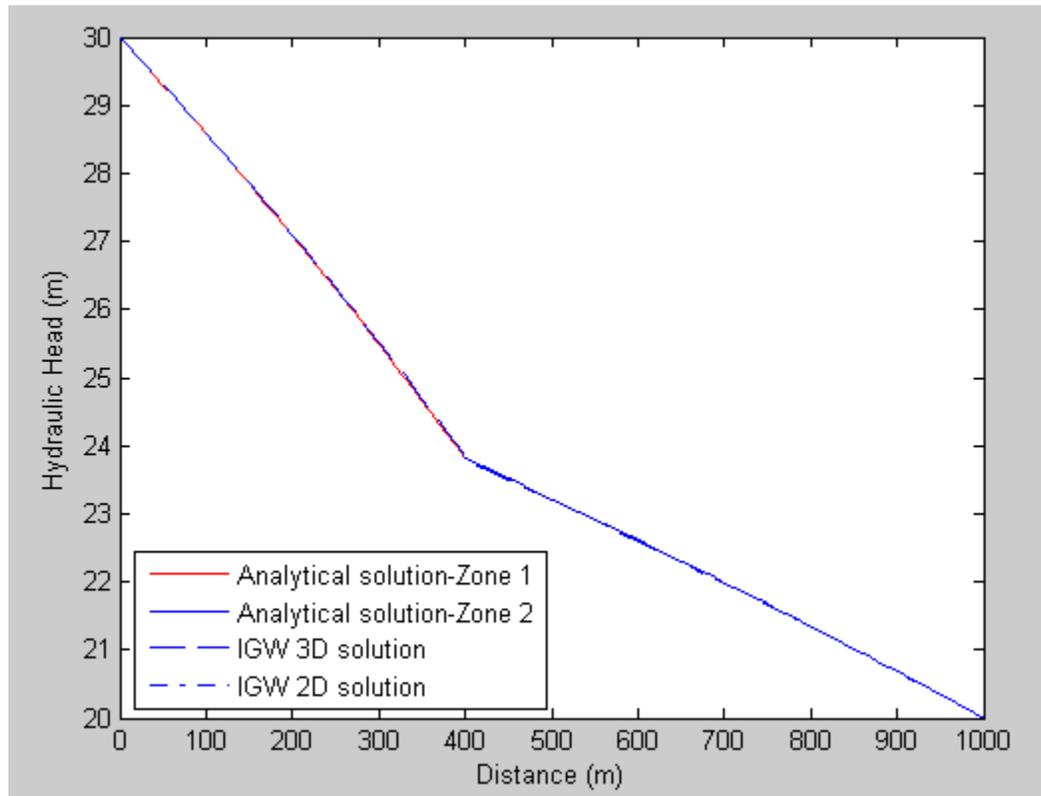
$$H_2 = H_1 - c_2 + c_2 e^{aL}$$

$$c_2 = \frac{H_1 - H_2}{1 - e^{aL}}, \quad c_1 = \frac{H_2 - H_1 e^{aL}}{1 - e^{aL}}$$

$$h(x) = \frac{H_2 - H_1 e^{aL} + (H_1 - H_2) e^{ax}}{1 - e^{aL}}$$



11.2 UNCONFINED AQUIFER WITH VARIABLE HYDRAULIC CONDUCTIVITY



CONFINED AQUIFER WITH VARIABLE HYDRAULIC CONDUCTIVITY

For steady state flow conditions; $\frac{d}{dx} \left[k(x) \frac{dh}{dx} \right] = 0$

where $k(x)$ = linear hydraulic conductivity function,

$$k(x) \frac{dh}{dx} = A$$

$$k(x) \frac{dh}{dx} = A, \frac{dh}{dx} = \frac{A}{ax+b}, \boxed{h = A \frac{\ln(ax+b)}{a} + B} \text{ (General solution)}$$

$$x = 0: k(0) = 20 \text{ m/d}$$

$$k_1(x) = 20 + 40 \frac{x}{L_1}, 0 < x < 340 \text{ (conductivity is increasing linearly)}$$

$$k_2(x) = 60 - 50 \frac{(x-340)}{L_2}, 340 < x < 1000 \text{ (conductivity is decreasing linearly)}$$

where L_1, L_2 : domain size of the conductivity zones

$$\int_0^{L_1} \frac{\partial}{\partial x} \left(k_1(x) \frac{\partial h}{\partial x} \right) dx + \int_{L_1}^{L_2} \left(k_2(x) \frac{\partial h}{\partial x} \right) dx = 0$$

$$x = 1000: k(1000) = 10 \text{ m/d}$$

$$x = 0: h_1 = 30 \text{ m}$$

$$x = 1000: h_3 = 20 \text{ m}$$

i) Boundary conditions on each extremes can be satisfied as;

$$x = 0: 30 = \frac{1}{40} C_1 340 \ln(20 * 340 + 40 * 0) + C_2$$

$$\boxed{30 = 75.0098 C_1 + C_2}$$

$$x = 1000: 20 = -\frac{1}{50} C_3 660 \ln(60 * 660 - 50(1000 - 340)) + C_4$$

$$\boxed{20 = -116.0917 C_3 + C_4}$$

ii) Hydraulic heads at the interface separating the two zones should be equal to each other;

$$h|_{x=340}^- = h|_{x=340}^+$$

$$\frac{1}{40} C_1 340 \ln(20 * 340 + 40 * 340) + C_2 = -\frac{1}{50} C_3 660 \ln(60 * 660 - 50(1000 - 340)) + C_4$$

$$\boxed{84.3480 C_1 + C_2 = -139.7429 C_3 + C_4}$$

iii) Gradients at the interface separating the two zones should be equal to each other;

$$\boxed{\partial h|_{x=340}^- = \partial h|_{x=340}^+}$$

$$\frac{\partial}{\partial x} \left(\frac{1}{40} C_1 340 \ln(20 * 340 + 40x) + C_2 \right) = \frac{\partial}{\partial x} \left(-\frac{1}{50} C_3 660 \ln(60 * 660 - 50x) + C_4 \right)$$

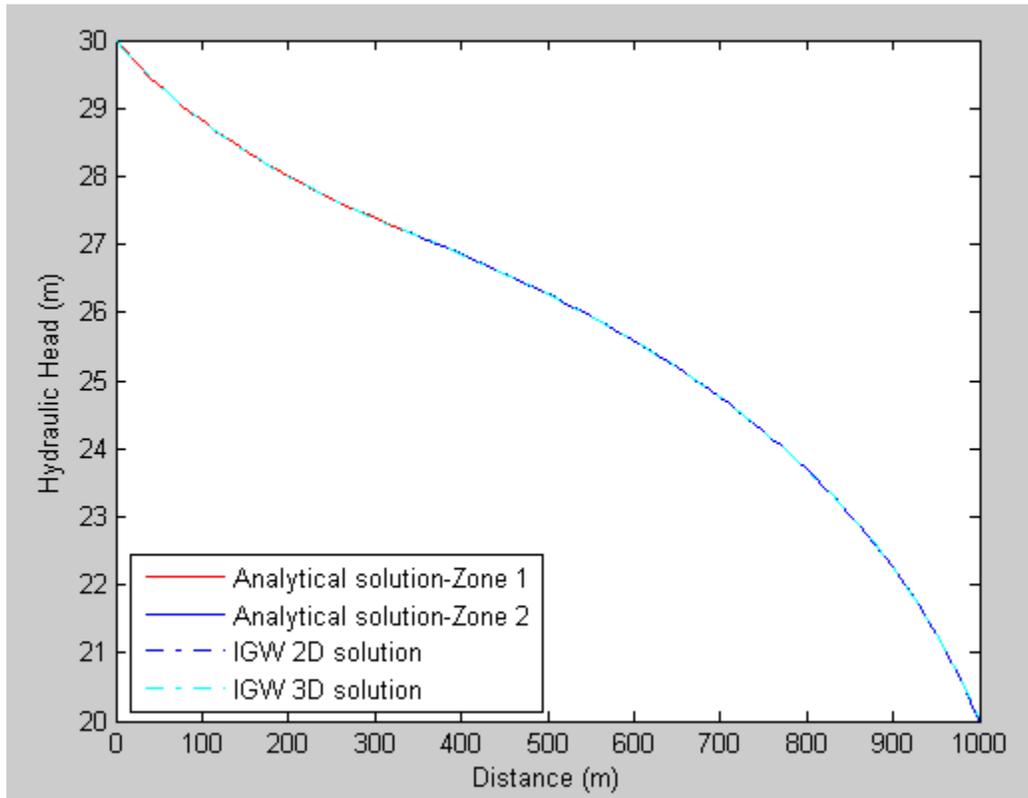
$$\frac{340 C_1}{20400} = \frac{660 C_3}{39600}, \boxed{C_1 = C_3}$$

$$\boxed{C_1 = C_3 = -0.3031}, \boxed{C_2 = 52.7355}, \boxed{C_4 = -15.1874}$$

Using the constants, we can find the head distributions;

$$\boxed{h_1(x) = -2.5763 \ln(6800 + 40x) + 52.7355}$$

$$\boxed{h_2(x) = 4.0009 \ln(39600 - 50x) - 15.1874}$$



CONFINED AQUIFER WITH ONE VARIABLE HYDRAULIC CONDUCTIVITY ZONE

For steady state flow conditions; $\frac{d}{dx} \left[k(x) \frac{dh}{dx} \right] = 0$

where $k(x)$ = linear hydraulic conductivity function,

$$k(x) \frac{dh}{dx} = A$$

$$k(x) \frac{dh}{dx} = A, \quad \frac{dh}{dx} = \frac{A}{ax+b}, \quad \boxed{h = A \frac{\ln(ax+b)}{a} + B} \quad (\text{General solution})$$

$$x = 0: k(0) = 20 \text{ m/d}$$

$$k(x) = 20 + 40 \frac{x}{L}, \quad 0 < x < 1000 \quad (\text{conductivity is increasing linearly})$$

where L : length of the aquifer

$$x = 0: h_1 = 30 \text{ m}$$

$$x = 1000: h_3 = 20 \text{ m}$$

According to the continuity assumption, hydraulic gradients in two zones are equal to each other;

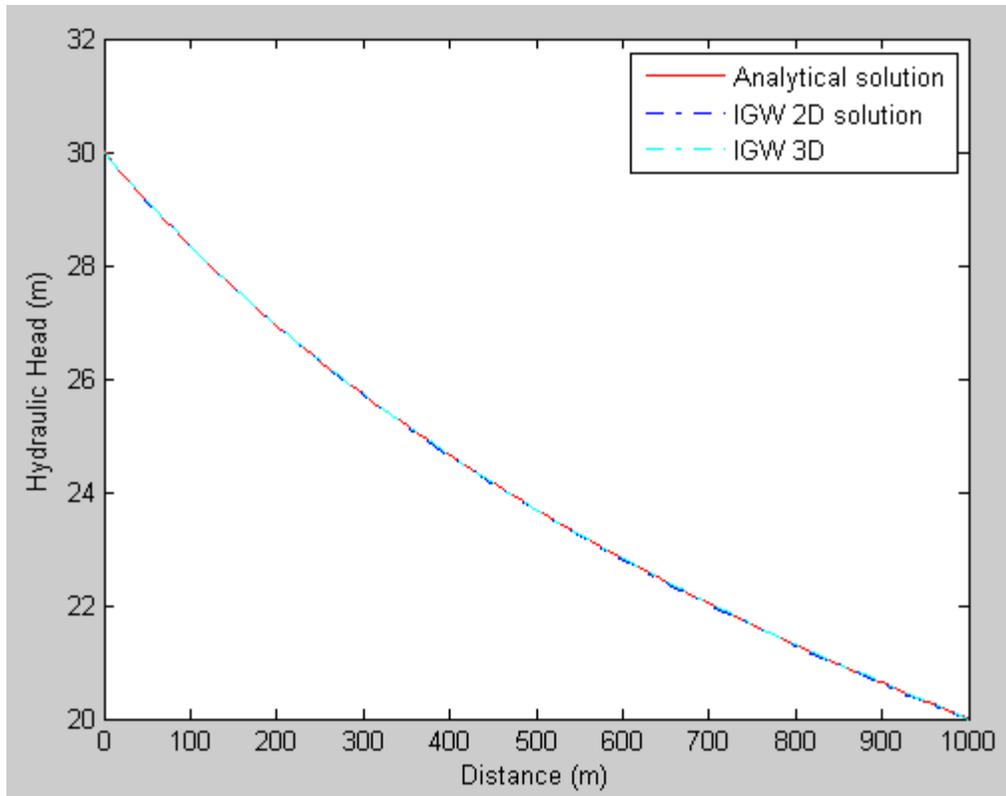
$$\boxed{h = \frac{1}{40} C_1 L_1 \ln(20L + 40x) + C_2, \quad 0 < x < L}$$

We can find using constant head boundary conditions on each sides of the problem domain;

$$C_1 = -0.3641 \text{ and } C_2 = 120.147$$

Hydraulic head function can be defined as;

$$\boxed{h = (-9.1025) \ln(20000 + 40x) + 120.147, \quad 0 < x < 1000}$$



TOTH SOLUTION

Assumptions:

1. Aquifer is isotropic and homogeneous which is bounded with an impermeable basement.
2. Flow is restricted to a two-dimensional vertical section and water table can be approximated by a sine wave.
3. Upper boundary of the two-dimensional vertical section is the water table.

Toth considered a sinusoidal water table with a regional slope of the form

$$h(x, z_0) = \left[z + \frac{B'x}{L} + b \sin\left(\frac{2\pi x}{\lambda}\right) \right]$$

where z : vertical distance from the major topographic low to the bottom of the aquifer, [L],

B' : height of the major topographic high above the major topographic low, [L],

x : horizontal distance from the major topographic low to the major topographic high, [L],

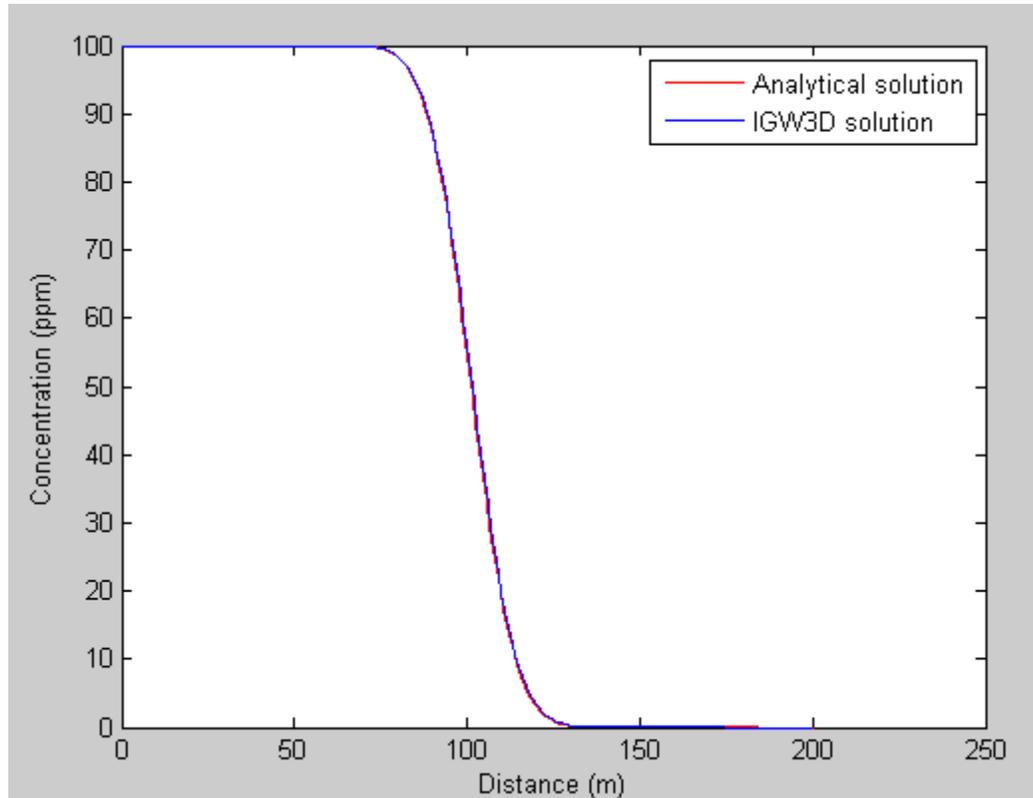
L : total horizontal length of the aquifer, [L],

λ : number of oscillations from the major topographic low to the major topographic high, [L]

11.2 TRANSPORT SOLVER VERIFICATION EXERCISES

CONCENTRATION IN THE RESERVOIR IS CONSTANT

$$C = \frac{C_0}{2} \left(\operatorname{erfc} \left(\frac{x-vt}{\sqrt{4Dt}} \right) + e^{\left(\frac{xv}{D} \right)} \operatorname{erfc} \left(\frac{x+vt}{\sqrt{4Dt}} \right) \right)$$



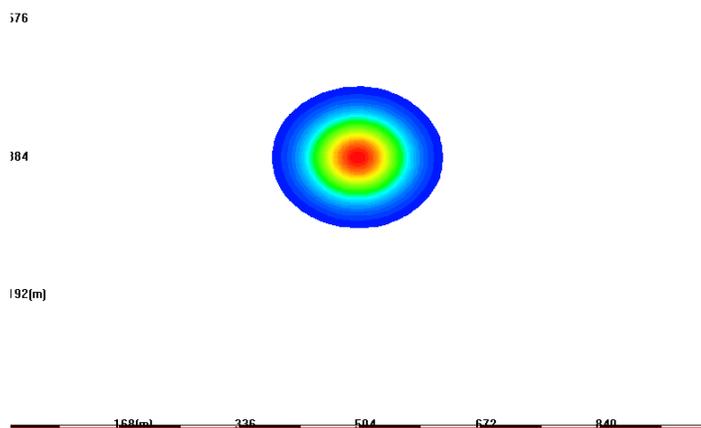
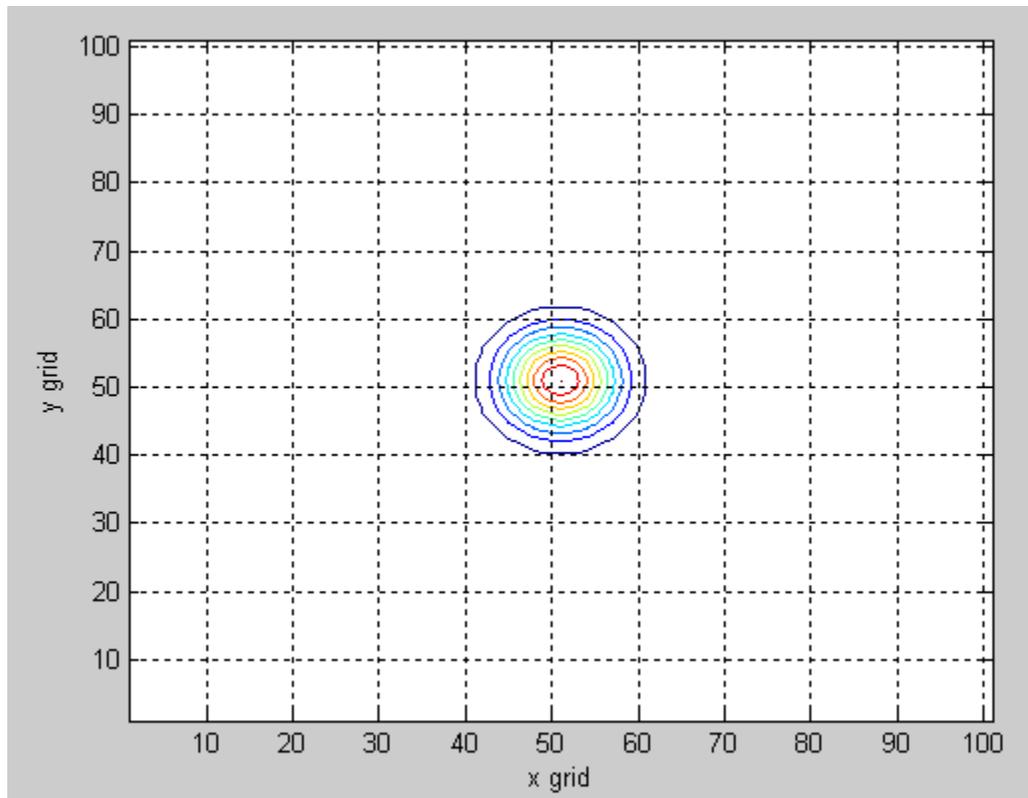
The exercises presented in the following subsections test various aspects of the IGW 3 transport solver.

DISPERSION IN RADIAL FLOW

$$C = \frac{C_0}{2} \operatorname{erfc} \left[\left(\frac{r_D^2}{2} - t_D \right) \left(\frac{4}{3} r_D^3 \right)^{-1/2} \right]$$

$$r_D = \frac{r}{\alpha_L}, t_D = \frac{Qt}{2\pi bn\alpha_L^2}$$

11.2.1 ADVECTION AND ANISOTROPIC DISPERSION



An analytical solution for the migration of a gaussian plume in uniform flow (see **Figure 11.2.1-1**) was presented by Baetsle^x as

$$C(x, y, t) = \frac{C_m \sigma_{x_0} \sigma_{y_0}}{\sigma_x \sigma_y} e^{-\frac{(X-x_0)^2}{2\sigma_x^2} - \frac{(Y-y_0)^2}{2\sigma_y^2}} \quad (11.2.1-1)$$

where C_m = the initial concentration of the gaussian plume [ML^{-3}],

σ_{x_0} = the initial standard deviation at x direction [L],

σ_{y_0} = the initial standard deviation at y direction [L],

x_0, y_0 = the central coordinate of the gaussian plume[L],

D_L = the longitudinal dispersion coefficient [L^2T^{-1}],

D_T = the transverse dispersion coefficient [L^2T^{-1}],

\bar{v}_x = the velocity of uniform flow in x direction [LT^{-1}]

and where

$$\sigma_x^2 = \sigma_{x_0}^2 + 2D_L t, \quad (11.2.1-2)$$

$$\sigma_y^2 = \sigma_{y_0}^2 + 2D_T t, \quad (11.2.1-3)$$

$$X = x - \bar{v}_x t, \quad (11.2.1-4)$$

and

$$Y = y. \quad (11.2.1-5)$$

NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.2.1-1**.

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Figures 11.2.1-2, 11.2.1-3, and 11.2.1-4 show comparisons between the analytical and numerical solutions. Note that the two solutions are indistinguishable. **Figure 11.2.1-5** shows the legend.

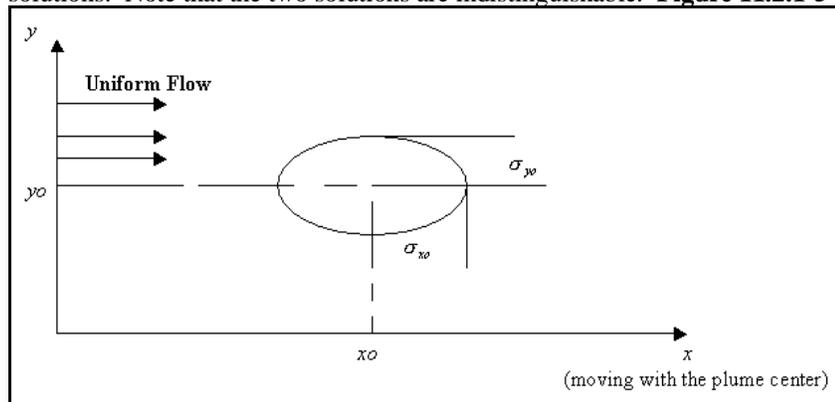
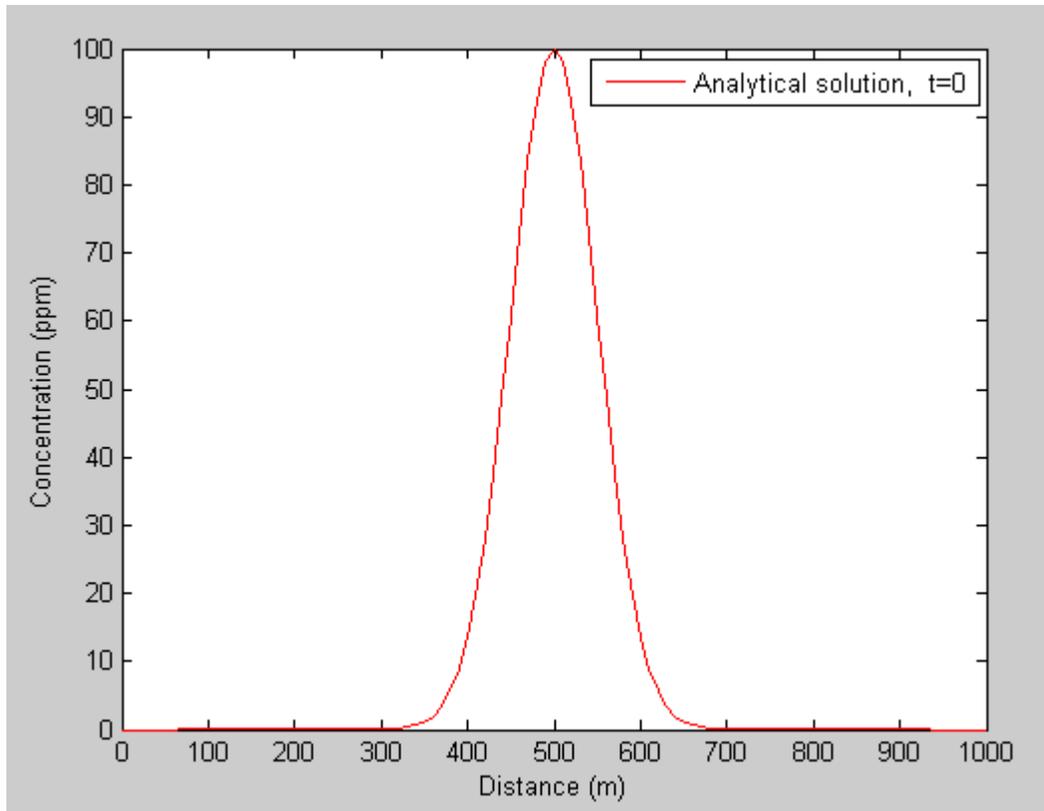
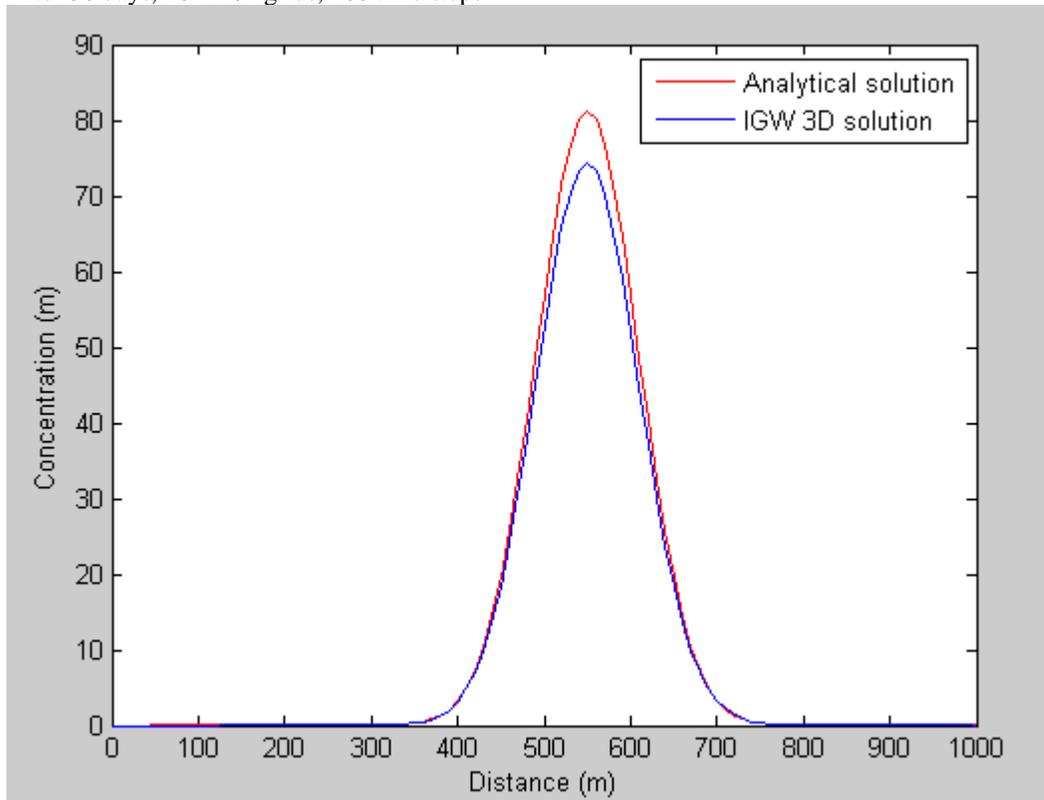


FIGURE 11.2.1-1

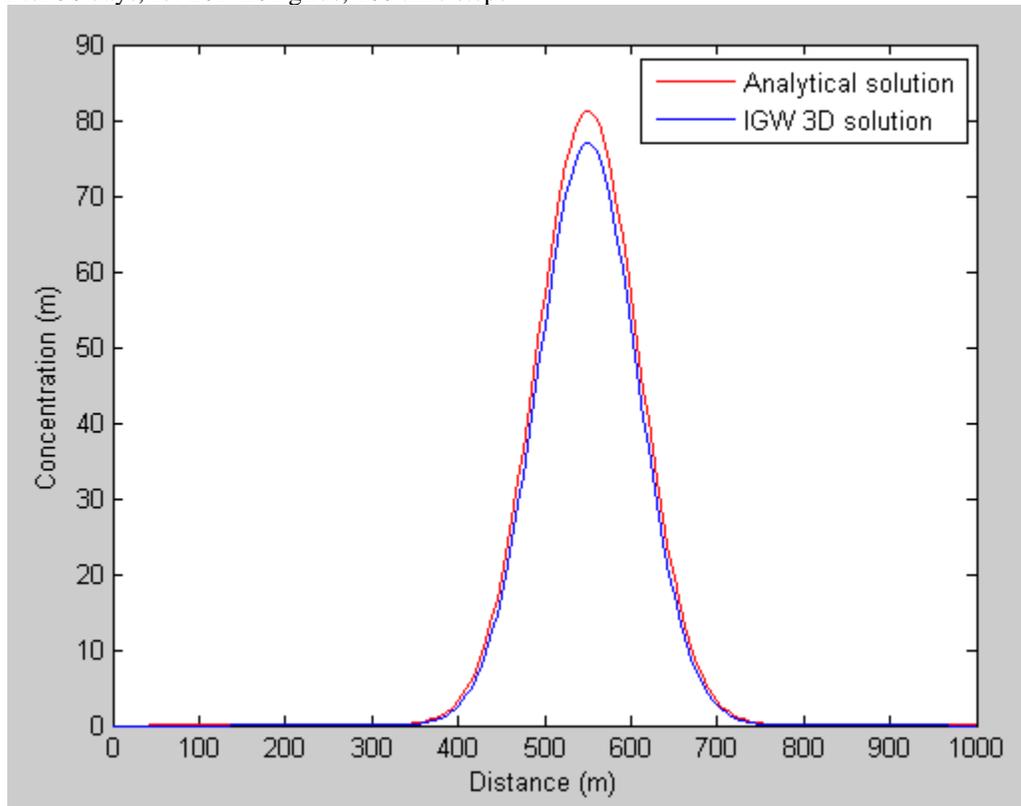
The initial contamination is a gaussian plume centered at x_0, y_0



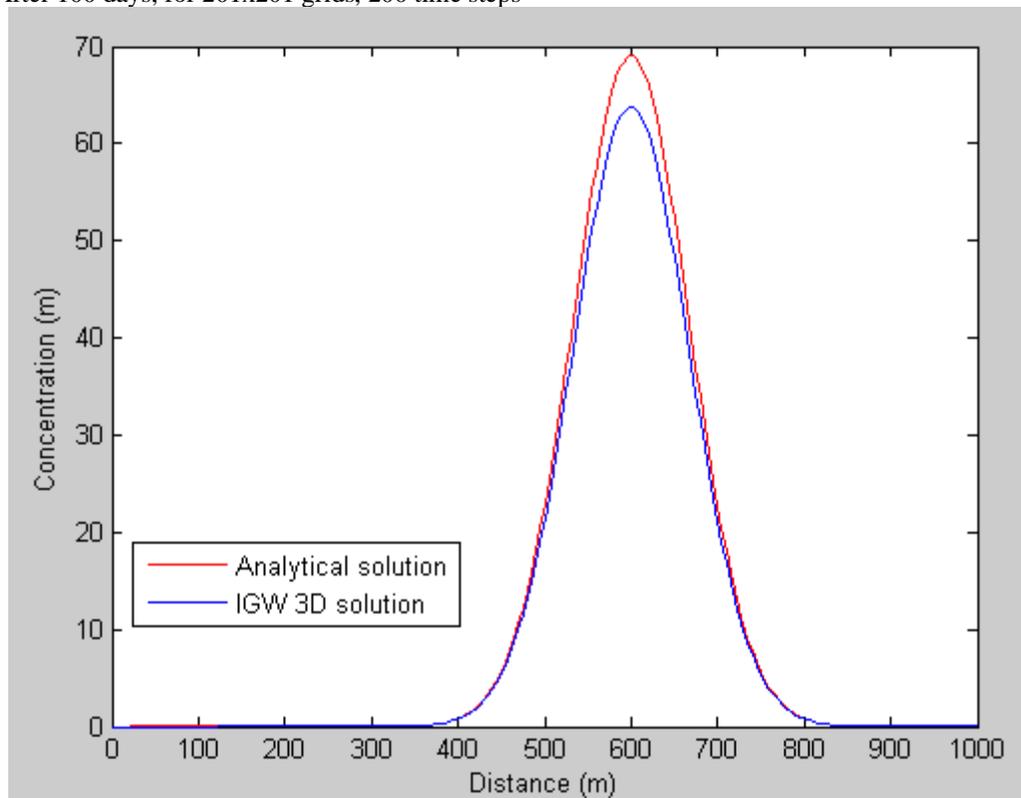
After 50 days, 101x101 grids, 100 time steps



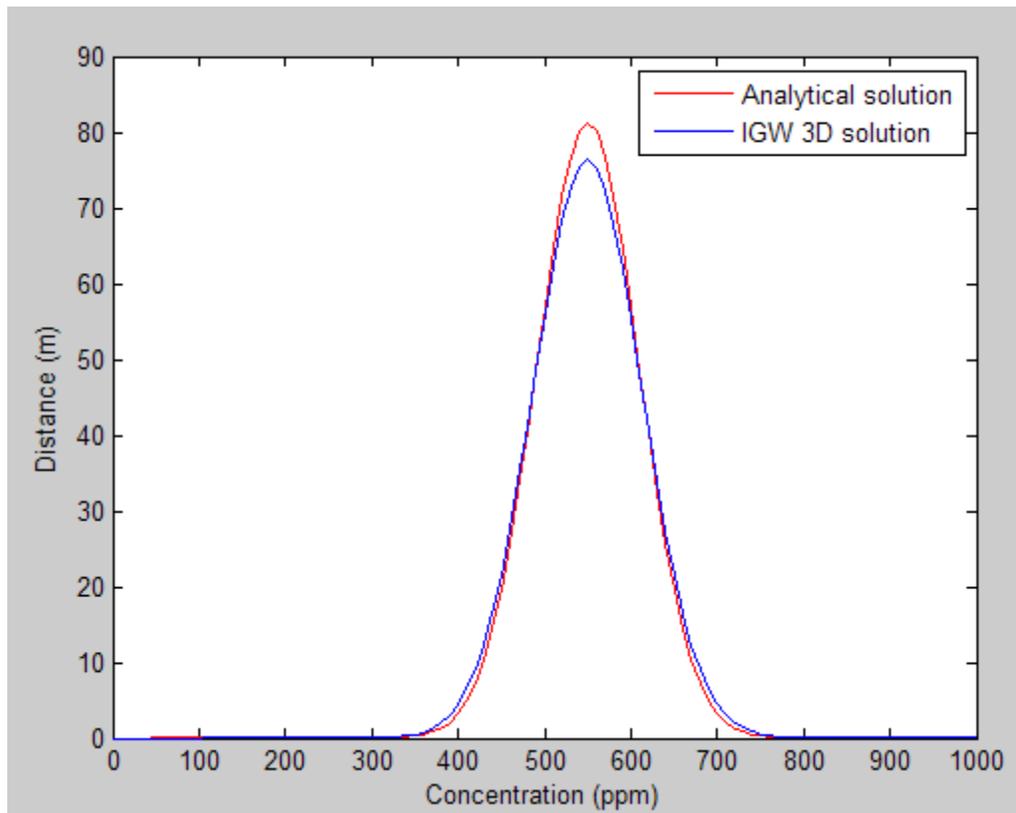
After 50 days, for 201x201 grids, 100 time steps



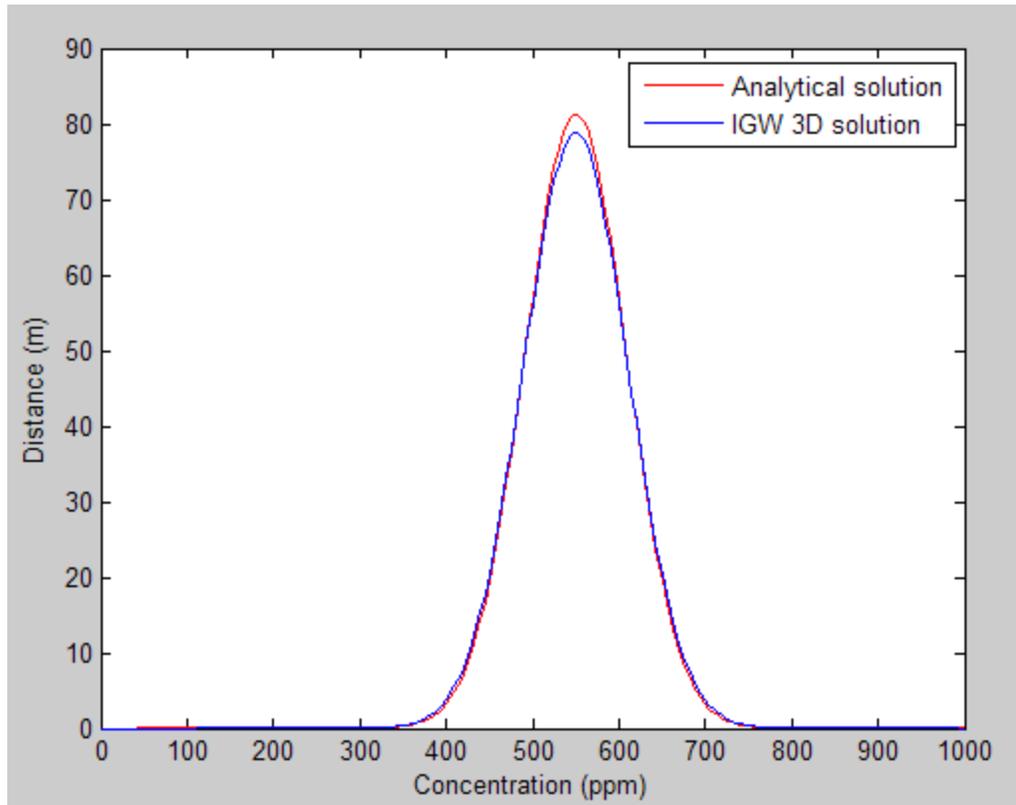
After 100 days, for 201x201 grids, 200 time steps

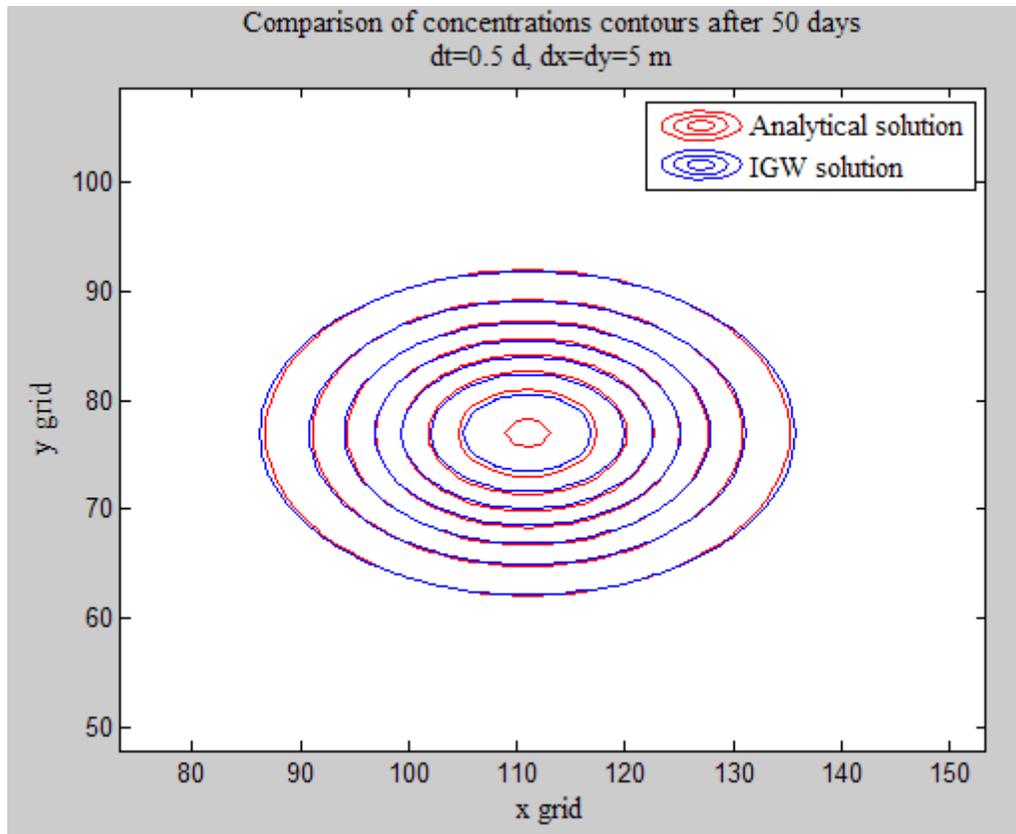


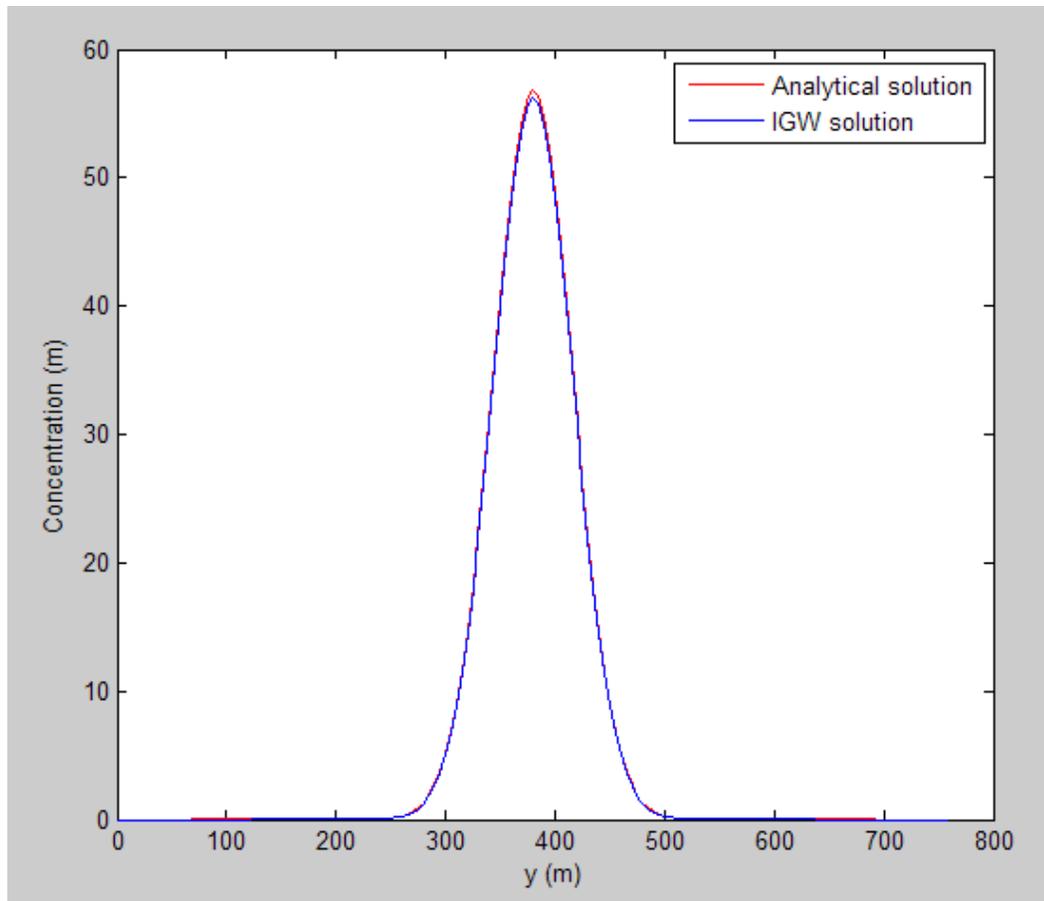
After 50 days, for 101x77, 100 time steps, dx=10 dy=10



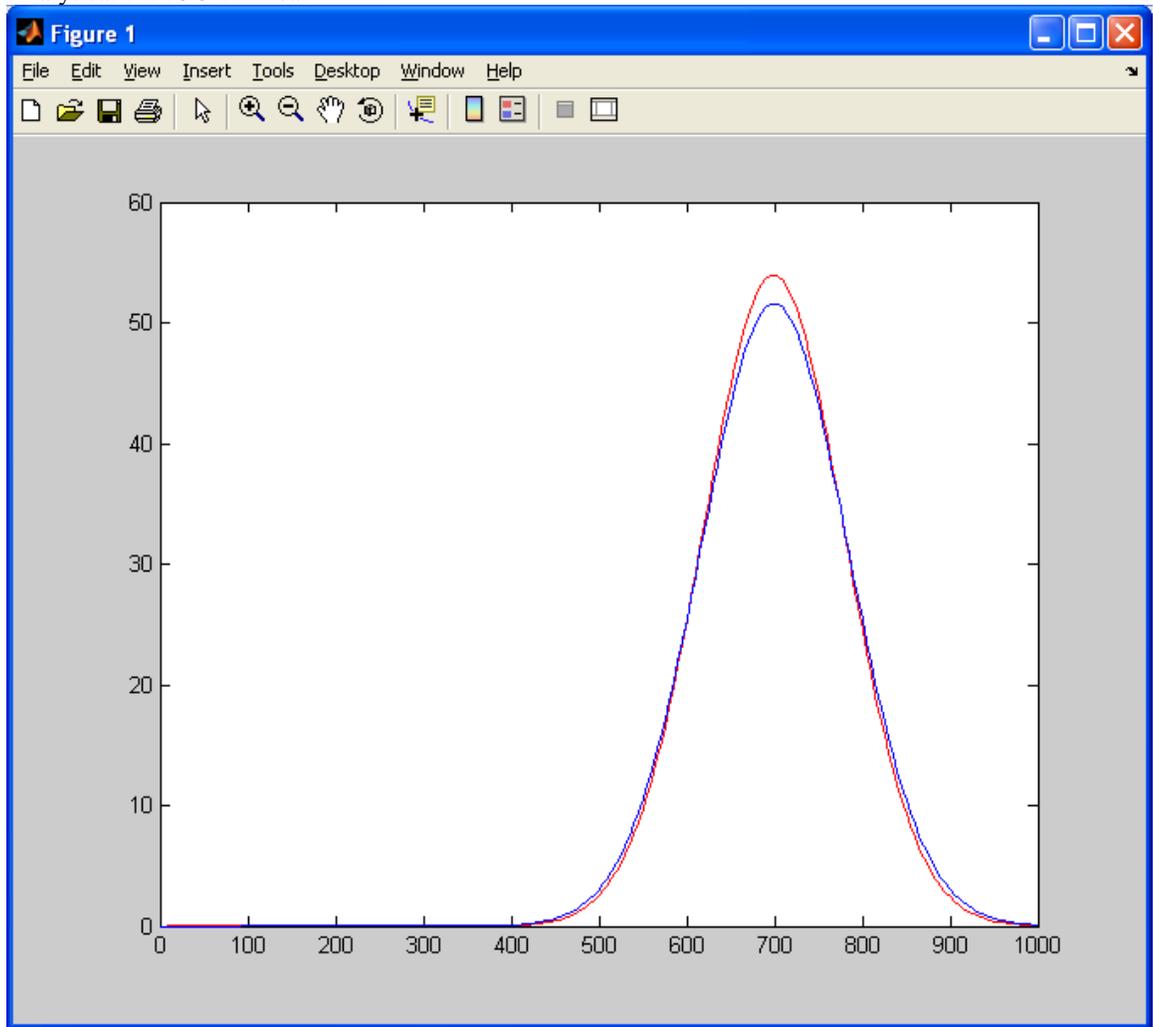
After 50 days, for 201x152, 100 time steps, dx=5 dy=5

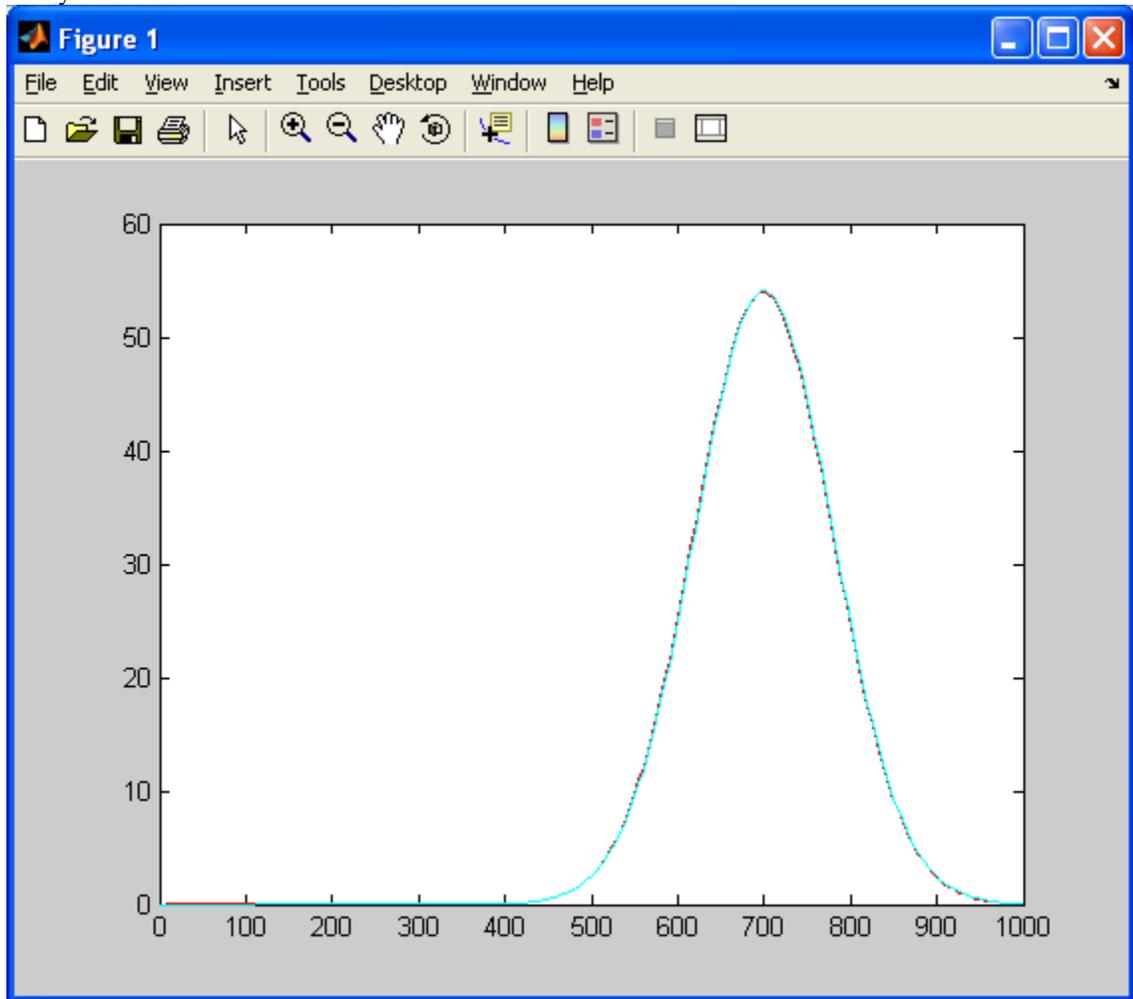






Analytical-MMOC Trilinear





Analytical-others

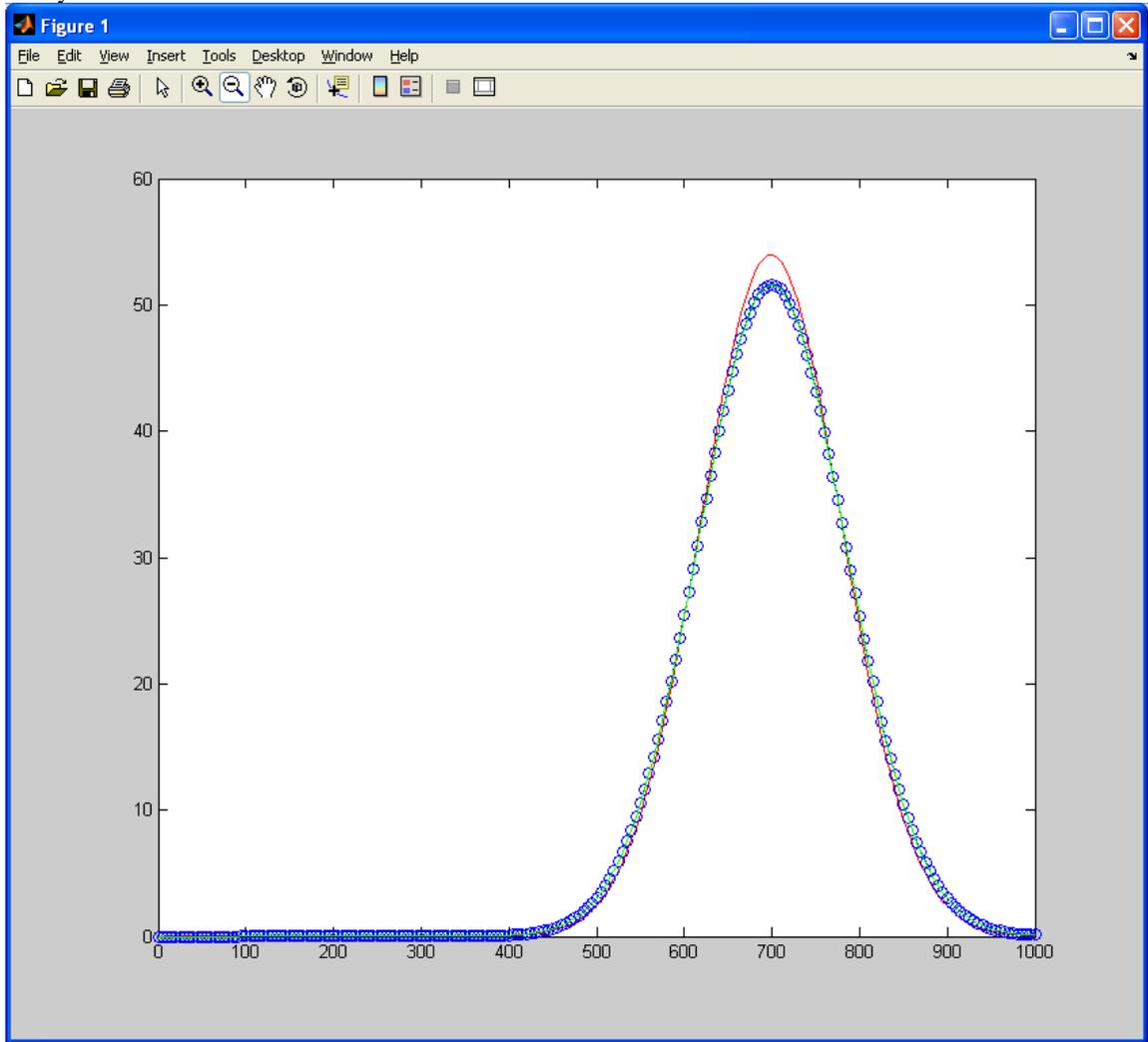


TABLE 11.2.1-1

Assumptions related to IGW numerical solution

Physical Parameters								Numerical Parameters		
σ_{x_0}	σ_{y_0}	D_L	x_0	\bar{v}_x	C_m	D_T	y_0	Δx	Δy	Δt
50 m	35 m	10 m ² /d	500 m	1 m/d	100 ppm	1 m ² /d	375 m	10 m	10 m	0.5 d

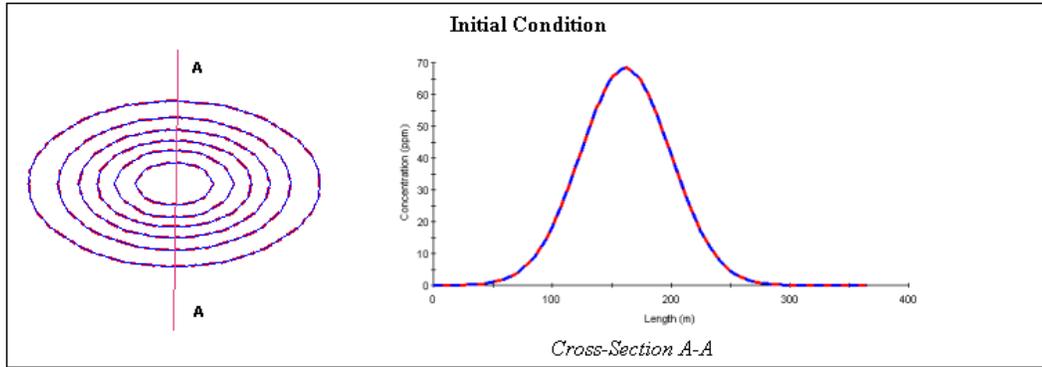


FIGURE 11.2.1-2
Comparison between IGW and analytical solutions at initial conditions

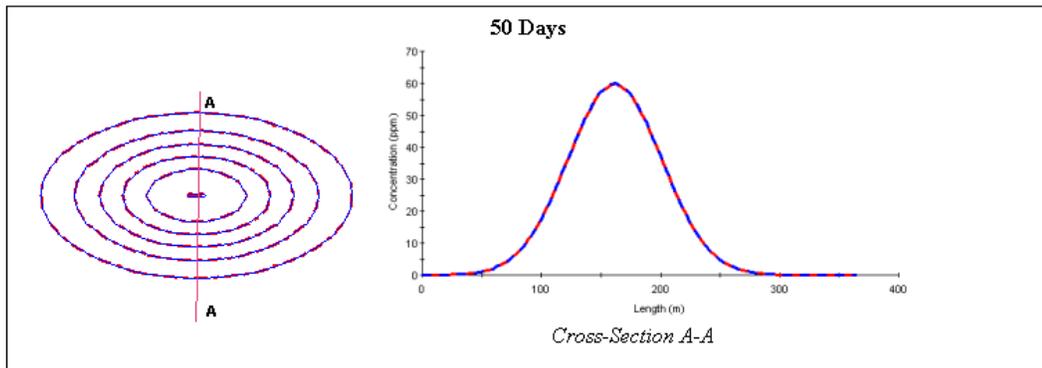


FIGURE 11.2.1-3
Comparison between IGW and analytical solutions at 50 days

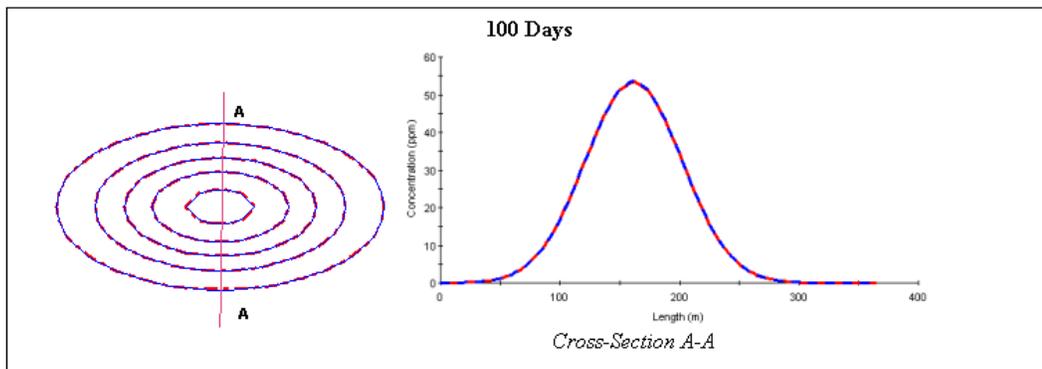


FIGURE 11.2.1-4
Comparison between IGW and analytical solutions at 100 days

IGW		Exact	
-----	60	————	60
-----	50	————	50
-----	40	————	40
-----	30	————	30
-----	20	————	20
-----	10	————	10

FIGURE 11.2.1-5
Legend for Figures 11.2.1-2, 11.2.1-3, and 11.2.1-4

11.2.2 ADVECTION AND ANISOTROPIC DISPERSION WITH ANGLE

A special case of the solution presented in **Section 11.2.1** where there is an orientation in the uniform flow to the horizontal axes (illustrated in **Figure 11.2.2-1**), is presented in this section.

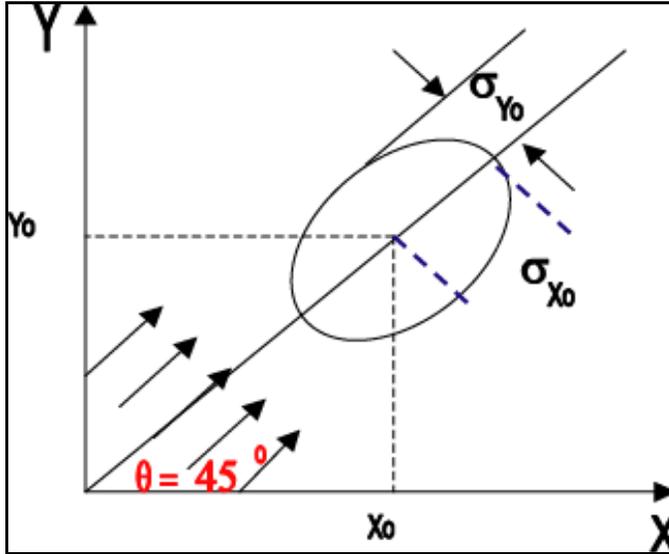


FIGURE 11.2.2-1
The initial contamination is a gaussian plume centered at x_0, y_0 ; note the angle of orientation

For this special case

$$X = x' - \bar{v}_x t \quad (11.2.2-1)$$

and

$$Y = y' - \bar{v}_y t \quad (11.2.2-2)$$

where \bar{v}_y = the velocity of uniform flow the y direction [L/T].

These equations replace **Equations 11.2.1-4** and **11.2.1-5**. Note that

$$x' = x \cos(\theta) + y \sin(\theta) \quad (11.2.2-3)$$

and

$$y' = y \cos(\theta) - x \sin(\theta). \quad (11.2.2-4)$$

where θ = the angle of uniform flow with the horizontal axis [-].

NUMERICAL SOLUTION

IGW is applied to solve the flow problem given the assumptions presented in **Table 11.2.2-1**.

TABLE 11.2.2-1

Assumptions related to IGW numerical solution

Physical Parameters								Numerical Parameters		
σ_{x_0}	σ_{y_0}	D_L	x_0	\bar{v}_x, \bar{v}_y , and θ	C_m	D_T	y_0	Δx	Δy	Δt
50 m	35 m	10 m ² /d	500 m	10 m/d, 1 m/d and 45°	100 ppm	1 m ² /d	375 m	25 m	25 m	0.5 d

COMPARISON OF ANALYTICAL SOLUTION AND IGW SOLUTION

Figure 11.2.2-2 shows a comparison between the IGW 3 solution and the analytical solution.

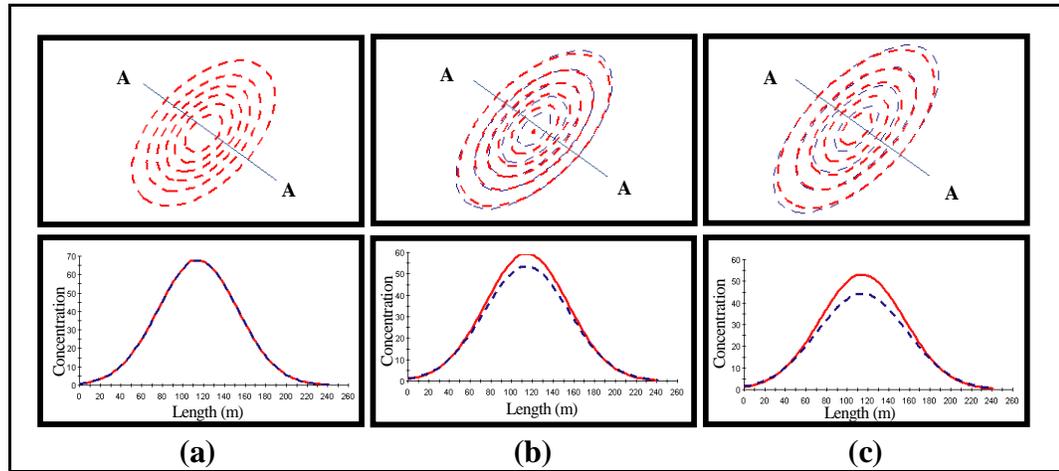


FIGURE 11.2.2-2

Comparison of the IGW solution and the analytical solution at: (a) 0 days, (b) 50 days, and (c) 100 days

Note that the results obtained from the IGW solution are physically realistic and show no unphysical oscillations or negative concentrations.

Figure 11.2.2-3 shows a comparison between the traditional FD method (not the IGW 3 RCVT method) and the analytical solution.

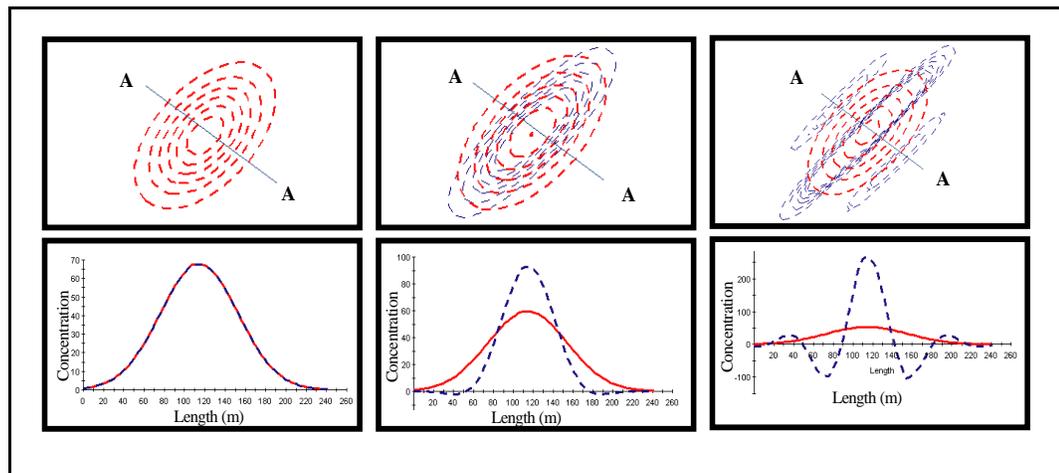


FIGURE 11.2.2-3

Comparison of the traditional FD solution and analytical solution at: (a) 0 days, (b) 50 days, and (c) 100 days

Notice the unphysical oscillations and negative concentrations that plague the traditional FD method.

A comparison between the analytical solution, IGW RCVT solution method and the traditional FD solution method is illustrated in **Figure 11.2.2-4**.

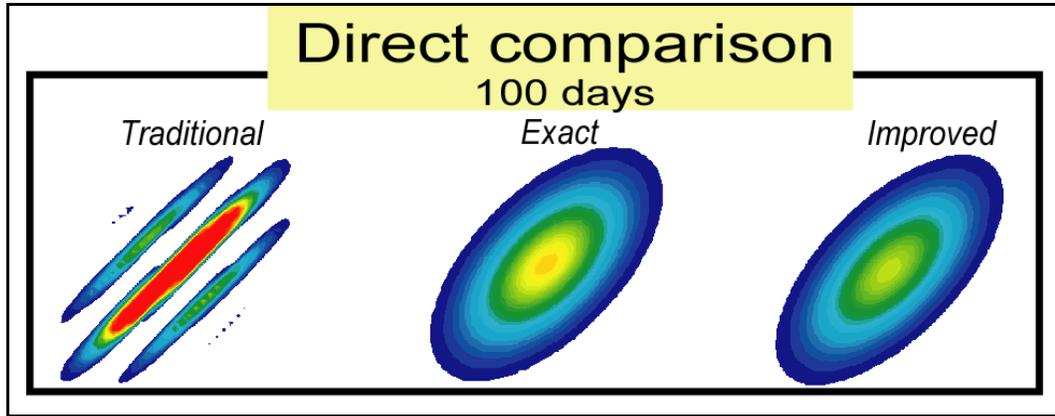


FIGURE 11.2.2-4
Comparison of the analytical solution, the IGW 3 RCVT solution method and the traditional FD solution method

The results indicate that the IGW solution and the analytical solution provide very similar solutions, however, the traditional method provides significantly different and incorrect results.

11.3 TRANSPORT SOLVER VERIFICATION EXERCISES

This section presents comparisons of measured and simulated heads for solutions from 3-D models, PV2-D models, and traditional 2-D models. The comparisons are based on a number of different cases and are organized into the following subsections as such.

A number of parameters were held consistent for each case and are given in **Table 11.3-1**.

TABLE 11.3-1
Parameters consistent for each case

Parameter	Value
Processor	Intel® Pentium 4 @ 1.7 GHz
Grid Dimensions	x = 4.0 m; y= 4.0 m; z = 4.0 m

11.3.1 UNIFORM FLOW CONDITION

Figure 11.3.1-1 shows a conceptual representation of the modeled situation.

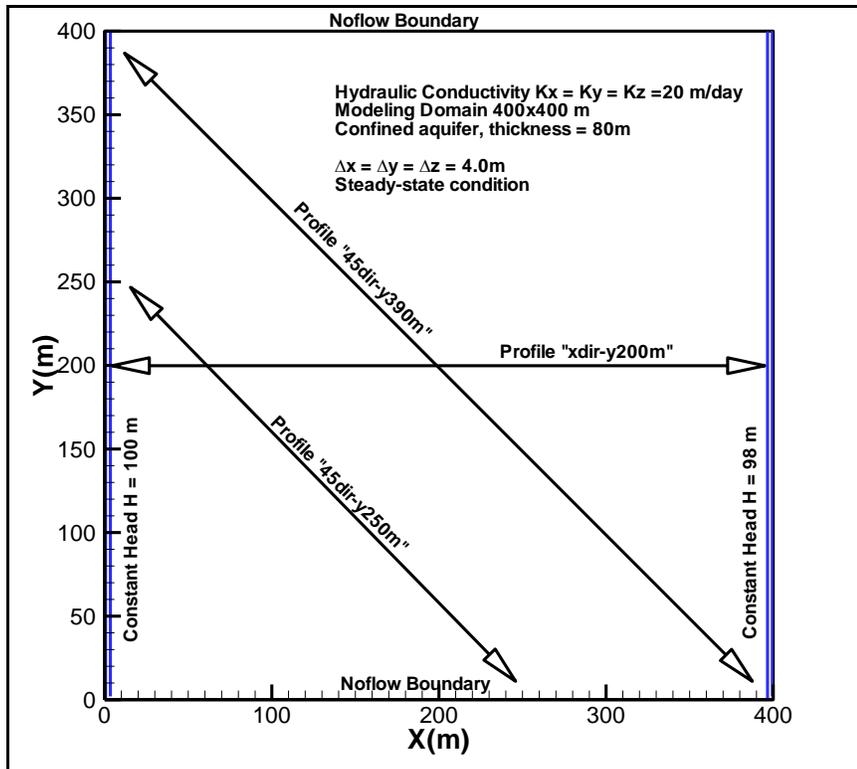


FIGURE 11.3.1-1
 Conceptual representation for uniform flow case

Figure 11.3.1-2 compares the full 3-D model solution and the plane 2-D solution. These results are identical.

Figures 11.3.1-3, 11.3.1-4, and 11.3.1-5 compare the layer-based 3-D, PV2-D, and traditional profile 2-D model solutions for the profile 'xdir-y200m', '45dir-y390m', and '45dir-y250m', respectively. The three models yield identical results for each of the three cases.

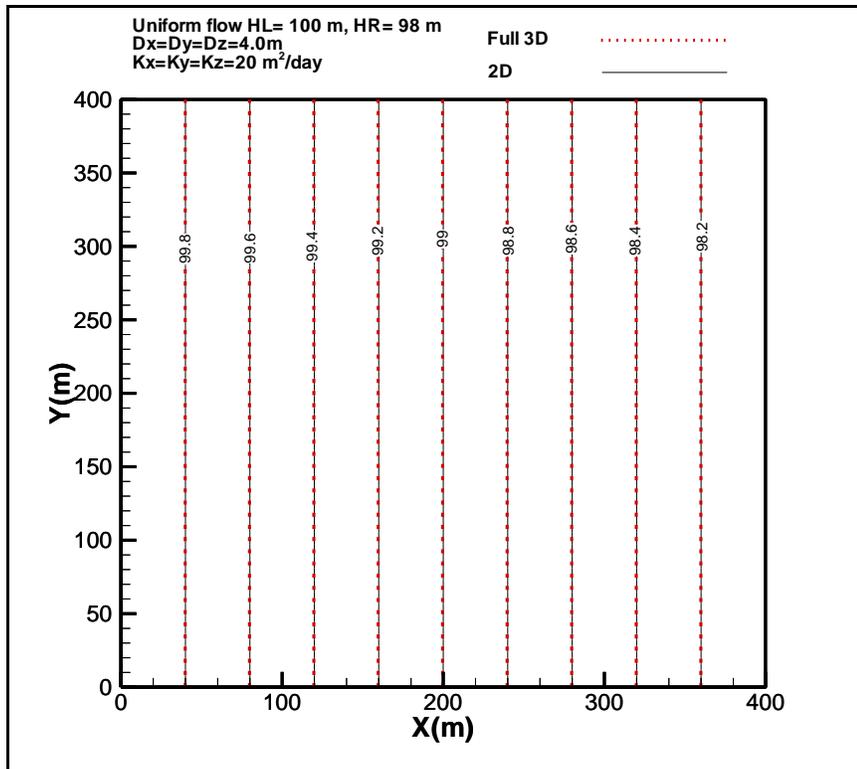


FIGURE 11.3.1-2
 Comparison of 3-D and plane 2-D solutions

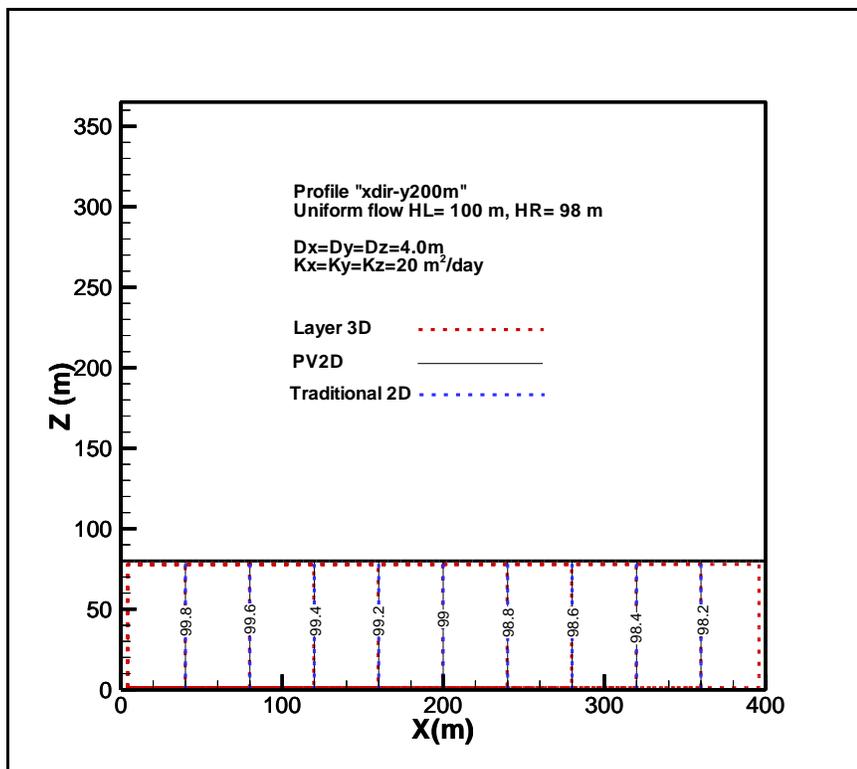


FIGURE 11.3.1-3
 Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the 'xdir-y200m' profile

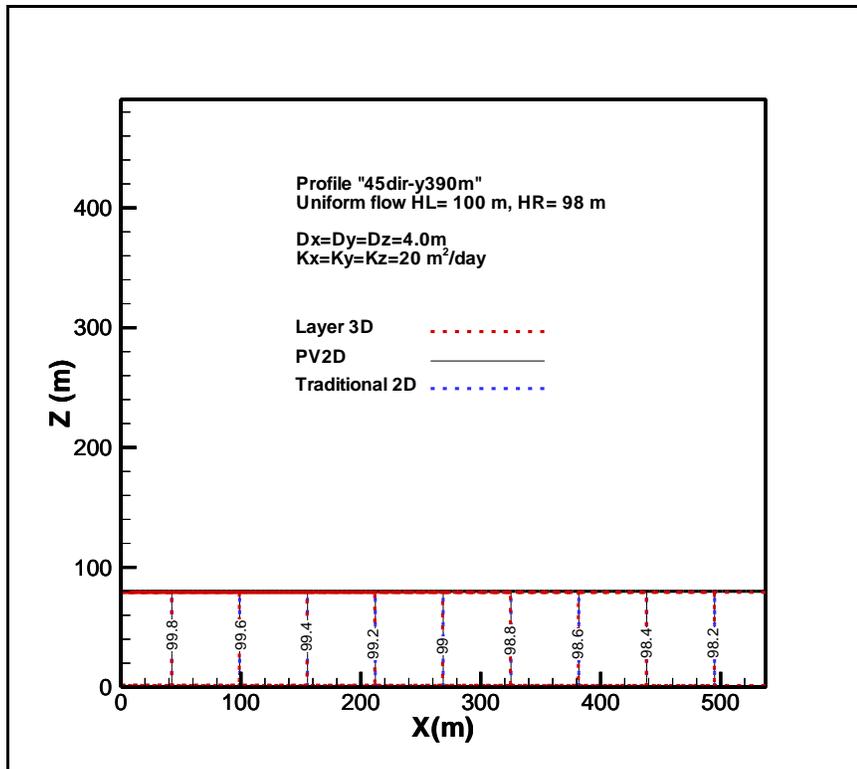


FIGURE 11.3.1-4
 Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the '45-y390m' profile

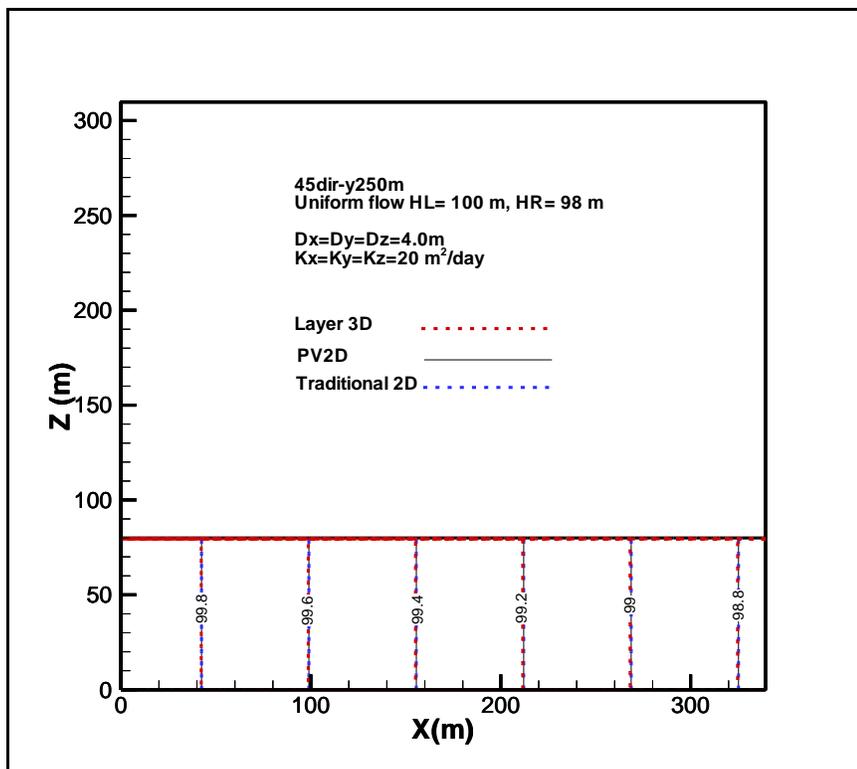


FIGURE 11.3.1-5
 Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the '45dir-y250m' profile

11.3.2 CONSTANT HEAD CONDITION

Figure 11.3.2-1 shows a conceptual representation of the modeled situation.

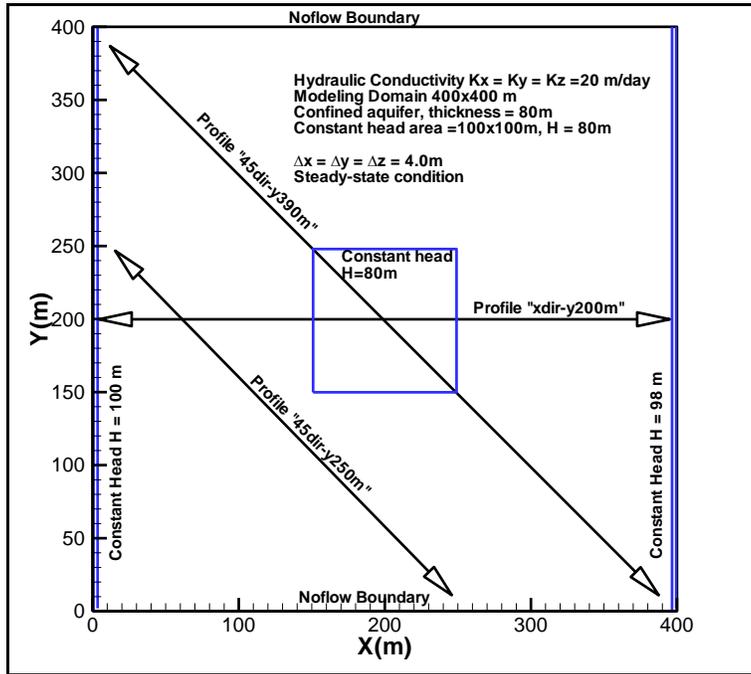


FIGURE 11.3.2-1
Conceptual representation for the constant head case

Figure 11.3.2-2 compares the full 3-D solution and plane 2-D solution. The results are identical.

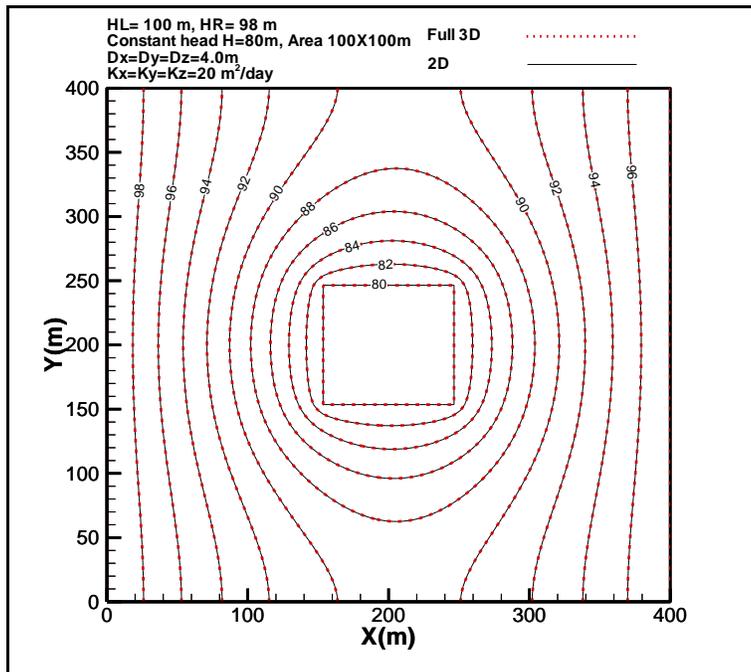


FIGURE 11.3.2-2
Comparison of 3-D and plane 2-D solutions

Figures 11.3.2-3, 11.3.2-4, and 11.3.2-5 compare the layer-based 3-D, PV2-D, and traditional profile 2-D model solutions for the profile 'xdir-y200m', '45dir-y390m', and '45dir-y250m', respectively.

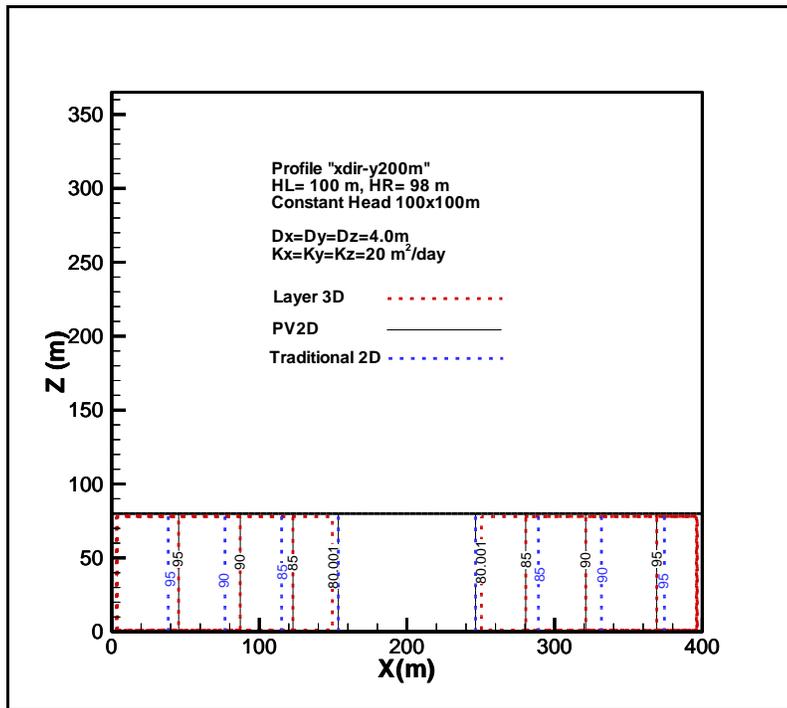


FIGURE 11.3.2-3
Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the 'xdir-y200m' profile

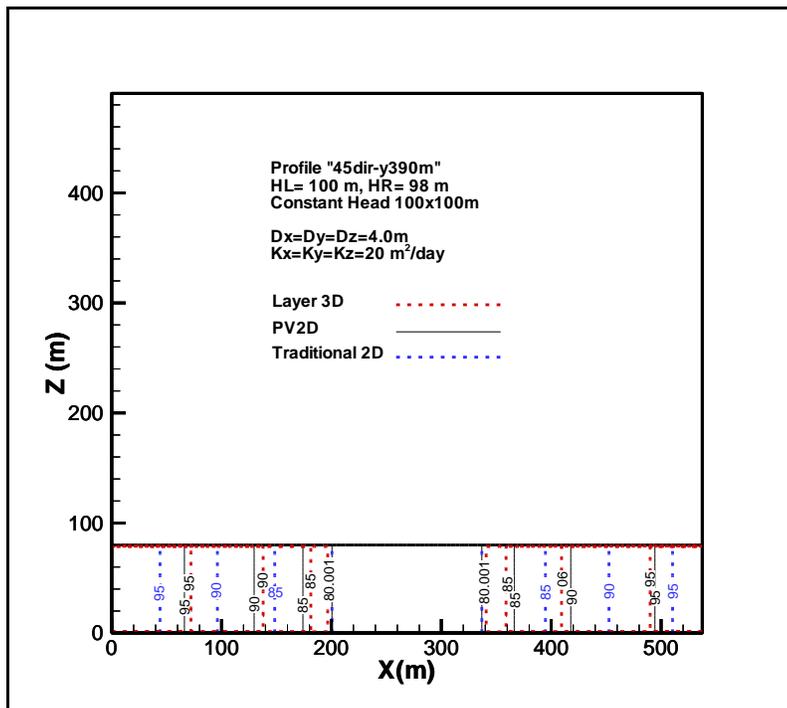


FIGURE 11.3.2-4
Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the '45-y390m' profile

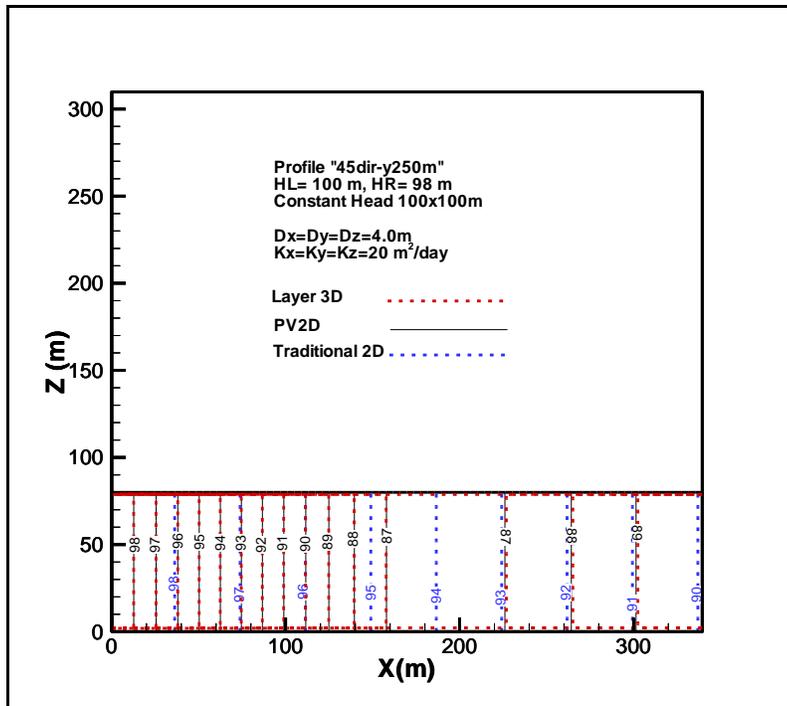


FIGURE 11.3.2-5
Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the ‘45dir-y250m’ profile

The profile model comparisons show that the results from the PV2-D model are identical to the results from the 3-D model. This is due to the fact that the PV2-D model considers cross-flow in its solution. The traditional profile 2-D model does not agree with the 3-D/PV2-D solution as it shows uniform head distributions along each of the cross-sections.

11.3.3 PARTIALLY PENETRATED CONSTANT HEAD CONDITION

Figure 11.3.3-1 shows a conceptual representation of the modeled situation.

Figures 11.3.3-2 and 11.3.3-3 compare the full 3-D solution and the plane 2-D solution (for a constant head feature leakance of 10/day and 50/day, respectively in the 2-D model). The plane 2-D model shows results that deviate from the 3-D model solution. This occurs because the plane 2-D model can only show the depth-averaged flow pattern and thus the vertical nuances introduced by the partially penetrating constant head feature are not fully realized. The highest deviations are found nearest to the constant head feature.

Figures 11.3.3-4, 11.3.3-5, 11.3.3-6, and 11.3.3-7 compare the full 3-D, PV2-D, and traditional profile 2-D model solutions for the profile ‘xdir-y200m’ with the leakance value of 10/day, ‘xdir-y200m’ with the leakance value of 50/day, ‘45dir-y390m’ with the leakance value of 10/day, and ‘45dir-y250m’ with the leakance value of 10/day, respectively. The results for the PV2-D model are more similar to the 3-D model than the traditional profile 2-D model, but they deviate greatly in the portion of the model below the lake. This occurs because the PV2-D model receives only average cross-flux data from the plane 2-D model and thus cannot resolve the vertical variations.

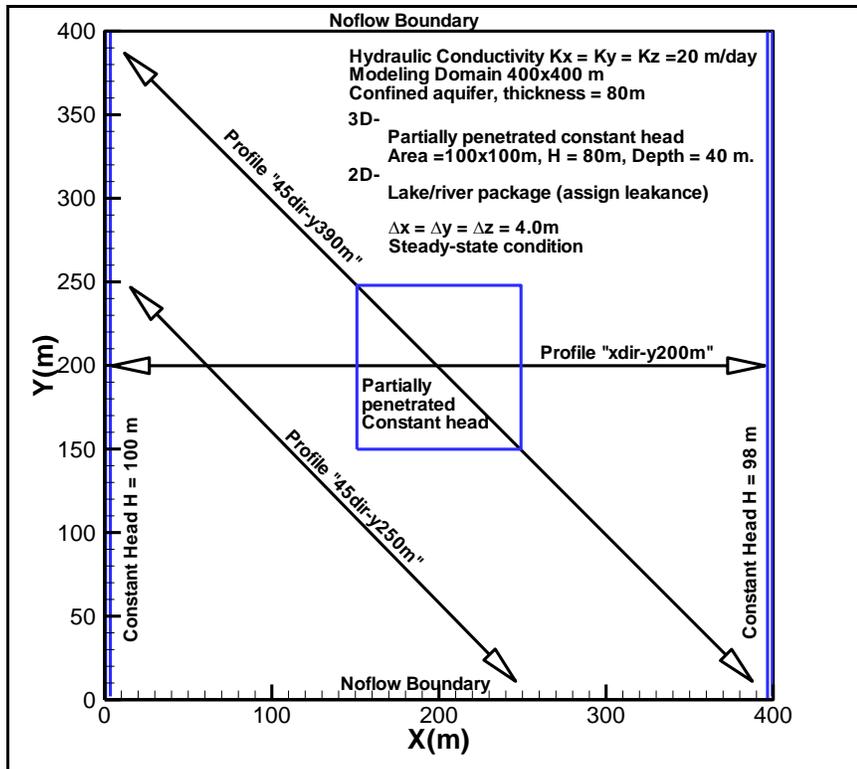


FIGURE 11.3.3-1
 Conceptual representation for the partially penetrating constant head case

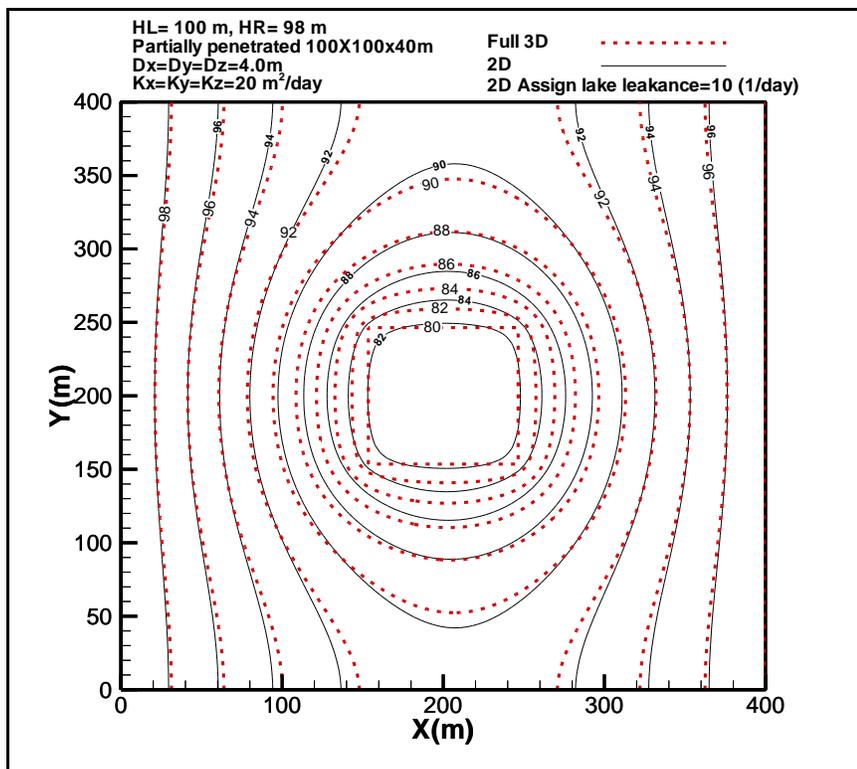


FIGURE 11.3.3-2
 Comparison of 3-D and plane 2-D solutions with constant head feature in 2-D model assigned leakage of 10/day

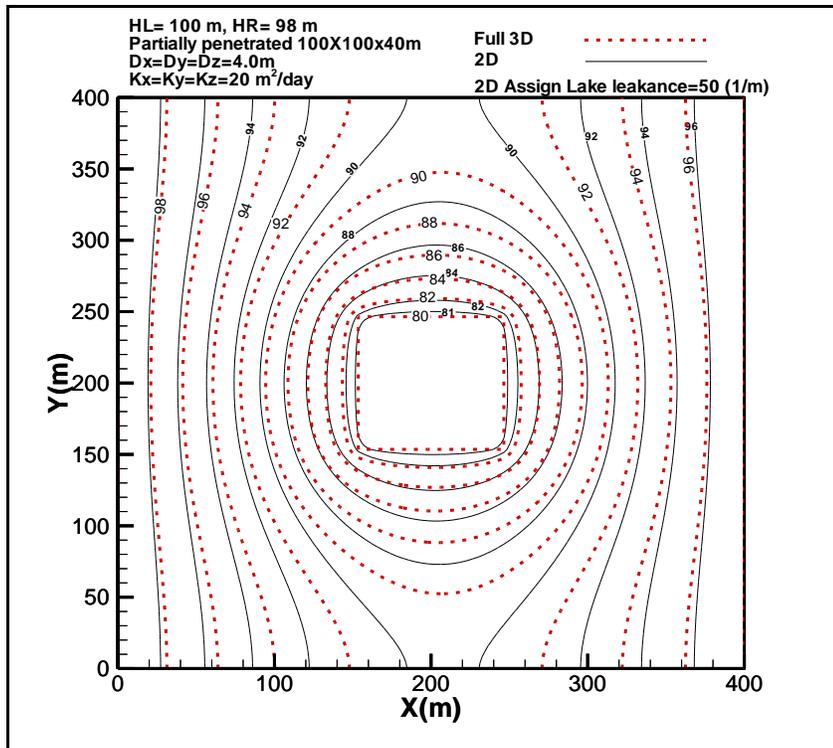


FIGURE 11.3.3-3
Comparison of 3-D and plane 2-D solutions with constant head feature in 2-D model assigned leakance of 10/day

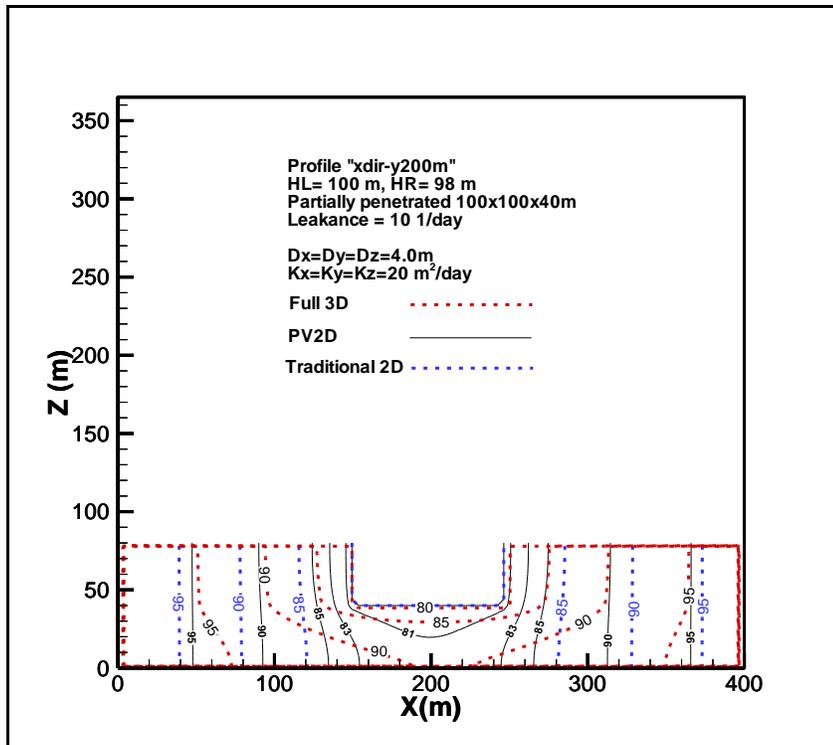


FIGURE 11.3.3-4
Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the 'xdir-y200m' profile with the leakance value of 10/day

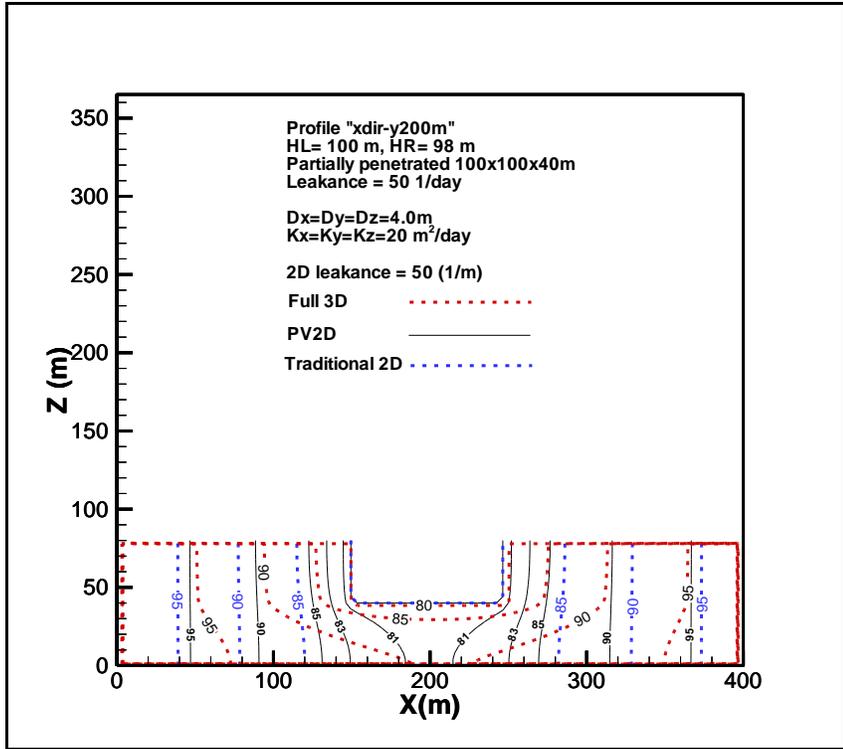


FIGURE 11.3.3-5
 Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the 'xdir-y200m' profile with the leakance value of 50/day

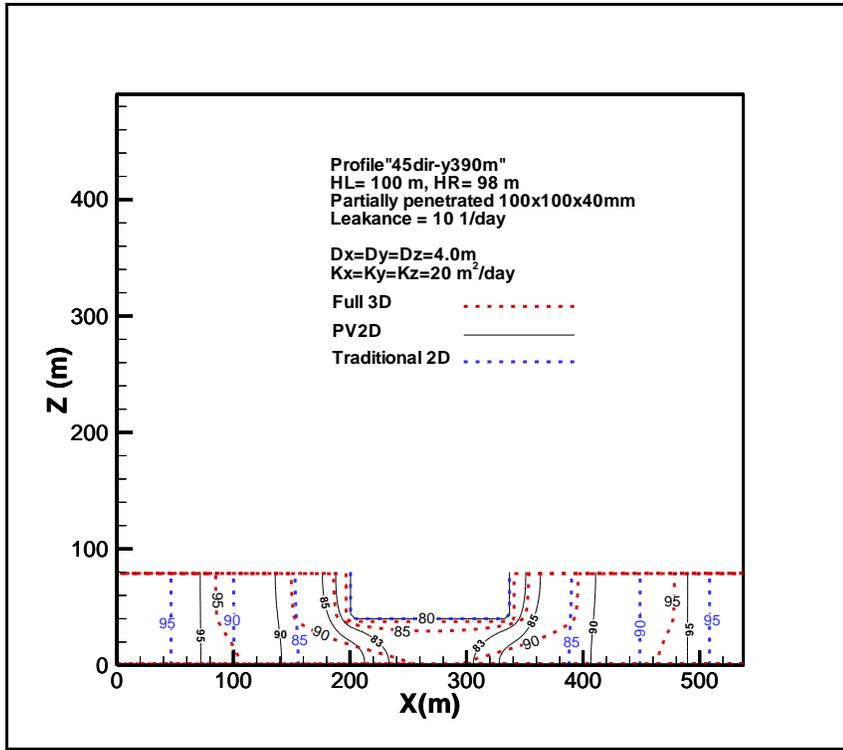


FIGURE 11.3.3-6
 Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the '45dir-y390m' profile with the leakance value of 10/day

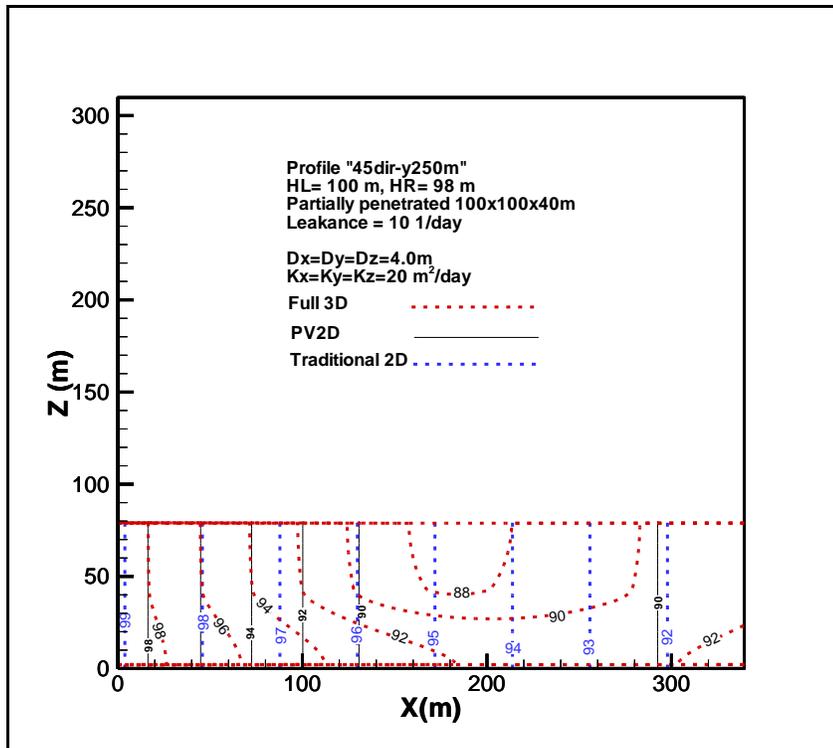


FIGURE 11.3.3-7
Comparison of the 3-D, PV2-D, and traditional profile 2-D model results for the '45dir-y250m' profile with the leakance value of 10/day

11.3.4 STEP GEOMETRY AT AQUIFER BOTTOM

Figure 11.3.4-1 shows a conceptual representation of the modeled situation.

Figures 11.3.4-2, 11.3.4-3, and 11.3.4-4 compare the full 3-D layer-based 3-D, PV2-D, and traditional profile 2-D model solutions for the 'xdir-200m', '45dir-250m', and '45dir-390m' profiles, respectively.

The results for the PV2-D model are more similar to the full 3-D model than the traditional profile 2-D model. This occurs because the PV2-D model considers cross-flow data in its solution. The layer-based 3-D model shows large deviations (from the full 3-D model) near the bottom of modeling domain. The deviations in the PV2-D occur because the PV2-D model receives only average cross-flux data from the plane 2-D model and thus cannot completely resolve the vertical variations.

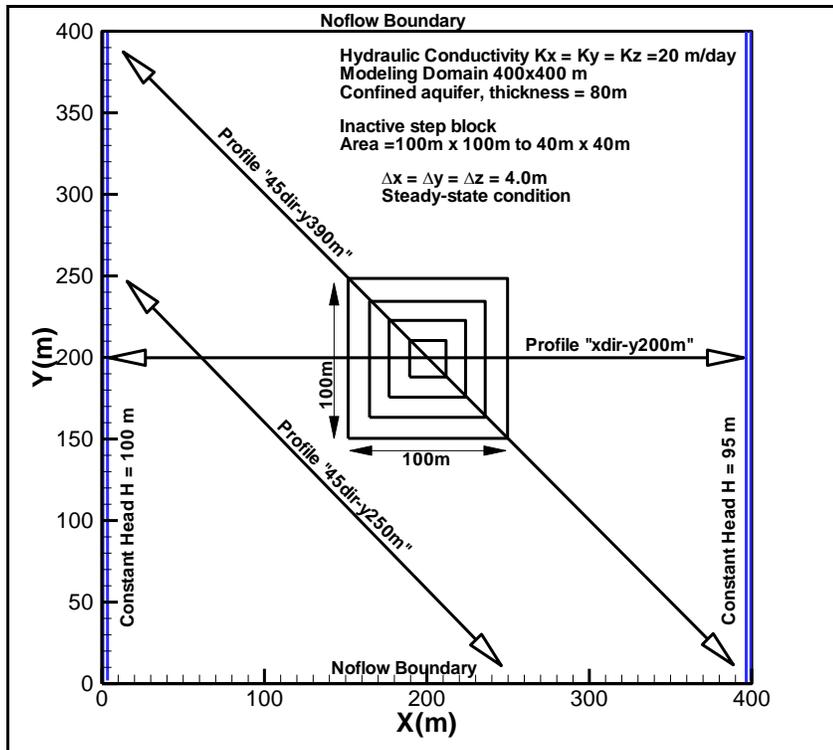


FIGURE 11.3.4-1
 Conceptual representation for the step geometry at aquifer bottom case

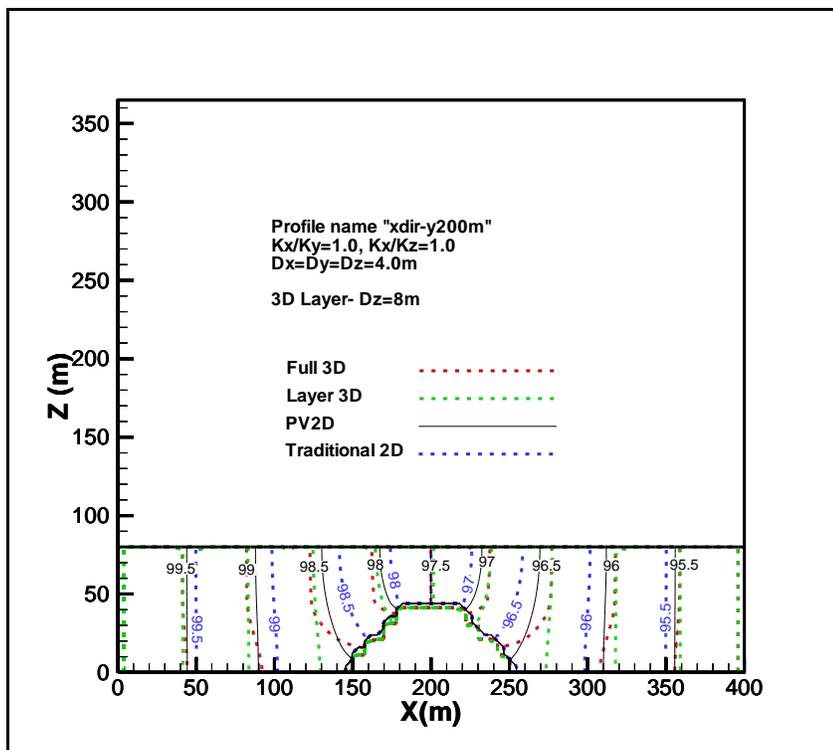


FIGURE 11.3.4-2
 Comparison of the full 3-D, layer-based 3-D, PV2-D, and traditional profile 2-D model results for the 'xdir-y200m' profile

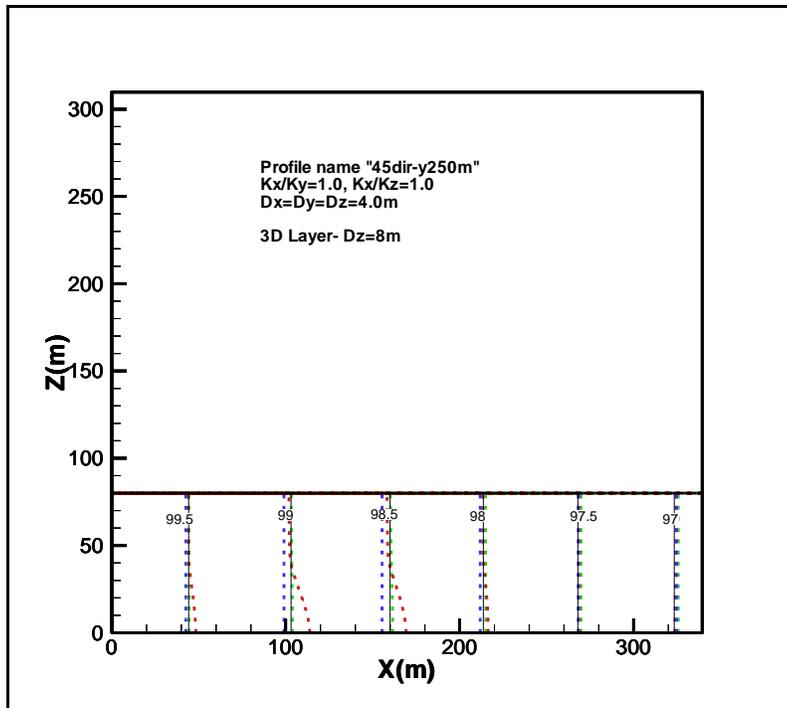


FIGURE 11.3.4-3
 Comparison of the full 3-D, layer-based 3-D, PV2-D, and traditional profile 2-D model results for the '45dir-y250m' profile

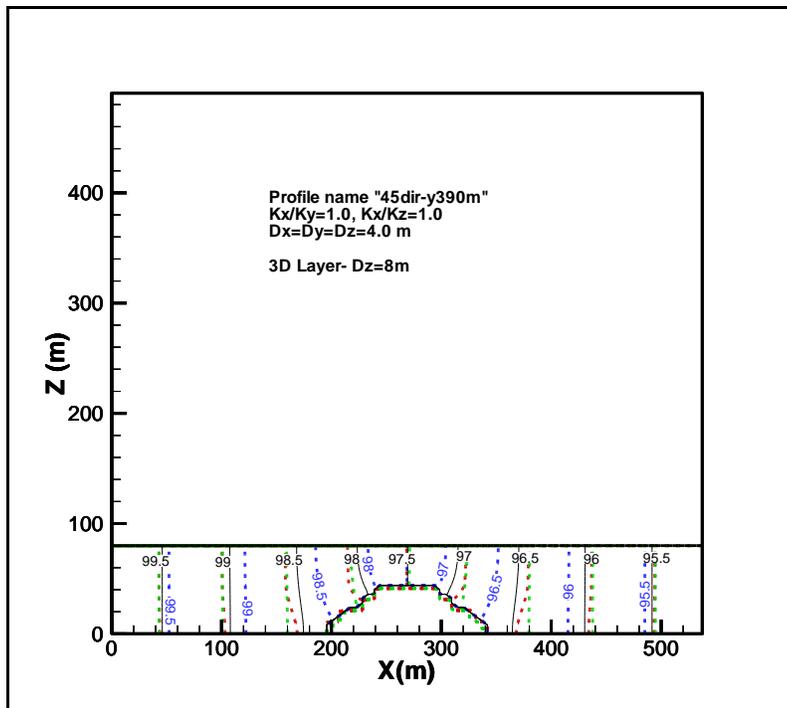


FIGURE 11.3.4-4
 Comparison of the full 3-D, layer-based 3-D, PV2-D, and traditional profile 2-D model results for the '45-y390m' profile

11.3.5 GENERAL STATEMENTS

The previous examples show the important role of the cross flux terms in profile model formulation. The traditional profile 2-D model, which does not consider the cross flux term, will either over or under estimate the profile head values.

The PV2-D model in a flow field with no vertical variations will perfectly reproduce the full 3-D model results. This is important when modeling large areas where a great number of nodes are needed to fully represent the area in the model. The PV2-D model may be employed in lieu of the 3-D model to significantly reduce the amount of computational power needed to obtain an accurate solution.

In a constant head area, the PV2-D solution is most applicable for a constant head feature that is fully penetrating (the solution is equivalent to the 3-D model solution). Significant deviations develop below the constant head feature bottom (when it is not fully penetrating), therefore it can be said that the PV2-D model increases in accuracy as the depth of the constant head feature increases. Further study has shown that a constant head feature depth larger than half of aquifer depth is acceptable for PV2-D model by adjusting the lake leakance to approach the full 3-D model.

The drawback of the PV2-D model is that its accuracy is dependent on the cross flux terms from the plane 2-D model. It can be said that the PV2-D model under estimates the head distributions near complex flow pattern areas (i.e. the bottom of a partially penetrating lake or an area of significant aquifer geometry change) because the cross flux terms are averaged over the depth of the aquifer in the plane 2-D model and therefore do not allow the PV2-D model to resolve any vertical variations.

REFERENCES

- i GSLIB
- ii GMS Brigham Young University 1998 Groundwater Modeling System Reference Manual
- iii Wang and Anderson – Introduction to GW modeling
- iv Theis 1935
- v Freeze and Cherry
- vi USGS
- vii Ferris, et al 1962
- viii Same as VII
- ix Ferris 1951
- x Beastle 1969