INVERSE ANALYSIS FOR DETECTION OF HIGH PERMEABILITY ZONES IN POROUS MEDIA

by

Seong Jun Lee

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Civil and Environmental Engineering

The University of Utah

May 2014

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of Seong Jun Lee

has been approved by the following supervisory committee members:

Brian McPherson	, Chair	Oct. 29, 2013 Date Approved
Steven Burian	, Member	Oct. 29, 2013 Date Approved
Andy Hong	, Member	Oct. 29, 2013 Date Approved
Fernando Guevara Vasquez	, Member	Oct. 29, 2013 Date Approved
Lisa Stright	, Member	Oct. 29, 2013 Date Approved

and by Michael Barber , Chair/Dean of

the Department/College/School of Civil and Environmental Engineering

and by David B. Kieda, Dean of The Graduate School.

ABSTRACT

The objective of this dissertation is to estimate possible leakage pathways such as abandoned wells and fault zones in the deep subsurface for CO_2 storage using inverse analysis. Leakage pathways through a cap rock may cause CO_2 to migrate into the layers above cap rock. An inverse analysis using iTOUGH2 was applied to estimate possible leakage pathways using pressure anomalies in the overlying formation induced by brine and/or CO_2 leaks. Prior to applying inverse analysis, sensitivity analysis and forward modeling were conducted. In addition, an inverse model was developed for single-phase flow and it was applied to the leakage pathway estimation in a brine/CO_2 system.

Migration of brine/CO_2 through the leakage pathway was simulated in the generic homogeneous and heterogeneous domains. The increased pressure gradient due to CO_2 injection continuously induced brine leaks through the leakage pathway. Capillary pressure was induced by the migration of CO_2 along the leakage pathway saturated by brine. Pressure anomalies due to capillary pressures were propagated to the entire overlying formation. The sensitivity analysis was focused on how the hydrogeological properties affect the pressure signals at monitoring wells.

Parameter estimation using the iTOUGH2 model was applied to detect locations of leakage pathways in homogeneous and heterogeneous model domains. For homogeneous models, the parameterization of uncertain permeability in an overlying formation could improve location estimation accuracy. Residual analysis illustrated that

pressure anomalies in the overlying formation induced by leaks are critical information for the leakage pathway estimation. For heterogeneous models, the calibration of renormalized permeability values could reduce systematic modeling errors and should improve the leakage pathway location estimation accuracy. The weighting factors significantly influenced the accuracy of the leakage pathway estimation.

The developed inverse model was applied to estimate the leakage pathway in a brine/CO_2 system using pressure anomalies induced by only brine leaks. To estimate a possible leakage pathway, the developed inverse model calibrated each integrated parameter (of both cross-sectional area and vertical hydraulic conductivity) of initial guesses of the leakage pathway. This application can provide warning before the CO_2 leaks, and will be useful in mitigating the risk of CO_2 leaks.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

NOMENCLATURE

K_x, K_y, K_z	X, Y and Z-directional hydraulic conductivity, respectively
h	Hydraulic head
W	Sources and/or sinks of water
L	Leakage term
Ss	Specific storage
t	Time
Q_x	X-directional discharge
Q_y	Y-directional discharge
Q_z	Z-directional discharge
f	Notation of water flows into node (i, j, k) from the upstream node
b	Notation of water flows out from node (i, j, k) to downstream node
n	Certain time step
I	Leakage column (or pathway) number
$Aleak_{i,j,k}$	Leakage area at node (i, j, k)
$Dzbleak(I)$	Length of I-th leakage pathway
$Kzbleak_{i,j,k}$	Z-directional hydraulic conductivity of leakage pathway at node (i, j, k)

zls(I)	Z-coordinate at an injection aquifer of I-th leakage
zlsu(I)	Z-coordinate at an upper aquifer of I-th leakage
k_{leak}	Vertical permeability of leakage pathway
k_{xyz}	Average value of x, y, and z-directional permeability
y	Independent variable of parameter x_i
y^0	Expectation or mean of y
x_i	Input parameters for $\mathrm{i} = 1, 2, .., n$
x_i^0	Expectation or mean of x_i ($x_i^0 = E(x_i)$)
$\left[\frac{\partial y}{\partial x_i}\right]_{x^0}$	Sensitivity coefficient evaluated at $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$
Cov $[x_i, x_j]$	Covariance between the x_i and x_j

ACKNOWLEDGEMENTS

This study was supported by the Department of Energy (DOE), a United States Government agency, under contract number DE-FC26-05NT42591.

CHAPTER 1

INTRODUCTION

1.1 Objectives and Background

The objective of this dissertation is to use mathematical inverse analysis to identify possible locations of abandoned wells, or other possible leakage zones, in subsurface reservoirs. This research is related to separation of CO_2, a primary greenhouse gas from coal fired power plants and other point sources, and storage of that CO_2 in geological formations. Fig. 1.1 illustrates a simple schematic of CO_2 storage system in geological formations with the brine/CO_2 leakage pathways.

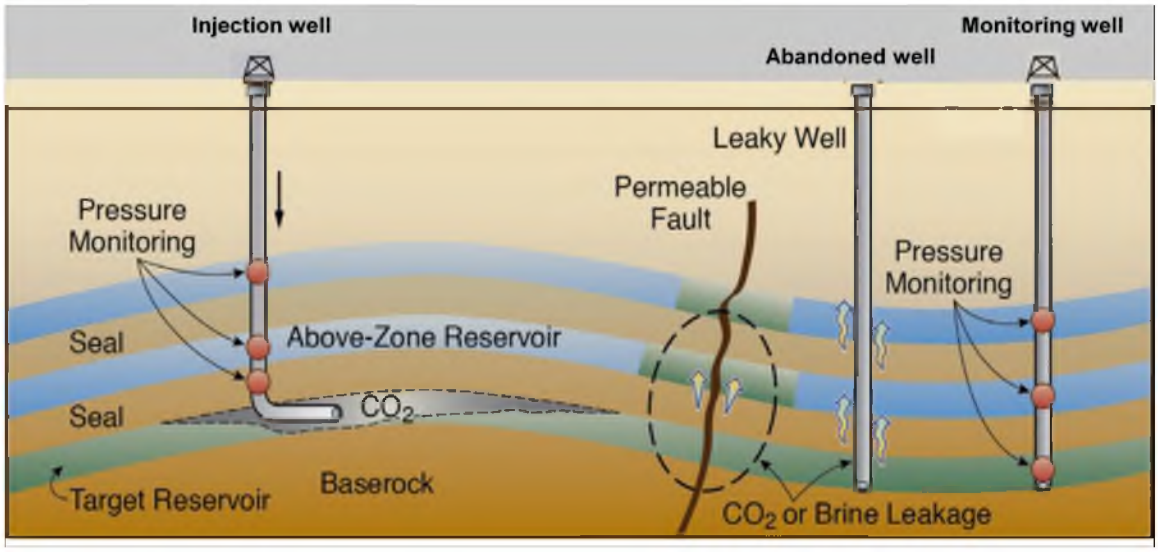


Fig. 1.1 A simple schematic of brine/CO_2 system in geological formations. Modified from Jung et al. (2012a).

This research focuses on storage of CO_2 in oil and gas fields as well as saline formations. However, storage of CO_2 in deep geological formations has risks, and perhaps the most important risk is leakage of CO_2. For a reservoir to store CO_2 ideally it will exhibit high porosity, high permeability and be capped by a low-permeability seal layer (or caprock above the reservoir). The existence of pathways that will release CO_2 from the reservoir and through the seal rock layer may allow CO_2 to escape into the atmosphere or to migrate into adjacent aquifers. Detection of these pathways is a very significant objective.

CO_2 may leak through fractures, faults, or abandoned pre-existing wells (Metz et al., 2005). CO_2 leakage through abandoned pre-existing wells is identified as one of the most probable leakage pathways. More than 350,000 abandoned oil and gas wells have been drilled in Alberta, Canada (Gasda et al., 2004). The state of Texas is a major energy producer in the United States, and has more than 1,000,000 abandoned wells. Particularly, uncompleted or improperly plugged abandoned wells are most susceptible to leakage of buoyant fluids such as CO_2 (Metz et al., 2005).

Abandoned wells and other leakage pathways typically exhibit higher vertical hydraulic conductivity (or permeability) than the reservoir. Therefore, leakage pathways may cause anomalies of pressure that induce transient flow in reservoirs. Thus, this study especially focuses on pressure anomalies to estimate locations of abandoned wells and other potential leakage zones.

Before applying inverse methods to identify leakage pathways, more general numerical modeling was performed to evaluate flow patterns, and impacts of leakage zones on flow patterns in confined brine aquifers with anisotropic, heterogeneous and

isothermal conditions. To achieve these objectives, I developed new simulation codes (Chapter 5) and also used simulation codes developed by the Lawrence Berkeley National Laboratory (Chapter 2).

1.2 Literature Survey

In carbon capture, utilization and storage (CCUS), CO_2 leakage is the most significant risk, so studies about CO_2 leakage have been recently of great interest. In particular, a number of studies related to brine/CO_2 leakage detection in porous media have been performed, including applications of seismic, InSAR (Interferometric Synthetic Aperture Radar) data of surface deformation and numerical inverse modeling. As mentioned earlier, this study applies numerical inverse analysis as a risk assessment tool to detect potential CO_2 leakage in mature sedimentary basins. Therefore, the review of previous studies focused on numerical modeling associated with CO_2 leakage. In addition, some recent literature related to other techniques for leakage detection were investigated. Table 1.1 summarizes selected recent studies of fluid reservoir storage and potential leakage. Studies using numerical modeling to quantify leaks can be distinguished by forward vs. inverse modeling. In general, forward simulations can be performed to realize leakage behaviors and pressure perturbations using model parameters investigated from other techniques. On the other hand, inverse modeling is usually used for estimating those model parameters.

Several studies utilized forward numerical modeling for quantification of leaks in groundwater aquifers; including Cobb et al. (1982), Chen (1989), Christensen and Cooley (1999), and Singh (2009) (Table 1.1).

Table 1.1 Selected studies of potential leakage from subsurface storage reservoirs using numerical modeling and other techniques.

Reference	**Forward/ Inverse**	**Fluid**	**Laboratory/ Field/Generic**	**Comment**
Cobb et al. (1982); Chen (1989); Christensen and Cooley (1996); Singh (2009)	Forward	Pure water	Field/ Generic	Solute transport in Groundwater
Anderson and Woessner (1992)	Forward	Pure water	Generic	Analytical solution of leaks
Beckford et al. (2003); Chan-Hilton et al. (2004); Espinoza et al. (2005); Babbar and Minsker (2006); Ko and Lee (2008)	Inverse	Pure water	Field/ Generic	Solute transport and remediation problems
Pruess and Garcia (2002)	Forward	Brine/ CO_2	Generic	Brine/CO_2 leaks in 1-D, homogeneous domain
Gasda et al. (2004)			Field investigation	Investigation of abandoned well distribution
Pruess (2004)	Forward	Brine/CO_2	Generic	Brine/CO_2 leaks in 2-D, homogeneous domain
Doughty and Pruess (2004)	Forward	Brine/CO_2	Field	Frio formation, Texas
Altevogt and Celia (2004)	Forward	CO_2	Field	Natural CO_2 leaks Mammoth Mountain, California
Nordbotten et al. (2004)	Forward	Brine	Generic	Analytical solution, water only
Nordbotten et al. (2008)	Forward	Multiple fluids	Generic	Analytical solution of leaks, multiphase fluids
Zhou et al. (2009)	Forward	Brine	Generic	Semi-analytical solution of leaks, water only
Cihan et al. (2011)	Forward	Brine	Generic	Analytical solution of leaks, water only
Nogues et al. (2011)	Forward	Brine	Generic	Analytical solution of monitoring well locations, 2-D, homogeneous domain
Hou et al. (2012)	Forward	Brine/CO_2	Generic	Brine/CO_2 leaks through intact caprock, 3-D, heterogeneous domain
Gasda et al. (2011)	Inverse	CO_2	Field investigation	Investigation of permeability of injection wells and calibration with NLINFIT and SCEM-UA
Jung et al. (2012a)	Inverse	Brine/CO_2	Generic	Framework of early leakage detection using inverse analysis and InSAR
Jung et al. (2012b)	Inverse	Brine	Generic	Sensitivity analysis of permeability of caprock and leakage detection using inverse analysis including random and systematic errors
Carroll et al. (2009)	Forward	Brine/CO_2	Generic	CO_2 leakage detection in near surface dilute aquifers using water chemistry perturbations
Onuma and Ohkawa (2009)		CO_2	Field	Monitoring of ground displacement using InSAR at In Salah, Algeria
Krevor et al. (2010)		CO_2	Laboratory	Surface CO_2 leak detection by WS-CRDS using carbon isotopes of CO_2
Sun et al. (2013)	Forward	Brine	Generic	Identify reliability of PCM for detectability of CO_2 leakage

These studies emphasize resolving solute transport and pumping test data to quantify aquifer properties in leaky aquifers. On the other hand, Anderson and Woessner (1992) used Darcy's law for interpreting leakage migration in aquifers with the assumption that the hydraulic head in sources overlying leaky confined aquifers is invariant with time. That is, when leaky aquifers have leakage pathways connected with huge source aquifers or rivers, the effect of leaks can be simulated in those leaky aquifers. Anderson and Woessner (1992) demonstrated that Darcy's law is useful for leakage. If leakage rates can be simulated by evaluating changes of hydraulic head with time (transient flow) using Darcy's law, it follows that such analysis may be applied to leakage from CO_2 storage formations.

Application of inverse analysis has been performed largely in the fields of groundwater flow, solute transport and remediation (Beckford et al., 2003; Chan-Hilton et al., 2004; Espinoza et al., 2005; Babbar and Minsker, 2006; Ko and Lee, 2008). However, it is limited to inverse analysis for detection of leaks in groundwater aquifers. In CCUS and multiphase flow, many researchers have studied forward analysis to solve leakage problems. For example, Pruess and García (2002) modeled the effects of CO_2 discharge along a fault zone, including impacts of salinity on CO_2 migration. Pruess and García (2002) considered how pressure drop reduces fluid mobility, thus decreasing vertical CO_2 flow but increasing lateral migration of CO_2. Consequently, such lateral migration of CO_2 can raise the possibility of more CO_2 diffusion to the land surface. Pruess (2004) followed with analysis of CO_2 migration patterns due to high-permeability faults.

Doughty and Pruess (2004) investigated the effects of heterogeneity on CO_2 migration. They stochastically generated heterogeneity in a three-dimensional domain, and examined CO_2 migration in such domains. Their study described how buoyancy driven CO_2 moves through a preferential flow path with higher permeability. Furthermore, they evaluated and compared generic vs. Frio-like relative permeability curves.

Altevogt and Celia (2004) explored flux mechanisms (of CO_2 transport) in the vadose zone. The simulations were applied to a natural CO_2 leakage site, Mammoth Mountain, California. For CO_2 transport simulations, the mass fraction gradient for diffusive and slip fluxes induced less plume spreading than advection alone. As a result, density contrasts between air and CO_2 leads to higher CO_2 mass and CO_2 mass fractions in the vadose zone than if equivalent densities of components are employed.

Gasda et al. (2004) investigated potential CO_2 leakage pathways of abandoned wells. Nordbotten et al. (2004) studied perturbations of hydraulic heads induced by leakage rates through abandoned wells, in systems with two aquifers and one aquitard. This study used Darcy's law to characterize a leakage term in the governing equation defined by mass conservation. They verified the methodology through comparison with Avci's solution (1994). Nordbotten et al. (2008) studied CO_2 leakage in multiple geological layers. This study described a framework to solve for leakage of multiphase fluids, using a leakage term (or sink/source term) in the governing equation. Analytical solutions were compared with traditional numerical reservoir model results. They concluded that using a leakage term ultimately provides a robust, grid-free approximation to CO_2 and brine leakage.

Zhou et al. (2009) developed semi-analytical solutions to simulate induced pressure perturbations and vertical leakage rates in a system consisting of multiple aquifers. They used a one-dimensional radial flow equation for the aquifer, and a one-dimensional vertical flow equation for the aquitard. As a result, the leakage rates and volumes are stipulated by the hydrogeologic properties and thicknesses of both the aquifer and the aquitard, as well as pumping or injection duration. To confirm their results they examined relationships between leakage rate (and volume) and the radial extent of the aquifer and wellbore radius. Cihan et al. (2011) developed a methodology to solve pressure perturbations by leakage wells and associated groundwater injection/pumping. They tested their analytical solutions through comparison with the results of Nordbotten et al. (2004), as well as comparison to a high resolution numerical solution.

Nogues et al. (2011) studied limits and extents of monitoring wells to measure pressure anomalies induced by leakage wells. They investigated effective radial extents induced by pressure anomalies in a homogeneous, two-dimensional domain, using an assumption that the pressure anomalies of single-phase sufficiently represents that of two phase (brine/CO_2) flow. They suggested that this study can be useful to design strategies for monitoring systems, and may improve leak detection with unique quantitative design of monitoring wells.

Hou et al. (2012) quantified postinjection impacts of CO_2 leakage through heterogeneous caprock without specific leakage pathways. Their analysis focused on CO_2 migration and rates of leakage through caprock, as dictated by differences of the mean and standard deviation of heterogeneity in both caprock and reservoir. They concluded

that the factors with the most impact are both the caprock permeability and the caprock thickness.

Several CCUS researchers have applied inverse analysis to detect leakage pathways. However, it seems that fewer people are interested in application of inverse modeling for leakage estimation than those using forward modeling to evaluate leakage features.

Gasda et al. (2011) investigated the actual permeability of several wells, in the field, using what is called the Vertical Interference Test (VIT) to measure pressure due to fluid movement outside of the casing of wells. They evaluated a nonlinear regression method (NLINFIT) and a Shuffled complex evolution metropolis method (SCEM-UA), and compared results to wellbore permeability from VIT. In addition, permeability of shale and compressibility of both the wellbore and the shale were investigated. They concluded that the two estimation methods gave reliable results when two parameters, wellbore and shale permeability, are estimated, while estimating more than two parameters decreases the accuracy of estimation. This study is not directly associated with leakage detection using the inverse analysis, but it is important in the context of estimating permeability values of actual abandoned wells, one of the most important sources of CO_2 leaks.

Jung et al. (2012a) developed a framework for early leakage detection. The framework consists of inverse modeling with high-spatial-resolution surface deformation (InSAR) data. Leakage pathways are calibrated from inverse analysis using measured pressure data; pathway estimations by inversion are combined with surface deformation data to increase accuracy. The basic concept of early leakage detection is to detect brine

leakage before actual CO_2 leaks through the estimated (locations of) leakage pathways. The possibility of early leakage detection can depend on (1) high sensitivity of pressure anomalies at monitoring wells, (2) the distance of monitoring wells from injection wells and leakage wells, and (3) spatial anomalies in surface deformation due to leakage.

Jung et al. (2012b) also utilized inverse modeling for leakage detection. They focused on the simple single-phase and homogeneous problem. Pressure anomalies from monitoring wells (through overlying and storage reservoirs) were used for early leakage detection by inverse modeling. The analysis estimates absolute permeability of potential leakage pathways (or initial guesses of such) by matching calculated pressure data from forward modeling with observed pressure data from monitoring wells. The monitoring data may have random errors due to various sources and systematic errors due to drift in pressure gauges. Thus, these errors in observed pressure may reduce efficiency of leakage detection. In addition, uncertain values of caprock permeability may have a significant impact on parameter estimation. That is, the uncertainty of hydrogeololgical properties may also reduce effectiveness of leakage detection. Therefore, they employed a sensitivity analysis of caprock permeability to examine the impact of its uncertainty. The modeling approach of Jung et al. (2012b) consists of four steps. The first step is to develop a conceptual model and its properties. The second step consists of sensitivity analysis of pressure anomalies in the overlying aquifer in the homogeneous system, with/without a leakage well, with respect to three different values of aquitard permeability (10^{-19}, 10^{-18}, and 10^{-17} m^2). The third step consists of parameter estimation for detecting a leakage well location through an idealized monitoring scenario. The final step is application of inverse modeling to reduce impact of uncertainty (systematic error)

of aquitard permeability and the random and systematic errors of measured data. These authors concluded that such specific inverse modeling can improve the possibility of leakage detection. Fig. 1.2 summarizes the workflow of Jung et al. (2012b).

Other methods for risk assessment of CO_2 leaks include seismic monitoring or imaging, land surface deformation monitoring, electrical and electromagnetic techniques, CO_2 land surface flux monitoring, and soil gas sampling (Carroll et al., 2009; Onuma and Ohkawa, 2009; Krevor et al., 2010; Sun et al., 2013). Carroll et al. (2009) simulated perturbations in water chemistry induced by CO_2 leakage into near-surface, dilute aquifers. The anomalies of water chemistry can be an important indicator for the potential release of CO_2 to the atmosphere. Specifically, the change of pH in aquifers can be an effective proxy for detection of CO_2 leaks, and the best monitoring/sampling location is the bottom of a confining layer, near the top of the dilute aquifer in question.

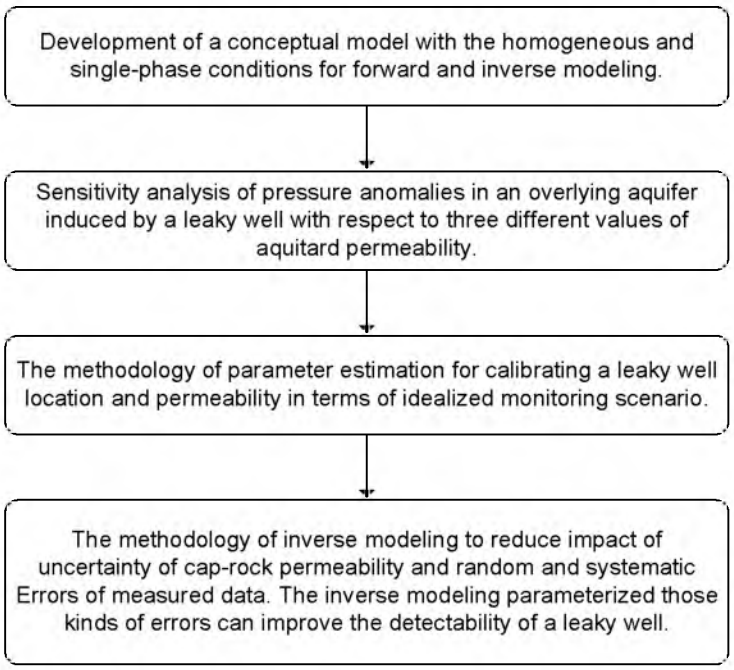


Fig. 1.2 The workflow of Jung et al. (2012b).

Onuma and Ohkawa (2009) analyzed InSAR data of surface deformation around actual CO_2 injection wells at In Salah, Algeria. The In Salah Gas Project is one of the largest CCUS projects, along with the Sleipner Project, Norway and the Weyburn Project, Canada. The InSAR is a remote sensing technique for mapping topography and monitoring of ground displacement, typically with accuracy of millimeters. The Onuma and Ohkawa (2009) InSAR analysis used time-dependent data from July 2003 to May 2008, and identified swelling rates of +14 mm/year around one injection well, +8 mm/year around another injection well, and a subsidence rate of -3 mm/year around producing wells. They suggested that this technique is powerful, efficient and low-cost, and may be an even better monitoring system when combined with geophysical methods.

Krevor et al. (2010) applied portable carbon isotope ratio analysis with wavelength scanned cavity ringdown spectroscopy (WS-CRDS) for near-surface detection of CO_2 leaks in an experimental facility with intentional CO_2 leakage. In general, the direct detection of CO_2 leaks into the atmosphere can be difficult because of large temporal and spatial variations from natural biological processes. However, carbon isotopes of CO_2 can distinguish between natural biogenic CO_2 fluxes and petrogenic CO_2 in deep formations because of distinct isotopic signatures. Consequently, they suggest that WS-CRDS can rapidly detect leakage locations and identify the isotopic composition of the source CO_2 flux.

Sun et al. (2013) applied a stochastic response surface method, PCM (probabilistic collocation method) for assessing leakage identification/location in a two-dimensional domain with single-phase and heterogeneous conditions. The PCM was used to assess the impact of heterogeneity on detectability of pressure anomalies in overlying

formations including uncertainty quantification (UQ). The PCM requires a smaller computational expense than full-scale Monte Carlo simulation because it uses points that are orthogonal with the assumed probability distributions. They compared PCM results to Monte Carlo results. They concluded that detectability depends on (1) degree of uncertainty of hydraulic conductivity and (2) the location of the monitoring well.

1.3 Conceptual Framework and Model

This study focuses on feasibility of leakage detection in CCUS, using inverse analysis of multiple formations. Multiphase flow (brine/CO_2) was considered and I employed the iTOUGH2 simulator developed by Lawrence Berkeley National Laboratory (Finsterle, 2007a). If a formation exhibits leakage, brine or CO_2 is able to move into the confined aquifer from the source reservoir as well as move out into adjacent confined aquifers, i.e., a leakage pathway can induce pressure anomalies in adjacent aquifers. Fluid pressure can be variously distributed and propagated in adjacent formations, depending on leakage locations and rates. But, specific pressure anomalies induced from leakage can provide information about leakage locations and rates. The inverse method employed here estimates leakage locations and rates by calculating the discrepancy between the calculated and observed pressure data at monitoring wells. Calculated pressures are obtained through forward simulation (TOUGH2), parameterized with vertical permeability values of randomly-selected initial guesses of leakage pathways. The forward simulation is repeated with updated parameter values, and then when discrepancies are minimized, the resulting set of parameter values is deemed the best estimation.

For successful inversion, the magnitude of pressure anomalies needs to be sufficient when brine and CO_2 migrate into the adjacent aquifers, and the randomly-selected parameter values should affect pressures at monitoring wells. Thus, sensitivity analysis is implemented to examine the effect of geological properties on magnitude of pressure, and to evaluate the sensitivities of pressure at monitoring wells to parameter values.

The hydrogeological parameters may include inherent errors. Uncertainties of parameters can influence calculated pressures, and thus may decrease accuracy of leakage pathway estimation (because the parameters-with errors-are assigned as known values in the inverse modeling). Therefore, the impact of parameter uncertainties is examined in the inverse analysis. Also, uncertainties are estimated through inverse modeling to improve accuracy of leakage pathway estimation.

1.4 Research Hypothesis

In this research, I identify applicability of a specific inverse method to detect CO_2 leakage via abandoned wells in deep geological storage formations. This study focuses on proving the following hypothesis: **Significant leakage zones can be detected using only observed hydraulic head or pressure data in multiple aquifers**. As mentioned in previous sections, if the CO_2 injection formations exhibit leakage pathways, leakage can flow to adjacent formations through those pathways, and associated pressure changes can be variously distributed and propagated depending on leakage locations and rates. Therefore, the goal of this research is to estimate leakage locations using inverse analysis of pressure perturbations induced by brine/CO_2.

1.5 References

Altevogt, A. S., and Celia, M. A. (2004). "Numerical modeling of carbon dioxide in unsaturated soils due to deep subsurface leakage." *Water Resources Research*, 40(3).

Anderson, M. P., and Woessner, W. W. (1992). *Applied groundwater modeling: simulation of flow and advective transport*, Elsevier Science, Oxford, UK.

Avci, C. B. (1994). "Evaluation of flow leakage through abandoned wells and boreholes." *Water Resources Research*, 30(9), 2565-2578.

Babbar, M., and Minsker, B. (2006). "Groundwater remediation design using multiscale genetic algorithms." *Journal of Water Resources Planning and Management*, 132(5), 341-350.

Beckford, O., Hilton, A. C., and Liu, X. "Development of an enhanced multi-objective robust genetic algorithm for groundwater remediation design under uncertainty." *Proc., Proc. Am. Soc. Civ. Eng*, 98.

Bennett, G. D. (1976). *Introduction to groundwater hydraulics*, U.S. Geological Survey, Denver, CO.

Carroll, S., Hao, Y., and Aines, R. (2009). "Transport and detection of carbon dioxide in dilute aquifers." *Energy Procedia*, 1(1), 2111-2118.

Chan-Hilton, A. B., Iyer, S. K., Magar, V., and Kelley, M. "Optimization of natural attenuation with active remediation under uncertainty." *Proc., Seventh International In Situ and On-Site Bioremediation Symposium, Orlando, Florida, USA, 2-5 June 2003. Part H. Natural Attenuation, Long-Term Monitoring, and Site Closure*, Battelle Press, Columbus, OH.

Chen, C. S. (1989). "Solutions approximating solute transport in a leaky aquifer receiving wastewater injection." *Water Resources Research*, 25(1), 61-72.

Christensen, S., and Cooley, R. (1999). "Evaluation of confidence intervals for a steady-state leaky aquifer model." *Advances in Water Resources*, 22(8), 807-817.

Cihan, A., Zhou, Q., and Birkholzer, J. T. (2011). "Analytical solutions for pressure perturbation and fluid leakage through aquitards and wells in multilayered-aquifer systems." *Water Resources Research*, 47(10), W10504.

Cobb, P., McElwee, C., and Butt, M. (1982). "Analysis of leaky aquifer pumping test data: An automated numerical solution using sensitivity analysis." *Groundwater*, 20(3), 325-333.

Doughty, C., and Pruess, K. (2004). "Modeling supercritical carbon dioxide injection in heterogeneous porous media." *Vadose Zone Journal*, 3(3), 837-847.

Espinoza, F. P., Minsker, B. S., and Goldberg, D. E. (2005). "Adaptive hybrid genetic algorithm for groundwater remediation design." *Journal of Water Resources Planning and Management*, 131(1), 14-24.

Finsterle, S. (2004). "Multiphase inverse modeling." *Vadose Zone Journal*, 3(3), 747-762.

Finsterle, S. (2007a). "iTOUGH2 user's guide." *Report LBNL-40040*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007b). "iTOUGH2 command reference." *Report LBNL-40041*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007c). "iTOUGH2 sample problems." *Report LBNL-40042*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2010). "iTOUGH2 V3. 2, verification and validation report." *Report LBNL-42002*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S., Moridis, G., and Pruess, K. (1994). "A tough2 equation-of-state module for the simulation of two-phase flow of air, water, and a miscible gelling liquid." *Report LBL-36086*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Gasda, S. E., Bachu, S., and Celia, M. A. (2004). "Spatial characterization of the location of potentially leaky wells penetrating a deep saline aquifer in a mature sedimentary basin." *Environmental Geology*, 46(6-7), 707-720.

Gasda, S. E., Wang, J. Z., and Celia, M. A. (2011). "Analysis of in-situ wellbore integrity data for existing wells with long-term exposure to CO2." *Energy Procedia*, 4, 5406-5413.

Hou, Z., Murray, J. C., and Rockhold, L. M. (2012). "CO2 migration in intact caprock and leakage risk in three-dimensional heterogeneous formations." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittssburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012a). "Early detection of brine or CO2 leakage through high-permeability pathways using pressure-based monitoring data." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittsburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012b). "Impact of data uncertainty on identifying leakage pathways in CO2 geologic storage systems and estimating their hydrogeological properties by inverse modeling." *TOUGH Symposium 2012,* Lawrence Berkeley National Laboratory, Berkeley, CA.

Ko, N.-Y., and Lee, K.-K. (2008). "Reliability and remediation cost of optimal remediation design considering uncertainty in aquifer parameters." *Journal of Water Resources Planning and Management*, 134(5), 413-421.

Krevor, S., Perrin, J.-C., Esposito, A., Rella, C., and Benson, S. (2010). "Rapid detection and characterization of surface CO2 leakage through the real-time measurement of δ13 C signatures in CO2 flux from the ground." *International Journal of Greenhouse Gas Control*, 4(5), 811-815.

Liggett, J. A., and Chen, L.-C. (1994). "Inverse transient analysis in pipe networks." *Journal of Hydraulic Engineering*, 120(8), 934-955.

Metz, B., Davidson, O., de Coninck, H., Loos, M., and Meyer, L. (2005). *IPCC special report on carbon dioxide capture and storage*, Cambridge University Press, New York, NY.

Nogues, J. P., Nordbotten, J. M., and Celia, M. A. (2011). "Detecting leakage of brine or CO2 through abandoned wells in a geological sequestration operation using pressure monitoring wells." *Energy Procedia*, 4, 3620-3627.

Nordbotten, J. M., Celia, M. A., and Bachu, S. (2004). "Analytical solutions for leakage rates through abandoned wells." *Water Resources Research*, 40(4), W04204.

Nordbotten, J. M., Kavetski, D., Celia, M. A., and Bachu, S. (2008). "Model for CO2 leakage including multiple geological layers and multiple leaky wells." *Environmental Science & Technology*, 43(3), 743-749.

Onuma, T., and Ohkawa, S. (2009). "Detection of surface deformation related with CO2 injection by DInSAR at In Salah, Algeria." *Energy Procedia*, 1(1), 2177-2184.

Pruess, K. (1987). "TOUGH user's guide, nuclear regulatory commission report NUREG/CR-4645." *Report LBL-20700*, Lawrence Berkeley Laboratory, Berkeley, CA.

Pruess, K. (1991). "TOUGH2: A general-purpose numerical simulator for multiphase fluid and heat flow." *Report LBL-29400*, Lawrence Berkeley Laboratory, Berkeley, CA.

Pruess, K. (2004). "Numerical simulation of CO2 leakage from a geologic disposal reservoir, including transitions from super-to subcritical conditions, and boiling of liquid CO2." *Spe Journal*, 9(2), 237-248.

Pruess, K. (2005). "ECO2N: A TOUGH2 fluid property module for mixtures of water, NaCl, and CO2." *Report LBNL-57952*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Pruess, K., and García, J. (2002). "Multiphase flow dynamics during CO2 disposal into saline aquifers." *Environmental Geology*, 42(2-3), 282-295.

Pruess, K., Moridis, G., and Oldenburg, C. (1999). "TOUGH2 user's guide, version 2.0." *Report LBNL-43134*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Rao, S. S. (2009). *Engineering optimization: Theory and practice*, John Wiley & Sons, Hoboken, New Jersey.

Singh, S. K. (2009). "Simple method for quick estimation of leaky-aquifer parameters." *Journal of Irrigation and Drainage Engineering*, 136(2), 149-153.

Sun, A. Y., Zeidouni, M., Nicot, J.-P., Lu, Z., and Zhang, D. (2013). "Assessing leakage detectability at geologic CO2 sequestration sites using the probabilistic collocation method." *Advances in Water Resources*, 56, 49-60.

Zhou, Q., Birkholzer, J. T., and Tsang, C.-F. (2009). "A semi-analytical solution for large-scale injection-induced pressure perturbation and leakage in a laterally bounded aquifer–aquitard system." *Transport in Porous Media*, 78(1), 127-148.

CHAPTER 2

BASIC THEORY OF MULTIPHASE CO_2 FLOW AND DESCRIPTION OF TOUGH2/ITOUGH2 SIMULATORS

2.1 Multiphase CO_2 Flow

This section summarizes the governing equations and flow analysis of two fluid phases in a porous medium. The basic theory of multiphase flow is discussed by many, including Chen et al. (2006).

For derivation of these equations, multiphase flow of CO_2 and brine is assumed to be immiscible and the phases do not have mass transfer (e.g., no dissolution). Diffusive effects between two phases are ignored. The brine is assumed as wetting phase. The wetting phase (e.g., brine) and nonwetting phase (e.g., CO_2) are indicated by *w* and o, respectively. The governing equation is derived from mass conservation theory. The general mass conservation equation for single-phase flow expressed by pressure is

$$\frac{\partial \emptyset \rho}{\partial t} = \nabla \cdot \left(\frac{\rho}{\mu} k(\nabla p + \rho g \nabla z) \right) + q. \qquad (2.1)$$

The governing equation for single-phase flow can be extended to multiphase flow. The governing equations for immiscible flow within two phases are

$$\frac{\partial(\emptyset\rho_w S_w)}{\partial t} = \nabla\cdot\left(\frac{\rho_w}{\mu_w} k k_{rw}\left(\nabla p_w + \rho_w g \nabla z\right)\right) + q_w, \tag{2.2}$$

$$\frac{\partial(\emptyset\rho_o S_o)}{\partial t} = \nabla\cdot\left(\frac{\rho_o}{\mu_o} k k_{ro}\left(\nabla p_o + \rho_o g \nabla z\right)\right) + q_o. \tag{2.3}$$

Table 2.1 denotes the parameters of the governing equations. Additionally, the two fluids completely fill pore space, so the relation between saturations of two fluids is

$$S_w + S_o = 1. \tag{2.4}$$

The surface tension at the interface between the two fluids results in discontinuity of pressure. That is, the pressure difference occurs by the capillary pressure;

$$P_c = P_o - P_w. \tag{2.5}$$

The capillary pressure lowers the pressure in the wetting phase. This is a result from surface tension which exists at the interface between two immiscible fluids. The capillary pressure is a function of wetting phase saturation (S_w) based on empirical data. Fig. 2.1 presents a typical curve of capillary pressure. As shown in Fig. 2.1, capillary pressure depends on the direction of S_w change through drainage and imbibitions, including hysteresis.

Table 2.1 Parameters for the governing equations.

k	Absolute permeability	g	Gravity
k_r	Relative permeability	S	Saturation
p	Pressure	$\emptyset$	Porosity
p_c	Capillary pressure	μ	Viscosity
q	Sink/source	∇	Divergence operator
w	Wetting phase (brine)	ρ	Density
u	Darcy velocity	z	Depth
o	Nonwetting phase (CO_2)	t	Time

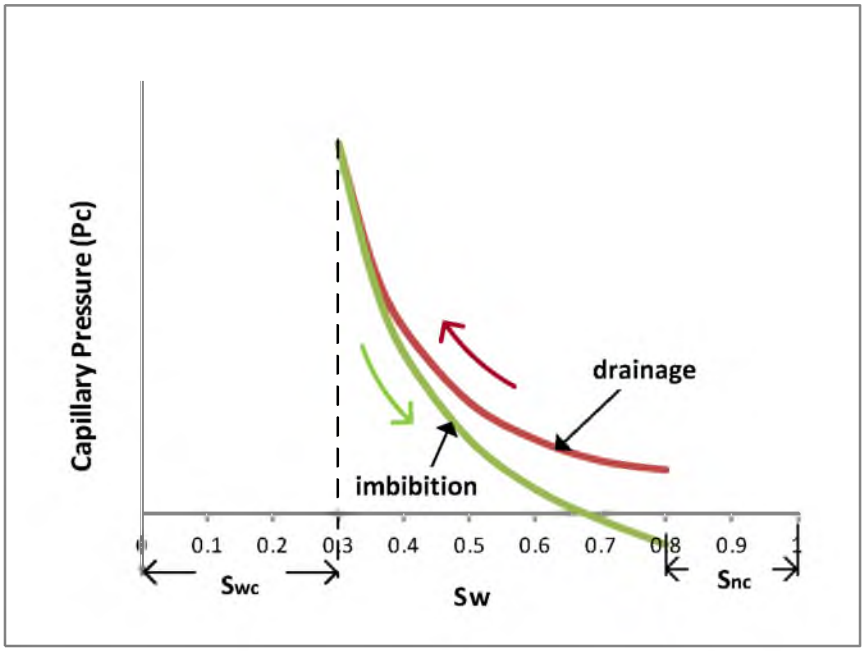


Fig. 2.1 Typical capillary pressure curve.

Referring to Fig. 2.1, S_{wc} is irreducible wetting phase saturation, or the wetting phase saturation value that cannot be reduced by migration of the nonwetting phase fluid. S_{nc} is referred to as irreducible nonwetting phase saturation, or the nonwetting phase saturation no longer displaceable by the wetting phase fluid. In general, capillary pressure is also related to surface tension, porosity, permeability and the contact angle between the rock surface and the wetting phase/nonwetting phase interface. In case that fully saturated brine is displaced by CO_2 during the CO_2 injection process, the change of capillary pressure will depend on the drainage curve.

The relative permeability (k_r) describes the reduction in the flux of one phase due to the interfering presence of the other phase. The values of k_r vary from 0 to 1. When the relative permeability of a certain phase is zero, it implies that the phase stops flowing because, from Darcy's law, volume flux becomes zero for this phase.

As for capillary pressure, the relative permeability is a function of wetting phase saturation (S_w), and the function of relative permeability must be empirically determined for the target formation for CO_2 storage. A generic curve of relative permeability in a porous media in which CO_2 displaces brine is shown in Fig. 2.2. The relative permeability for the nonwetting phase also depends on the direction of S_w change (through drainage and imbibitions) as manifested in capillary pressure. However, for the wetting phase, the relative permeability does not exhibit hysteresis. The process when the wetting phase is displaced by the nonwetting phase depends on the drainage curve, while the process when the nonwetting phase is displaced by the wetting phase depends on the imbibition curve. In an imbibition curve, the relative permeability becomes zero even if the saturation does not reach zero. That saturation is referred to as the residual saturation (S_{nc}).

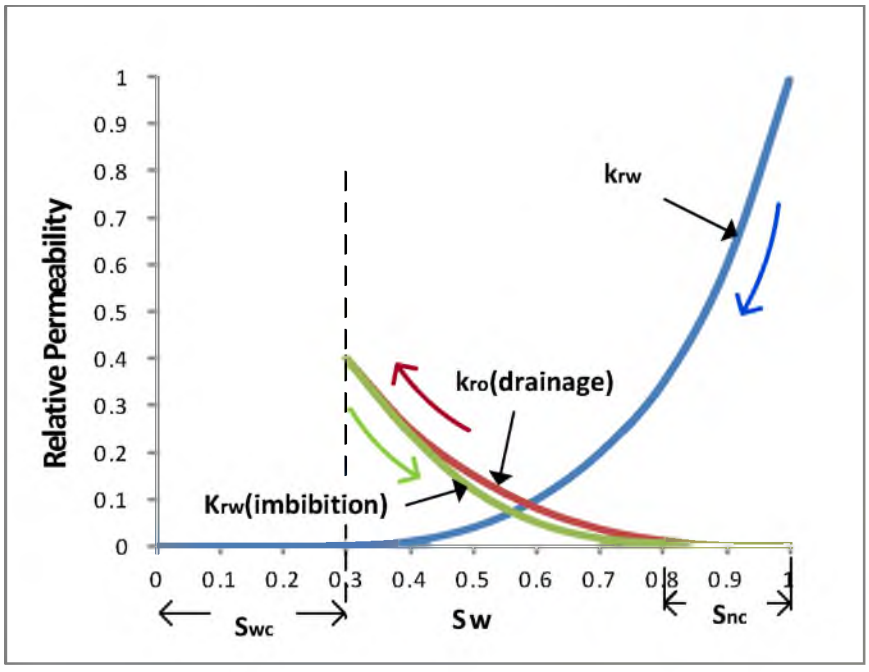


Fig. 2.2 Typical relative permeability curve.

The Darcy velocities (or volume fluxes) for each phase are expressed as follows:

$$u_w = -\frac{kk_{rw}}{\mu_w}\left(\nabla p_w + \rho_w g \nabla z\right), \tag{2.6}$$

$$u_o = -\frac{kk_{ro}}{\mu_o}\left(\nabla p_o + \rho_o g \nabla z\right). \tag{2.7}$$

Fluid properties such as PVT (pressure-volume-temperature) data for brine and CO_2, and equations of state (EOS) for compositional flow are needed to solve the governing equation. The EOS must calculate solubility, compressibility factor, density, viscosity, fugacity, enthalpy of CO_2 in gaseous and supercritical phases, and for mixtures or solutions of CO_2 in brine as functions of pressure and temperature.

2.2 TOUGH2 Simulator

The TOUGH2 program is a numerical simulator for nonisothermal flows of multicomponent, multiphase fluids in one-, two-, and three-dimensional porous and fractured media (Pruess et al., 1999). Fig. 2.3 depicts the general TOUGH2 structure.

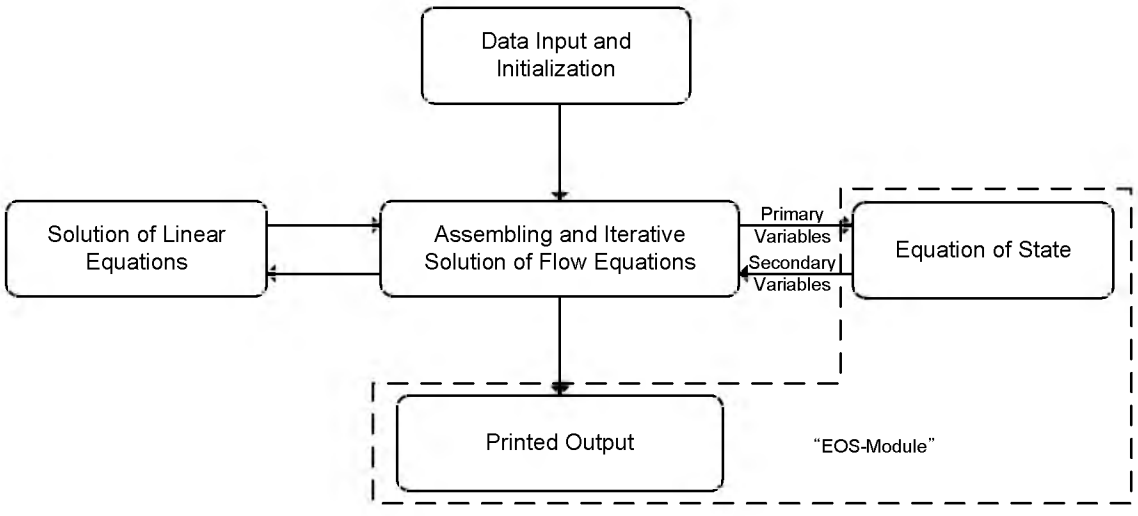


Fig. 2.3 Architecture of TOUGH2. Modified from Pruess et al. (1999).

TOUGH2 solves mass and energy balance equations using the "integral finite difference method" (IFDM) to describe fluid and heat flow in porous media. This method directly makes space discretization from the integral form of the conservation equations without converting the conservation equations into partial differential equations. Specific fluid properties such as fluid density, viscosity, enthalpy, etc. are provided by an equation of state (EOS) module, and the properties enter into the governing equations to calculate pressures for all phases and temperatures for all grid blocks in a formation. In the TOUGH2 simulator, the pressure and the temperature are described as the primary variables, and the fluid properties as secondary variables.

The next section deals with an EOS module of TOUGH2 to solve the properties of brine and CO_2 mixtures. The general description of discretization of the governing equations to apply a finite difference method is discussed in Chapter 5.

2.2.1 ECO2N Module

The TOUGH2 simulator provides many EOS modules to calculate fluid properties, tailored for specific subjects to be modeled. The ECO2N module used in this study was designed for fluid properties appropriate to geologic sequestration of CO_2 in saline aquifers (Pruess, 2005). The ECO2N simulates partitioning of H_2O and CO_2 using correlations developed by (Spycher and Pruess, 2005). The ECO2N equation of state represents the thermodynamics and thermophysical properties of H_2O - NaCl - CO_2 mixtures within 10 °C $\leq$ T $\leq$ 110 °C and P $\leq$ 600 bar. Readers are referred to Pruess (2005) and Spycher and Pruess (2005) for more details. This study assumed two-phase fluids, isothermal condition, and the primary variables were pressure and CO_2 saturation

for each grid block. For transient flow modeling, the time dependent primary variables are specified as unknowns to be calculated in each time step.

The ECO2N EOS can represent five phase options consisting of one or two phases as follows:

(1) Aqueous phase (brine with or without dissolved CO_2);

(2) Liquid phase (liquid CO_2 with or without dissolved water);

(3) Gaseous phase (gaseous CO_2 with or without dissolved water);

(4) Aqueous and liquid phases (brine and liquid CO_2); and

(5) Aqueous and gaseous phases (brine and gaseous CO_2).

The ECO2N EOS cannot describe two phase mixtures of liquid and gaseous CO_2. Thus, the simulator assumes no phase change between liquid and gaseous CO_2. In this dissertation, separate phase CO_2 is referred to as "gas." The mole fraction is used for phase-partitioning of water and CO_2 between aqueous and gas phases, because mole fraction dictates phase equilibrium relations.

The molar densities in the aqueous (water-rich phase) and gas phases (CO_2-rich phase) are

$$m_a = \sum_{i=1}^{NK} m_{ia}, \tag{2.8}$$

$$m_g = \sum_{i=1}^{NK} m_{ig}. \tag{2.9}$$

The variable NK describes the number of components in all phases, so NK is two (brine and CO_2). The mole fractions of two components in the aqueous phase are

$$x_1 (= x_{H2O}) = \frac{m_{1a}}{m_a}, \tag{2.10}$$

$$x_2 (= x_{CO2}) = \frac{m_{2a}}{m_a}. \tag{2.11}$$

The mole fractions of two components in the gas phase are

$$y_1 (= y_{H2O}) = \frac{m_{1g}}{m_g}, \tag{2.12}$$

$$y_2 (= y_{CO2}) = \frac{m_{2g}}{m_g}. \tag{2.13}$$

The algebraic constraints of mole fraction are

$$\sum_{i=1}^{2} x_i = \sum_{i=1}^{2} y_i = 1. \tag{2.14}$$

The TOUGH2 simulator with ECO2N describes phase compositions with respect to mass fractions. Equations and parameters are needed for conversion from mole fractions and molalities to mass fractions (Pruess, 2005). The total mass per kg of water, including m-molal in NaCl and n-molal in CO_2, is

$$M = 1000(gH_2O) + mM_{NaCl}(gNaCl) + nM_{CO2}(gCO_2). \tag{2.15}$$

Assuming that NaCl is fully dissociated, the total mass per kg of water is

$$m_T = 1000/M_{H2O} + 2m + n. \tag{2.16}$$

A relation between CO_2 mole fraction (x_2) in the aqueous phase and n-molal in CO_2 is

$$n = x_2 m_T, \tag{2.17}$$

so n-molal in CO_2 is

$$n = \frac{x_2(2m + 1000/M_{H2O})}{1 - x_2}. \tag{2.18}$$

The CO_2 mass fraction (X_2) in the aqueous phase can be calculated by dividing the CO_2 mass in n moles by total mass, so X_2 is

$$X_2 = \frac{nM_{CO2}}{1000+mM_{NaCl}+nM_{CO2}} . \quad (2.19)$$

The water mass fraction (Y_1) in the CO_2 rich phase is

$$Y_1 = \frac{y_1 \cdot M_{H2O}}{y_1 \cdot M_{H2O}+(1-y_1)M_{CO2}} . \quad (2.20)$$

In each fluid phase, the constraints of component mass fractions are

$$\sum_{i=1}^{2} X_i = \sum_{i=1}^{2} Y_i = 1 . \quad (2.21)$$

Thus, we can describe phase compositions using mass fractions. The thermophysical properties density, viscosity, and specific enthalpy of the fluid phases are functions of temperature, pressure, and composition. These properties must be calculated to simulate the flow of H_2O - NaCl - CO_2 mixtures. Table 2.2 denotes the parameters of the EOS.

Table 2.2 Parameters for ECO2N.

a	Aqueous phase	m	Molal in NaCl
g	Gas phase	n	Molal in CO_2
m_i	Molar density of component i in the phase	M	Total mass per kg of water with dissolved NaCl and CO_2
NK	Total number of component in the phase	m_T	Total moles per kg of water with dissolved CO_2
m_a	Total molar density in aqueous phase	M_{H2O}	Molecular weight of water
m_g	Total molar density in gas phase	M_{NaCl}	Molecular weight of NaCl
x_1	Water mole fraction in aqueous phase	M_{CO2}	Molecular weight of CO_2
x_2	CO_2 mole fraction in aqueous phase	X_1	Water mass fraction in aqueous phase
y_1	Water mole fraction in gas phase	X_2	CO_2 mass fraction in aqueous phase
y_2	CO_2 mole fraction in gas phase	Y_1	Water mass fraction in gas phase
		Y_2	CO_2 mass fraction in gas phase

The ECO2N module calculates water properties from the steam table equations of the International Formulation Committee (1967). The module also obtains CO_2 properties from tabular data in a "CO2TAB" file from correlations developed by Altunin (1975). In general, CO_2 solubility in the aqueous phase can be modeled from Henry's law (Pruess and García, 2002). However, the ECO2N models the CO_2 solubility from a methodology of mutual solubilities of water and CO_2 (Spycher and Pruess, 2005). The density of the aqueous phase with dissolved CO_2 is calculated by

$$\frac{1}{\rho_{aq}} = \frac{1\text{-}X_2}{\rho_b} + \frac{X_2}{\rho_{CO2}}. \tag{2.22}$$

Brine density (ρ_b) for a water-salt system is calculated from Battistelli et al. (1997). The partial density of dissolved CO_2 (ρ_{CO2}) is calculated using the molar volume (V_{ϕ}) of dissolved CO_2 at infinite dilution (Garcia, 2001), or

$$V_{\phi} = a+bT+cT^2+dT^3, \quad \text{and} \tag{2.23}$$

$$\rho_{CO2} = \frac{M_{CO2}}{V_{\phi}} \cdot 10^3 . \tag{2.24}$$

The density of the CO_2 gas phase is obtained by approximating the density of pure CO_2 from tabular data (in the CO2TAB file), neglecting dissolved water because the dissolved water amount is very small.

Brine viscosity is obtained from a correlation by Phillips et al. (1981). The CO_2 viscosity of the gas phase is also approximated from tabular data, neglecting dissolved water. This study focuses on isothermal conditions, so specific enthalpy is not described. A further detailed overview of the ECO2N is summarized in Pruess (2005). Table 2.3 denotes the parameters for density.

Table 2.3 Parameters for density.

ρ	Density	V_{ϕ}	Molar volume of dissolved CO_2 in units of cm^3 per gram-mole
aq	Aqueous phase	T	Temperature
ρ_b	Brine density	a	37.51
X_2	CO_2 mass fraction in aqueous phase	b	-9.585e-2
M_{CO2}	Molecular weight of CO_2 (=44.0)	c	8.740e-4
m_g	Total molar density in gas phase	d	-5.044e-7

2.3 iTOUGH2 Simulator

Solving the forward problem is intended to predict values of the dependent variables, like hydraulic head or pressure, that depend on given values of model parameters like hydraulic conductivity, injection rate, and so on. On the other hand, solving the inverse problem is intended to estimate the values of model parameters from given measured values of dependent variables. Therefore, inverse modeling is, effectively, parameter estimation by model calibration (Finsterle, 2007a). The parameter estimation function of iTOUGH2, the inverse simulator used in this analysis (Chapter 2), is applied to estimate CO_2 leakage locations via pressure anomalies induced by abandoned wells or geologic faults.

The sensitivity of measurements with respect to parameters can be related to stable solutions to the inverse problem. The inverse analysis can also yield several solutions (nonuniqueness) if more than one set of parameters satisfy specified criteria for

minimization in the process of optimization. Furthermore, the uncertainty of hydrogeological properties, systematic errors (errors in devices) and random errors (noises in measured data) can have an impact on the accuracy of the inversion (Finsterle, 2007a). The uniqueness in the inverse problem can be associated with the number of measured data in the system (Liggett and Chen, 1994). Modern monitoring devices like pressure gauges and hydrometers can provide continuous data with time from multiple points in a hydrodynamic system. Sufficient measured data of high quality are very important for reliable inverse modeling. In addition, the uncertainty of model parameters fixed as known values can influence the accuracy of inversion, so it is also important that the model parameters values should be characterized as exact as possible.

The iTOUGH2 simulator offers five optimization methods for minimization algorithm. The model supports three applications including parameter estimation, sensitivity analysis and uncertainty analysis. The main purpose of iTOUGH2 is estimation of model parameters by matching the calculated data from forward models to the measured data from the laboratory or field. The forward model used by iTOUGH2 is TOUGH2 (Pruess et al., 1999). Sensitivity analysis is possible with the calculated response of specific input parameters, and uncertainty analysis may be conducted to quantify impact of parameter uncertainties.

This section summarizes these three applications of iTOUGH2, and the five optimization methods. In particular, the procedures of parameter estimation for evaluating leakage pathways are described. The applicability of the inverse modeling to detect a leakage pathway is described in Chapter 4.

2.3.1 Sensitivity Analysis

An objective of sensitivity analysis is to determine how much parameters affect pressures at measurement posts. The sensitivity of measurements with respect to the parameters is closely related to whether the inverse analysis is successful (or stable). If a certain unknown parameter of inverse analysis does not affect the dependent variables of forward analysis such as hydraulic head or pressure, the inverse solution will not be identified (Liggett and Chen, 1994). For example, if leakage in the form of high permeability pathways, the main unknown parameters of a given inverse analysis, is so far away from monitoring wells that it cannot affect pressure distributions at those wells, leakage locations and rates cannot be identified through inverse simulation. Thus, the sensitivities of measured data with respect to the unknown parameters are important to generate satisfactory inverse solutions. In general, the sensitivity analysis is performed by examining sensitivity coefficients ($\partial P_i / \partial a_j$), where P_i : pressure at the i-th measurement point, and a_j : the j-th parameter value.

Fig. 2.4 depicts a simple example of a sensitivity analysis with respect to hydraulic conductivity at fifteen grid blocks in a modeling domain assigned as single-phase and isothermal. Fig. 2.5 denotes the modeling domain with a leakage pathway, consisting of overlying, confining and storage formations. The domain has a leakage pathway in the confining layer, and this pathway induces hydraulic head anomalies in the overlying formation. The overlying aquifer is assigned 15 m hydraulic head and the storage aquifer is assigned 20 m hydraulic head as initial conditions. Table 2.4 presents the hydrogeological properties of the domain, Table 2.5 describes the conditions of water injection, and Table 2.6 denotes leaky conditions.

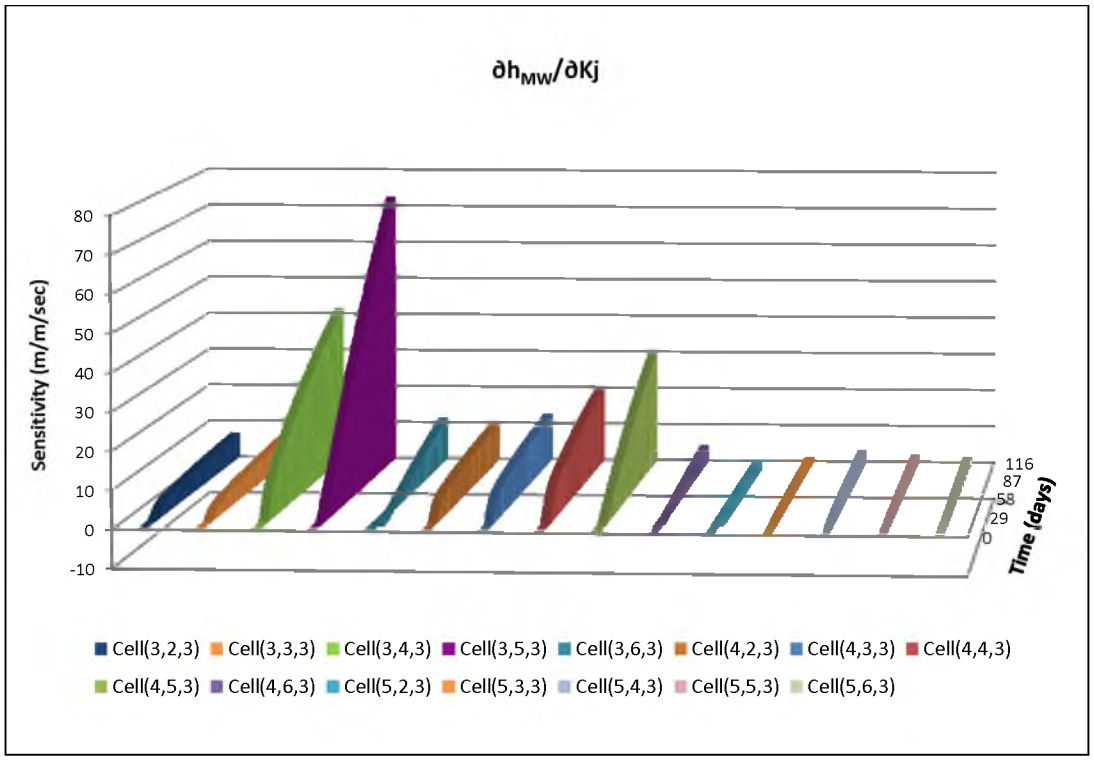


Fig. 2.4 An example of sensitivity analysis results corresponding to hydraulic conductivity at each cell.

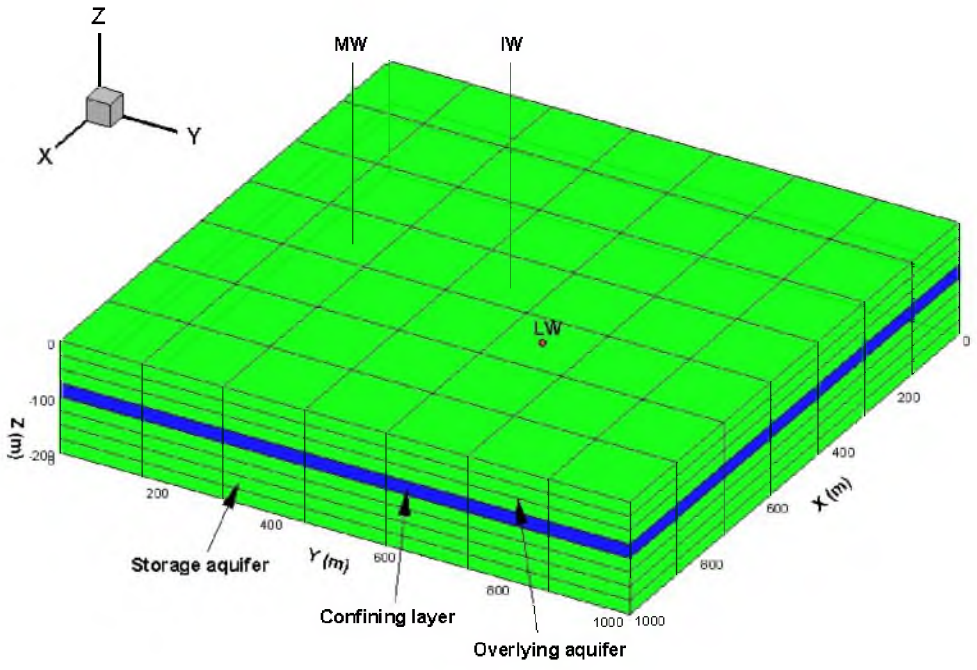


Fig. 2.5 Three-dimensional model domain for sensitivity analysis.

Table 2.4 Specification of the conceptual model for the sensitivity analysis.

Cubic size (m)	1,000 ×1,000× 200	**Total # of cells**		7 ×7× 8 (392)
Monitoring well (MW) location from origin	500 m, 200 m, -62.5 m	**Specific storage (m^{-1})**		0.0005
Simulation time (sec)	0 – 10,000,000	**Hydraulic conductivity (m/s)**	**Overlying and storage aquifers**	$Kx_{i,j,k} = Ky_{i,j,k} = Kz_{i,j,k} = 0.0001$
Tolerance	1e-5		**Confining layer**	$Kx_{i,j,k} = Ky_{i,j,k} = Kz_{i,j,k} = 0.0$

Table 2.5 Water injection conditions for the sensitivity analysis.

Injection well (IW)	**Water Injection Condition**	
	Injection time (sec)	**Injected water (m^3/s)**
(4, 4, 6)	0	0
	4,000	0.05
	10,000,000	0.05

Table 2.6 Leakage conditions for the sensitivity analysis.

Leakage well	**Leakage Condition**	
	Hydraulic conductivity (m/s)	**Cross sectional area (m^2)**
(3, 5, 3)	0.1	1
(3, 5, 5)		

Fig. 2.4 illustrates sensitivity of the hydraulic head at a monitoring well (MW) in the overlying aquifer to the change of hydraulic conductivity in the fifteen cells evaluated specifically. During each simulation, the hydraulic conductivity at one cell is changed from 10^{-2} to 10^{-8} m/s, while the hydraulic conductivities of other cells are kept unchanged at 10^{-4} m/s. From the result of the sensitivity analysis, hydraulic conductivity at cell (3, 5, 3) exerts the strongest influence on the hydraulic head at the MW. However, the hydraulic conductivities at cells (4, 6, 3), (5, 2, 3), (5, 3, 3), (5, 4, 3), (5, 5, 3) and (5, 6, 3) do not have direct influence on the hydraulic head at the MW. In other words, the hydraulic conductivities at these cells cannot be evaluated by inverse analysis using measured heads at the MW, if the hydraulic conductivity is an unknown parameter in the inverse analysis. This sensitivity analysis can be applied to examination of the hydrogeologic properties which have an impact on estimation of leakage locations. Other various applications of sensitivity analysis are described in Chapter 3.

2.3.2 Parameter Estimation

The parameter estimation function of iTOUGH2 solves the inverse problem to determine input parameters for a forward model, in this case, TOUGH2 (Pruess et al., 1999), based on a corresponding TOUGH2 output variable. Parameters are estimated by automatically matching calculated data with measured data of the system response. Thus, objective functions and minimization algorithms are needed to calculate residuals and to obtain a best set of solutions through minimizing those residuals, respectively. Fig. 2.6 denotes a flow chart of the parameter estimation approach employed by iTOUGH2.

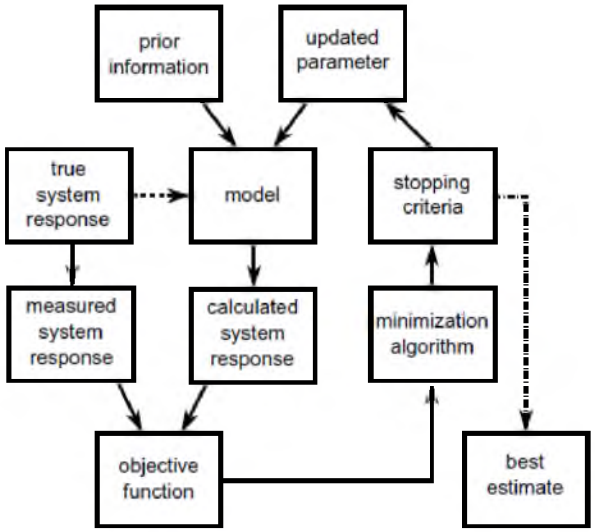


Fig. 2.6 Flow chart of parameter estimation. Modified from Finsterle (2007a).

Based on Fig. 2.6, the procedure of inverse modeling can consist of eight steps:

(1) Inverse analysis starts with the development of a conceptual model, including the model geometry, the initial and boundary conditions, the characterization of hydrogeological properties, and the discretization of model. In this step the model parameters are fixed as best known values. Those values may influence the uncertainty of modeling, so it is very important that the values are parameterized as the known values. This impact is described in more detail by Chapter 4.

(2) In this step, the parameters (vector $\mathbf{p}$) to be estimated are defined. The parameters can be chosen from the TOUGH2 input parameters and transformed by logarithm for the inverse modeling formulation to be more linear.

(3) The initial guesses must be assigned to each element of parameters in an input file of iTOUGH2. The initial guesses can be weighted by factors to scale parameters of different type, magnitude and/or accuracy. These weighting factors (diagonal elements of a matrix $\mathbf{C}_{zz}$) can also be used for measurements to be

scaled (refer to step 5).

(4) The values of each parameter set go into the forward simulator (TOUGH2), and then the TOUGH2 generates model output (vector $\mathbf{z(p)}$). The vector $\mathbf{z(p)}$is then compared to measurements (vector $\mathbf{z}^*$). As mentioned in section 2.3.1, the sensitivity of parameters to measurements is very important in terms of the reliability (or stability) of solutions in an inverse problem. Sufficient measured data of high quality are also critical for reliable parameter estimation.

(5) The objective function (S) compares model output with measured data at points in identical space and time, referred to as calibration points, and then calculates the sum of residuals ($\mathbf{r} = \mathbf{z}^* - \mathbf{z(p)}$), which is called the misfit. If the calculated and measured data do not correspond in terms of calibration points, the calculated data are interpolated to match measured data to calibration points. The objective function is usually some norm of the misfits. The objective function calculated using general least squares can be expressed as

$$\|\mathbf{r}\|_2 = \left(\sum_{i=1}^{m} |\mathbf{r}_i|^2 \right)^{1/2}, \tag{2.25}$$

where m is the number of calibration points. The weighted least-squares objective function used by iTOUGH2 is

$$S = \sum_{i=1}^{m} \frac{\mathbf{r}_i^2}{\sigma_{Z_i}^2}. \tag{2.26}$$

Here, $\sigma_{Z_i}$is prior error variance (weighting coefficient) for each observation. The measurement data can be appropriately weighted before calculating the sum of misfit, if the data need to be scaled and assessed based on measurement and

random errors. The weighting coefficient of a different scalar can be used for each measurement.

(6) An optimization method updates the unknown parameter values of a parameter set to reduce values of the objective function.

(7) Steps 4 through 6 are iterated until a minimum misfit from the objective function is obtained or the maximum iteration number, specified by the user, is exceeded.

(8) A set of parameters with the minimum or final misfit becomes the best estimation which is the solution of the inverse analysis. For that set of parameters, each value becomes the preferred parameter value. Table 2.7 presents the generalized inverse modeling procedure for leakage pathway estimation.

Table 2.7 Procedures of inverse modeling.

Step	Description
1	Development of a forward conceptual model
2	Selection of initial guesses of leakage pathway locations
3	Assignment of vertical permeabilities for each initial guess and lateral permeability for formations (for just overlying formation in case of homogeneous condition)
4	Calculation of TOUGH
5	Calculation of discrepancy between calculated and measured pressure at calibration points by objective function
6	Updating the parameter values to decrease discrepancy of objective function by an optimization method
7	Iterating from Steps 4 through 6 until minimum objective function values can be obtained or reaching iteration number specified by users
8	At best estimation, each parameter value of elements is estimated to high possible leaky location and formation permeability

In this study, the parameter estimation function of iTOUGH2 is applied to estimate leakage locations based on pressure anomalies due to leakage in multiple aquifers. The unknown parameters are the vertical permeability of initial guesses of locations of leakage wells (based on a priori information, for example) and the lateral heterogeneous or homogeneous permeability of overlying and storage formations. Previous iTOUGH2 modeling for leakage pathway estimation is discussed in Chapter 4.

2.3.3 Uncertainty Analysis

An objective of uncertainty analysis is to identify the main contributors that influence the outcome of a model. Model predictions inherently include uncertainty (Finsterle, 2007a). Among main sources of uncertainty are modeling errors. Modeling errors consist of errors in the input parameters and/or discretization errors (or truncation errors). Fundamentally, input parameter uncertainty is associated with heterogeneity of hydraulic conductivity, porosity and relative permeability, and other properties. In practice, additional investigation of those parameters (in the field or lab) should be performed to reduce errors in input parameters.

Numerical simulation will also have limited precision, primarily due to truncation errors. Truncation errors can be estimated from error propagation with respect to cell size, so we can estimate a maximum cell size to limit truncation errors (Finsterle, 2007a). For evaluating uncertainty, three methods usually can be used: Sensitivity analysis, the Monte Carlo method and first-order error analysis (or first-order second-moment; see Zheng et al. (2002)). iTOUGH2 provides two methods, the first-order second-moment method and the Monte Carlo method, to assess uncertainty propagation of output as a result of

parameter uncertainty. These three methods for the uncertainty analysis are described below.

2.3.3.1 Sensitivity Analysis

A general approach to sensitivity analysis was described in section 2.3.1. A sensitivity analysis can be used as a means of evaluating the input parameters that have the most effect on the outcome of the model. A sensitivity analysis can be performed based on the most important input parameters, such as absolute permeability, porosity and relative permeability. Each sensitivity coefficient of given input parameters serves as an indicator to quantify those parameters based on uncertainty propagation in calculated results. However, such sensitivity analysis for uncertainty cannot account for correlation of input parameters because each parameter is changed independently with other parameters, as discussed in section 2.3.1. In addition, such sensitivity analysis does not consider the probability distribution of the input parameters, so it cannot yield the quantitative probability distribution of the outcome. In fact, parameters are often correlated and exhibit a specific probability distribution. Nevertheless, sensitivity analysis can serve as a tool for approximate uncertainty analysis.

2.3.3.2 Monte Carlo Method

The Monte Carlo method is usually applicable to uncertainty analysis in terms of a stochastic approach. Each input parameter is defined as a random variable (X) by a probability density function (PDF) or by a cumulative distribution function (CDF). The PDF represents the probability of an uncertain input parameter on a particular value, i.e.,

$P(X = a)$. The CDF represents the probability of an uncertain input parameter within a range of a particular value, i.e., $P(X \leq a)$ or $P(X \geq a)$ (Zheng et al., 2002).

The Monte Carlo simulation starts with generating a set of random samples of each input parameter with respect to a PDF or CDF. Then the simulation model runs to obtain the model outcome with a specified combination of each parameter. The number of possible sets of samples of each input parameter is innumerable, and thus must be limited by the worker. The Monte Carlo method can demand a tremendous computational expense, and thus a method for sampling is needed to significantly reduce parameter space and calculation effort.

The iTOUGH2 simulator uses the technique of Latin hypercube sampling for reducing random sampling of independent variables. The PDF or CDF of the model output from the samples of each input parameter is estimated using a histogram or frequency plot. The mean, variance, median, and the probability of the model outcome exceeding or not exceeding a specific limit can be also calculated. A histogram or frequency plot is updated by the model output from the second set of random sampling of the input parameter. The procedure of Monte Carlo simulation can be repeated with more sets of sampling until satisfying a specified convergence criterion (e.g., until the difference in the results is not significant). Fig. 2.7 illustrates the procedure of the Monte Carlo simulation for uncertainty analysis.

To simulate a Monte Carlo analysis, a forward flow model like TOUGH2 can be combined with pre- and postprocessing codes to generate random samples of input parameters and to make histograms to illustrate the probability of model outcomes, respectively.

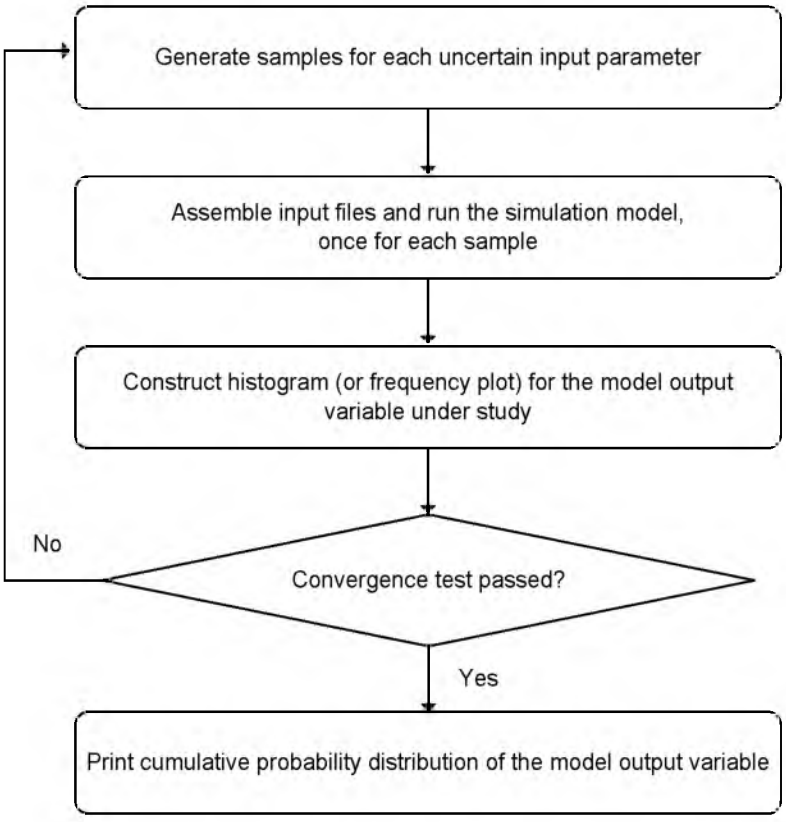


Fig. 2.7 Flowchart of uncertainty analysis using the Monte Carlo method. Modified from Zheng et al. (2002).

The main advantages of the Monte Carlo method are the simple concept and full applicability to uncertainty analysis, but the method has two primary disadvantages. The first problem is computational expense. As mentioned earlier, special sampling techniques of parameters are needed to reduce computational demand. The second problem in the Monte Carlo method is defining a PDF for each input parameter. Field data are rarely sufficient, so developing a PDF of input parameters inherently includes uncertainty, and this invariably propagates to uncertainty in the simulation results (Zheng et al., 2002). In general, hydraulic conductivity (or permeability) is processed as following a lognormal distribution, while porosity is treated as following a normal distribution (Benjamin, 1970).

2.3.3.3 First-Order Error Analysis

First-order error analysis is a direct method to quantify uncertainty propagation from input parameters to model output by linearization (Zheng et al., 2002). This method uses a Taylor series with n variables. The deviation between y and y^0 can be obtained from the Taylor series:

$$y - y^0 = \sum_{i=1}^{n} (x_i - x_i^0) \left[\frac{\partial y}{\partial x_i}\right]_{x^0} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_i^0)(x_j - x_j^0) \left[\frac{\partial^2 y}{\partial x_i \partial x_j}\right]_{x^0} + \cdots , \quad (2.27)$$

where $y = f(x_1, x_2, \cdots, x_n)$, y^0 is an expectation of y, x_i means input parameters for $i = 1, 2, .., n$, x_i^0 is equal to an expectation or mean of x_i ($x_i^0 = E(x_i)$), and $\left[\frac{\partial y}{\partial x_i}\right]_{x^0}$ expresses the derivatives (or sensitivity coefficients) evaluated at $x^0 = (x_1^0, x_2^0, \ldots, x_n^0)$.

Under the first-order approximation, the variance of the output is obtained as

$$\text{Var}\,[y] = E\,[(y - y^0)^2] \approx E\left[\left(\sum_{i=1}^{n} (x_i - x_i^0) \left[\frac{\partial y}{\partial x_i}\right]_{x^0}\right)^2\right]$$

$$= \sum_{i=1}^{n} \text{Var}\,[x_i] \left[\frac{\partial y}{\partial x_i}\right]_{x^0}^2 + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{Cov}\,[x_i, x_j] \left[\frac{\partial y}{\partial x_i}\right]_{x^0} \left[\frac{\partial y}{\partial x_j}\right]_{x^0} \quad (2.28)$$

where $\text{Cov}\,[x_i, x_j]$ is the covariance between x_i and x_j. From Equation (2.28), the uncertainty in the model output can be directly approximated by the variances of the input parameters (x_i), covariance between x_i and x_j, and each first order derivative ($\left[\frac{\partial y}{\partial x_i}\right]_{x^0}$) of the parameters. This first-order error analysis can be applicable to uncertainty propagation when the variation of model input parameters are sufficiently small (< 10~20%) (Zheng et al., 2002). Because this method neglects the higher-order terms in the Taylor series, if the variances of parameters are significant it cannot provide accurate

propagation of uncertainties in the input parameters. Such first order error analysis takes into account correlations among the parameters and reduces computational expense. For further information on the first order error analysis refer to Finsterle (2007a).

2.3.4 Optimization Methods in iTOUGH2

A minimization algorithm is needed to minimize objective functions while iteratively updating the parameters during the inverse modeling. The iTOUGH2 simulator provides five optimization methods as options:

(1) Gauss-Newton method;

(2) Levenberg-Marquardt method;

(3) Downhill Simplex method;

(4) Simulated Annealing method; and

(5) Grid Search method.

Each of these methods has advantages and disadvantages for inversion. The Levenberg-Marquardt method is the default method for minimization in iTOUGH2. This method is known to perform well for most inverse modeling applications for subsurface issues. The Levenberg-Marquardt method is a modified methodology of the Gauss-Newton method for nonlinear problems. The Downhill Simplex method does not involve derivatives in the objective function, unlike both the Levenberg-Marquardt method and the Gauss-Newton method. However, the Downhill Simplex method usually requires more simulation time. The Simulated Annealing method is suitable for local minimization problems, but it also requires lots of simulation time. The Grid Search method is used for simple problems with a relatively small number of variables (Finsterle,

2007a). In general, the descent techniques (Levenberg-Marquardt method and Gauss-Newton method) are more efficient than the direct search methods (Downhill Simplex method and Grid Search method) (Rao, 2009).

The parameter set (vector **p**) is updated at each iteration starting from an initial parameter set. The parameter set at the (k+1) iteration is

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta\mathbf{p}_k. \tag{2.29}$$

The objective function (S) requires that

$$S(\mathbf{p}_{k+1}) < S(\mathbf{p}_k). \tag{2.30}$$

Each optimization method proceeds with a different methodology for calculation of $\Delta\mathbf{p}_k$. That is, the objective of an optimization method is to calculate $\Delta\mathbf{p}_k$ to minimize the objective function.

2.3.4.1 Gradient, Jacobian and Hessian Matrix

First, Gradient, Jacobian and Hessian Matrix should be interpreted for the Gauss-Newton and Levenberg-Marquardt methods. The Gauss-Newton and Levenberg-Marquardt methods start with minimizing the objective function from a quadratic approximation of a Taylor series in n variables. From Equation (2.27), using the second-order approximation of the Taylor series, the objective function $S(\mathbf{p}_{k+1})$ can be expressed as

$$S(\mathbf{p}_{k+1}) = S(\mathbf{p}_k) + \sum_{i=1}^{n} \Delta p_i \left[\frac{\partial S}{\partial p_i}\right]_{\mathbf{p}_k} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \Delta p_i \Delta p_j \left[\frac{\partial^2 S}{\partial p_i \partial p_j}\right]_{\mathbf{p}_k} + \cdots$$

$$\approx \tau - \mathbf{d}\cdot(\mathbf{p}_{k+1} - \mathbf{p}_k) + \frac{1}{2}\,(\mathbf{p}_{k+1} - \mathbf{p}_k)\cdot\mathbf{H}\cdot(\mathbf{p}_{k+1} - \mathbf{p}_k), \tag{2.31}$$

in which $\mathbf{p}_{k+1}=(p_1, p_2, \cdots, p_n)$, $\tau = S(\mathbf{p}_k)$ is a constant value from the previous iteration k, $\mathbf{d}$ is a vector with negative gradient of the objective function evaluated at $\mathbf{p}_k$ (i.e., $\mathbf{d} = -\nabla S\big|_{\mathbf{p}_k}$), and $\mathbf{H}$ is the Hessian matrix of the objective function evaluated at $\mathbf{p}_k$ (i.e., $\mathbf{H} = \frac{\partial^2 S}{\partial p_i \partial p_j}\Big|_{\mathbf{p}_k}$).

The gradient of the objective function can be derived easily by differentiation of Equation (2.31) in terms of $\mathbf{p}_{k+1}$:

$$\nabla S = \mathbf{H} \cdot (\mathbf{p}_{k+1} - \mathbf{p}_k) - \mathbf{d} = \mathbf{H} \cdot \Delta\mathbf{p}_k - \mathbf{d} \quad . \tag{2.32}$$

If the left hand side of Equation (2.32) reaches an approximate minimum, we can obtain $\Delta\mathbf{p}_k$ by specifying the gradient of the objective function (∇S) to zero, and

$$\Delta\mathbf{p}_k = \mathbf{H}^{-1} \cdot \mathbf{d} \quad . \tag{2.33}$$

The first and second derivatives of the objective function with respect to each parameter should be calculated to obtain $\Delta\mathbf{p}_k$. From Press et al. (1992), the objective function of least squares (Equation (2.26)) differentiated by each parameter p_j is

$$\frac{\partial S}{\partial p_j} = -2 \sum_{i=1}^{m} \frac{r_i}{\sigma_{z_i}^2} \frac{\partial r_i}{\partial p_j} \quad , \quad j=1, \ldots, n \quad , \tag{2.34}$$

in which n is the number of parameters, m is the number of calibration points, $r_i = z_i^* - z_i$, z_i^* is the measurement at calibration point i, z_i is the model output at calibration point i and $\frac{\partial r_i}{\partial p_j}$ presents elements of the Jacobian matrix. An additional differentiation generates the elements of the Hessian matrix:

$$\frac{\partial^2 S}{\partial p_j \partial p_k} = 2 \sum_{i=1}^{m} \frac{1}{\sigma_{z_i}^2} \left[\frac{\partial z_i}{\partial p_j} \frac{\partial z_i}{\partial p_k} - (z_i^* - z_i) \frac{\partial^2 z_i}{\partial p_j \partial p_k} \right] \quad , \quad j \text{ and } k=1, \ldots, n \quad . \tag{2.35}$$

In Equation (2.35), $z_i^* - z_i$ can be almost zero near the solution. Besides, if the process maintains good matching, $z_i^* - z_i$ can exhibit either sign (positive or negative), so the combination of residual terms is likely to cancel one another during the summation from 1 to m. However, if the residuals are large, or if the model is highly nonlinear, the residuals will not cancel one another. In this case, the Hessian is not guaranteed to be positive definite to ensure the decrease of the objective function. By neglecting the second derivative term (by $z_i^* - z_i \approx 0$), the Hessian matrix can be simplified to evaluating the Jacobian matrix, $\mathbf{J}=\left[\frac{\partial z_i}{\partial p_j}\right]$. In Equation (2.34), the gradient is also evaluated by the Jacobian matrix. The Jacobian is an $(m \times n)$ matrix defined as

$$\mathbf{J}=\frac{\partial \mathbf{r}}{\partial \mathbf{p}}=\frac{\partial \mathbf{z}}{\partial \mathbf{p}}=\begin{bmatrix} \frac{\partial z_1}{\partial p_1} & \cdots & \frac{\partial z_1}{\partial p_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_m}{\partial p_1} & \cdots & \frac{\partial z_m}{\partial p_n} \end{bmatrix}. \tag{2.36}$$

The gradient vector $\mathbf{d}$ and the Hessian $\mathbf{H}$ can be written as matrix functions at iteration k:

$$\mathbf{d}_k = -\nabla S|_{\mathbf{p}_k} = 2\mathbf{J}_k^{\mathrm{T}}\mathbf{C}_{\mathrm{ZZ}}^{-1}\mathbf{r}_k\,, \tag{2.37}$$

$$\mathbf{H}_k = 2\mathbf{J}_k^{\mathrm{T}}\mathbf{C}_{\mathrm{ZZ}}^{-1}\mathbf{J}_k\,, \tag{2.38}$$

in which $\mathbf{d}_k$ is an $(n \times 1)$ matrix, $\mathbf{H}_k$ is an $(n \times n)$ matrix, and $\mathbf{C}_{\mathrm{ZZ}}$ is a covariance matrix that represents the weighting factors and the measurement error. $\mathbf{C}_{\mathrm{ZZ}}$ is an $(m \times m)$ diagonal matrix, or

$$\mathbf{C}_{\mathrm{ZZ}}=\begin{bmatrix} \sigma_{z_1}^2 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_{z_2}^2 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{z_m}^2 \end{bmatrix}. \tag{2.39}$$

With Equation (2.37) and (2.38), Equation (2.33) becomes

$$\Delta \mathbf{p}_k = (\mathbf{J}_k^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \mathbf{J}_k)^{-1} \mathbf{J}_k^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \mathbf{r}_k \ . \tag{2.40}$$

In iTOUGH2, the Jacobian matrix (2.36) is found using the Perturbation Method with a forward finite difference for the parameter estimation:

$$\mathbf{J}_{ij} = \frac{\partial z_i}{\partial p_j} = \frac{z_i\left(\mathbf{p}+\delta p_j\right) - z_i(\mathbf{p})}{\delta p_j} \qquad i = 1, .., m \text{ , and } j = 1, .., n \text{ ,} \tag{2.41}$$

in which m is the number of calibration points, n is the number of parameters and δp_j is a small perturbation, usually given as a fraction of the parameter value. The forward difference equation computes derivatives after evaluating the parameter vector ($\mathbf{p}$), that is, the forward difference has to be calculated at $\mathbf{p}$, $\mathbf{p}+\delta p_1$, $\mathbf{p}+\delta p_2$, …, $\mathbf{p}+\delta p_n$ to evaluate the Jacobian matrix. The forward calculation is likely to be inaccurate for two reasons:

(1) If δp is too small, the accuracy can be dropped in round off;

(2) If δp is too large, the approximation of difference may not be accurate.

Finsterle (2007a) describes the Perturbation Method in detail. For further information, refer to that reference.

In addition, both the inverse Hessian method and the Steepest Descent method can be easily interpreted. The first and second order derivatives of the objective function are defined by

$$\alpha_{jk} = \frac{1}{2}\frac{\partial^2 S}{\partial p_j \partial p_k} \ , \qquad \beta_j = -\frac{1}{2}\frac{\partial S}{\partial p_j} \ . \tag{2.42}$$

The symmetric matrix [α] is referred to as the "curvature matrix" because it is related to the curvature of the objective function (Press et al., 1992). With Equation (2.42), Equation (2.33) becomes

$$\sum_{k=1}^{n} \alpha_{jk} \Delta p_k = \beta_j \,. \tag{2.43}$$

The Δp_k term presents the correction of the k-th parameter of the current minimization step. Application of Equation (2.43) is called the "inverse Hessian method."

In Equation (2.33), assuming a linear small step down of gradient, Equation (2.33) becomes

$$\Delta \mathbf{p}_k = \lambda^* \cdot \mathbf{d} \,. \tag{2.44}$$

This algorithm is called the "steepest descent method," where λ^* is the step length in the steepest descent direction of gradient and **d** denotes the search direction for the minimum. In this method a parameter value iteratively moves along the steepest descent direction until the optimum value is found (Rao, 2009). The method to evaluate step length λ^* is described by Rao (2009) in detail.

2.3.4.2 Gauss-Newton Method

The Gauss-Newton method calculates $\Delta \mathbf{p}_k$ using Equation (2.40) to obtain a set of parameter values ($\Delta \mathbf{p}_{k+1}$) at iteration (k+1) to minimize the objective function. As mentioned in the previous section, Equation (2.40) neglects the second order term of the Hessian matrix, so this method is suitable for linear problems and for nonlinear problems near the solution, or if the initial guesses of parameters are close to the minimum. Otherwise, the value of the objective function may increase rather than decrease.

2.3.4.3 Levenberg-Marquardt Method

The Levenberg-Marquardt method is available for nonlinear models. The Levenberg-Marquardt method combines advantages of both the steepest descent method and the Gauss-Newton method (Bevington and Robinson, 1969; Press et al., 1992). The steepest descent method converges near the minimum when the parameter vector is away from the approximation. On the other hand, the Gauss-Newton method converges fast when the parameter vector is close to the approximation. The Levenberg-Marquardt method replaces the second order term of the Hessian matrix with an $(n \times n)$ diagonal matrix $\lambda_k \mathbf{D}_k$ or

$$\Delta \mathbf{p}_k = (\mathbf{J}_k^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \mathbf{J}_k + \lambda_k \mathbf{D}_k)^{-1} \mathbf{J}_k^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \mathbf{r}_k \,. \tag{2.45}$$

The multiplier λ_k is called "Levenberg parameter." The elements of $\mathbf{D}_k$ are

$$D_{jj} = (\mathbf{J}_k^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \mathbf{J}_k)_{jj} \;\;, \; j = 1, \ldots, n \,. \tag{2.46}$$

In Equation (2.45), the $(\mathbf{J}_k^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \mathbf{J}_k + \lambda_k \mathbf{D}_k)^{-1}$ term represents the step length along the search direction, $\mathbf{J}_k^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \mathbf{r}_k$. If λ_k is zero, $\Delta \mathbf{p}_k$ is evaluated identical to that of the Gauss-Newton method. On the other hand, if λ_k is large, $\Delta \mathbf{p}_k$ becomes parallel to the search direction of the steepest descent method and the step length decreases. The minimization process starts with a relatively large value of λ_k. Thus, with large λ_k, the Levenberg-Marquardt method evaluates the optimum parameter set using a small step length along the gradient of the objective function. If the objective function is improved (like Equation (2.30)), λ_k is decreased and the step length is increased. If the value of the objective function does not drop, λ_k is increased and that iteration is discarded. The minimization process is again applied by a shorter step length. The process is repeated until the convergence criteria are

satisfied or the objective function is reduced within criteria. The procedures of Levenberg-Marquardt method are summarized in Table 2.8.

2.3.4.4 Downhill Simplex Method

The Downhill Simplex method does not require the derivatives of the objective function. A "simplex" means the geometric figure formed by (n+1) points in n-dimensional space. When the distance between all points is the same, the simplex is referred to as regular.

Table 2.8 Procedures of Levenberg-Marquardt algorithm. Modified from Finsterle (2007a).

Step 1	Define initial values: - Iteration index $k = 0$ - Levenberg parameter (default: $\lambda_0 = 10^{-3}$) - Marquardt parameter (default: $\nu = 10$) - Initial parameter set: $\mathbf{p}_0$
Step 2	Run TOUGH2 with $\mathbf{p}_k$
Step 3	Calculate $\mathbf{r}(\mathbf{p}_k)$, $\mathbf{J}(\mathbf{p}_k)$, and $S(\mathbf{p}_k)$ with objective function ($S = \sum_{i=1}^{m} \frac{\mathbf{r}_i^2}{\sigma_{z_i}^2}$)
Step 4	Calculate $\Delta\mathbf{p}_k$; ($\Delta\mathbf{p}_k = (\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k + \lambda_k\mathbf{D}_k)^{-1}\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{r}_k$ with $D_{jj} = (\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k)_{jj}$)
Step 5	Update parameter set: $\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta\mathbf{p}_k$
Step 6	Run TOUGH2, and calculate $S(\mathbf{p}_{k+1})$
Step 7	If $S(\mathbf{p}_{k+1}) < S(\mathbf{p}_k)$, multiply λ by $1/\nu$ and go to Step 8 If $S(\mathbf{p}_{k+1}) > S(\mathbf{p}_k)$, multiply λ by ν and go to Step 4
Step 8	If satisfying convergent criteria, go to Step 9, else set $k = k$+1 and go to Step 2
Step 9	Minimization is terminated.

Thus, in three-dimensional modeling, the simplex is a tetrahedron. In minimization problems, the basic algorithm is to gradually move a simplex toward an optimum point with a minimum objective function for four vertices on the simplex. To achieve the optimum point, the movement of the simplex uses three operations including reflection, contraction and expansion (Finsterle, 2007a; Rao, 2009).

2.3.4.5 Simulated Annealing Method

The simulated annealing method is based on the process of slow cooling of heated solids, known as annealing (Finsterle, 2007a; Rao, 2009). If the temperature of the molten metal decreases by very fast rate, the metal reaches an incomplete solid state with high internal energy. To achieve a more complete crystalline state, the cooling rate needs to be controlled. The Metropolis algorithm is

$$\emptyset_k = e^{-\Delta S/\tau_k} \tag{2.47}$$

in which ΔS is the difference between objective functions ($\Delta S = S(\mathbf{p}_{k+1}) - S(\mathbf{p}_k)$), $\emptyset_k$ is the Metropolis criterion and τ_k is current temperature, or

$$\tau_k = \alpha^k \tau_0 , \tag{2.48}$$

$$\alpha^k = (1 - k/\boldsymbol{K})^{\beta}. \tag{2.49}$$

where, α is the temperature reduction factor, τ_0 is initial temperature, $\boldsymbol{K}$ is the total number of iterations, k is the number of the current step, and $\beta > 1$ is constant.

In application, the temperature is replaced by parameter $\mathbf{p}$. The Metropolis criterion ($\emptyset_k$) is used to determine if $\mathbf{p}_{k+1}$ at the next point (k+1) is acceptable (within probability). The reduction factor (α; $0 < \alpha < 1$) minimizes the objective function with parameters for successful convergence. The total number of iterations ($\boldsymbol{K}$) is examined to

determine if maximum iterations (specified) are exceeded. Choosing appropriate values of initial temperature τ_0, α and $\boldsymbol{K}$ is very important for successful convergence.

This method has the advantage of searching local minima. However, this method is not efficient, relatively, because $\mathbf{p}_{k+1}$ is chosen randomly. Therefore, for iTOUGH2 modeling, it is recommended to use the Simulated Annealing method by combining with other minimization algorithms (Finsterle, 2007a).

2.3.4.6 Grid Search Method

The Grid Search method evaluates the objective function at all grid points, so a suitable grid can be designed. This method can be used for inversion with the small number of parameters (Rao, 2009). Therefore, it is not recommended for iTOUGH2 modeling with large numbers of parameters. A further detailed overview of this grid search method is summarized in Rao (2009).

2.4 References

Altevogt, A. S., and Celia, M. A. (2004). "Numerical modeling of carbon dioxide in unsaturated soils due to deep subsurface leakage." *Water Resources Research*, 40(3).

Altunin, V. (1975). *Thermophysical properties of carbon dioxide*, Publishing House of Standards, Moscow, Russia.

Anderson, M. P., and Woessner, W. W. (1992). *Applied groundwater modeling: simulation of flow and advective transport*, Elsevier Science, Oxford, UK.

Babbar, M., and Minsker, B. (2006). "Groundwater remediation design using multiscale genetic algorithms." *Journal of Water Resources Planning and Management*, 132(5), 341-350.

Beckford, O., Hilton, A. C., and Liu, X. "Development of an enhanced multi-objective robust genetic algorithm for groundwater remediation design under uncertainty." *Proc., Proc. Am. Soc. Civ. Eng*, 98.

Benjamin, J. R. (1970). *Probability, statistics, and decision for civil engineers*, McGraw-Hill, New York, NY.

Bennett, G. D. (1976). *Introduction to groundwater hydraulics*, U.S. Geological Survey, Denver, CO.

Bevington, P., and Robinson, D. (1969). *Data reduction and data analysis for the physical sciences*, McGraw-Hill, New York, NY.

Carroll, S., Hao, Y., and Aines, R. (2009). "Transport and detection of carbon dioxide in dilute aquifers." *Energy Procedia*, 1(1), 2111-2118.

Chan-Hilton, A. B., Iyer, S. K., Magar, V., and Kelley, M. "Optimization of natural attenuation with active remediation under uncertainty." *Proc., Seventh International In Situ and On-Site Bioremediation Symposium, Orlando, Florida, USA, 2-5 June 2003. Part H. Natural Attenuation, Long-Term Monitoring, and Site Closure*, Battelle Press, Columbus, OH.

Chen, C. S. (1989). "Solutions approximating solute transport in a leaky aquifer receiving wastewater injection." *Water Resources Research*, 25(1), 61-72.

Chen, Z., Huan, G., and Ma, Y. (2006). *Computational methods for multiphase flows in porous media*, Siam, Philadelphia, PA.

Christensen, S., and Cooley, R. (1999). "Evaluation of confidence intervals for a steady-state leaky aquifer model." *Advances in Water Resources*, 22(8), 807-817.

Cihan, A., Zhou, Q., and Birkholzer, J. T. (2011). "Analytical solutions for pressure perturbation and fluid leakage through aquitards and wells in multilayered-aquifer systems." *Water Resources Research*, 47(10), W10504.

Cobb, P., McElwee, C., and Butt, M. (1982). "Analysis of leaky aquifer pumping test data: An automated numerical solution using sensitivity analysis." *Groundwater*, 20(3), 325-333.

Committee, I. F. (1967). "A formulation of the thermodynamic properties of ordinary water substance." *IFC Secretariat, Düsseldorf, Germany*, 26.

Doughty, C., and Pruess, K. (2004). "Modeling supercritical carbon dioxide injection in heterogeneous porous media." *Vadose Zone Journal*, 3(3), 837-847.

Espinoza, F. P., Minsker, B. S., and Goldberg, D. E. (2005). "Adaptive hybrid genetic algorithm for groundwater remediation design." *Journal of Water Resources Planning and Management*, 131(1), 14-24.

Finsterle, S. (2004). "Multiphase inverse modeling." *Vadose Zone Journal*, 3(3), 747-762.

Finsterle, S. (2007a). "iTOUGH2 user's guide." *Report LBNL-40040*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007b). "iTOUGH2 command reference." *Report LBNL-40041*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007c). "iTOUGH2 sample problems." *Report LBNL-40042*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2010). "iTOUGH2 V3. 2, verification and validation report." *Report LBNL-42002*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S., Moridis, G., and Pruess, K. (1994). "A tough2 equation-of-state module for the simulation of two-phase flow of air, water, and a miscible gelling liquid." *Report LBL-36086*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Garcia, J. E. (2001). "Density of aqueous solutions of CO2." *Report LBNL-49023*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Gasda, S. E., Bachu, S., and Celia, M. A. (2004). "Spatial characterization of the location of potentially leaky wells penetrating a deep saline aquifer in a mature sedimentary basin." *Environmental Geology*, 46(6-7), 707-720.

Gasda, S. E., Wang, J. Z., and Celia, M. A. (2011). "Analysis of in-situ wellbore integrity data for existing wells with long-term exposure to CO2." *Energy Procedia*, 4, 5406-5413.

Hou, Z., Murray, J. C., and Rockhold, L. M. (2012). "CO2 migration in intact caprock and leakage risk in three-dimensional heterogeneous formations." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittssburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012a). "Early detection of brine or CO2 leakage through high-permeability pathways using pressure-based monitoring data." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittsburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012b). "Impact of data uncertainty on identifying leakage pathways in CO2 geologic storage systems and estimating their hydrogeological properties by inverse modeling." *TOUGH Symposium 2012,* Lawrence Berkeley National Laboratory, Berkeley, CA.

Ko, N.-Y., and Lee, K.-K. (2008). "Reliability and remediation cost of optimal remediation design considering uncertainty in aquifer parameters." *Journal of Water Resources Planning and Management*, 134(5), 413-421.

Krevor, S., Perrin, J.-C., Esposito, A., Rella, C., and Benson, S. (2010). "Rapid detection and characterization of surface CO2 leakage through the real-time measurement of δ13 C signatures in CO2 flux from the ground." *International Journal of Greenhouse Gas Control*, 4(5), 811-815.

Liggett, J. A., and Chen, L.-C. (1994). "Inverse transient analysis in pipe networks." *Journal of Hydraulic Engineering*, 120(8), 934-955.

Metz, B., Davidson, O., de Coninck, H., Loos, M., and Meyer, L. (2005). *IPCC special report on carbon dioxide capture and storage*, Cambridge University Press, New York, NY.

Nogues, J. P., Nordbotten, J. M., and Celia, M. A. (2011). "Detecting leakage of brine or CO2 through abandoned wells in a geological sequestration operation using pressure monitoring wells." *Energy Procedia*, 4, 3620-3627.

Nordbotten, J. M., Celia, M. A., and Bachu, S. (2004). "Analytical solutions for leakage rates through abandoned wells." *Water Resources Research*, 40(4), W04204.

Nordbotten, J. M., Kavetski, D., Celia, M. A., and Bachu, S. (2008). "Model for CO2 leakage including multiple geological layers and multiple leaky wells." *Environmental Science & Technology*, 43(3), 743-749.

Onuma, T., and Ohkawa, S. (2009). "Detection of surface deformation related with CO2 injection by DInSAR at In Salah, Algeria." *Energy Procedia*, 1(1), 2177-2184.

Phillips, S. L., Igbene, A., Fair, J., Ozbek, H., and Tavana, M. (1981). *A technical databook for geothermal energy utilization*, Lawrence Berkeley Laboratory, University of California, Berkeley, CA.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in Fortran 77: The art of scientific computing second edition*, Cambridge University Press, New York, NY.

Pruess, K. (1987). "TOUGH user's guide, nuclear regulatory commission report NUREG/CR-4645." *Report LBL-20700*, Lawrence Berkeley Laboratory Berkeley, CA.

Pruess, K. (2004). "Numerical simulation of CO2 leakage from a geologic disposal reservoir, including transitions from super-to subcritical conditions, and boiling of liquid CO2." *Spe Journal*, 9(2), 237-248.

Pruess, K. (2005). "ECO2N: A TOUGH2 fluid property module for mixtures of water, NaCl, and CO2." *Report LBNL-57952*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Pruess, K., and García, J. (2002). "Multiphase flow dynamics during CO2 disposal into saline aquifers." *Environmental Geology*, 42(2-3), 282-295.

Pruess, K., Moridis, G., and Oldenburg, C. (1999). "TOUGH2 user's guide, version 2.0." *Report LBNL-43134*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Rao, S. S. (2009). *Engineering optimization: Theory and practice*, John Wiley & Sons, Hoboken, New Jersey.

Singh, S. K. (2009). "Simple method for quick estimation of leaky-aquifer parameters." *Journal of Irrigation and Drainage Engineering*, 136(2), 149-153.

Spycher, N., and Pruess, K. (2005). "CO2-H2O mixtures in the geological sequestration of CO2. II. Partitioning in chloride brines at 12–100° C and up to 600 bar." *Geochimica et Cosmochimica Acta*, 69(13), 3309-3320.

Sun, A. Y., Zeidouni, M., Nicot, J.-P., Lu, Z., and Zhang, D. (2013). "Assessing leakage detectability at geologic CO2 sequestration sites using the probabilistic collocation method." *Advances in Water Resources*, 56, 49-60.

Zheng, C., Bennett, G. D., Melton, J., and Simon, A. R. (2002). "Applied contaminant transport modeling." *Industrial and Commercial Training*, 34(7), 256-262.

Zhou, Q., Birkholzer, J. T., and Tsang, C.-F. (2009). "A semi-analytical solution for large-scale injection-induced pressure perturbation and leakage in a laterally bounded aquifer–aquitard system." *Transport in Porous Media*, 78(1), 127-148.

CHAPTER 3

FORWARD SIMULATION USING ITOUGH2

This chapter discusses my sensitivity analysis and flow analysis of a leakage pathway through a confining layer; I developed all simulations with the iTOUGH2 simulator.

For an idealized system to realize pressure perturbations induced by injection or pumping (Cihan et al., 2011), the conceptual domain should ideally consist of at least three layers, such as two sandstone layers and one confining layer for this analysis. It is assumed that each sandstone layer is homogenous or heterogeneous, isothermal, the entire domain is completely saturated by brine, and CO_2 is injected in an underlying formation. This sensitivity analysis quantifies impact of uncertainty of hydrogeological properties in terms of pressure anomalies in the overlying formation. Forward simulations show pressure perturbation and migration of CO_2 through the leakage pathway. The pressure data at simulated monitoring wells are used as "observed data" for the inverse simulation afterward.

3.1 Conceptual Domain for Homogeneous Modeling

A simplified conceptual domain is designed and parameterized for CO_2 storage in a homogeneous reservoir. Fig. 3.1 (a) is a schematic of multiple aquifers with a single

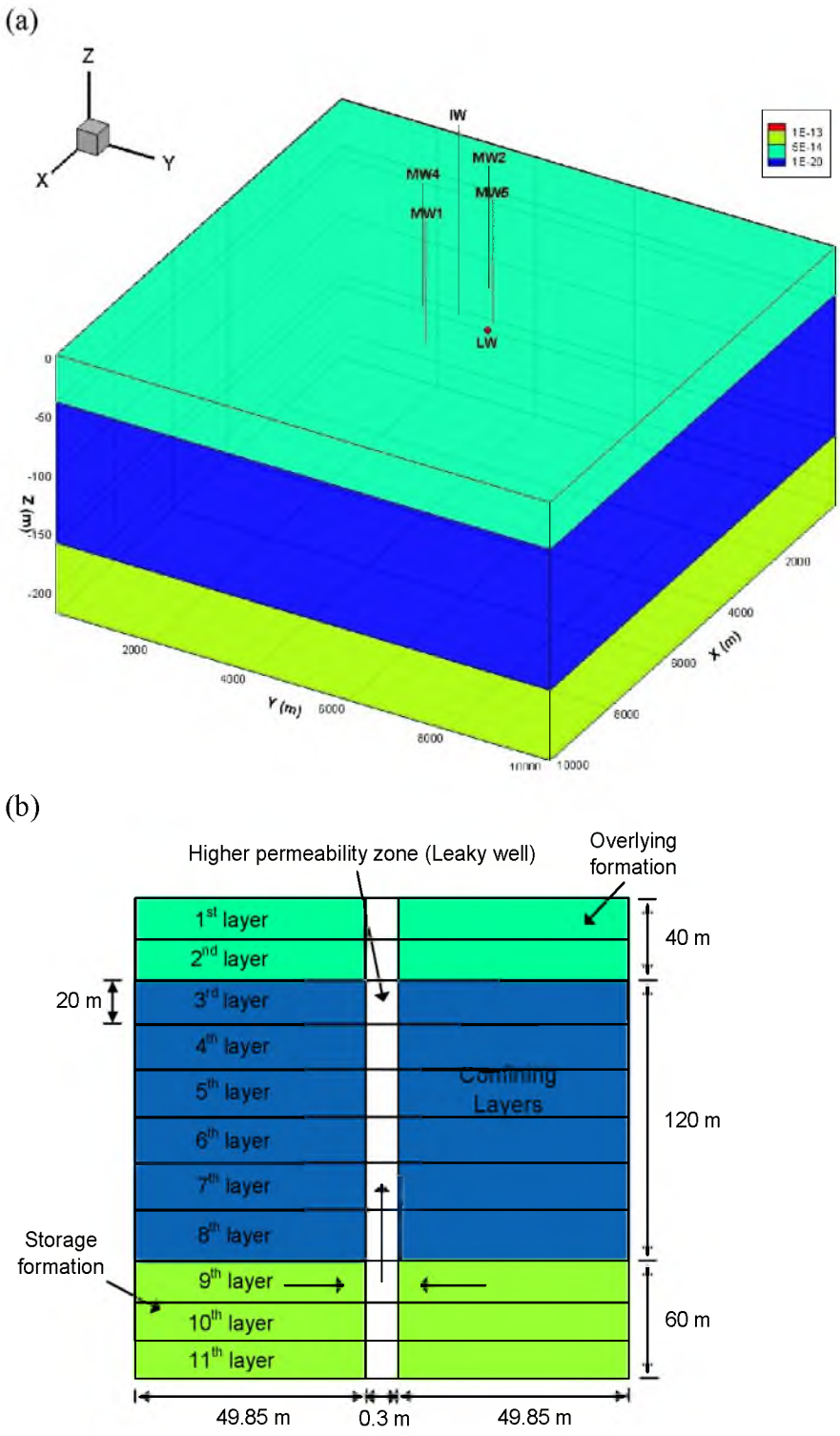


Fig. 3.1 Conceptual domain: (a) Schematic of model of multiple formations with leakage pathway. The permeability of the storage reservoir, the cap rock and the overlying formation are 10^{-13} m^2, 10^{-20} m^2 and 10^{-15} m^2, respectively. IW: injection well, MW: monitoring well, and LW: leakage well, and (b) schematic of specified LW (not to scale).

leakage pathway. Fig. 3.1 (b) is a schematic of the single leakage pathway. The domain consists of a storage formation, a confining formation (cap rock), and an overlying formation. The overlying and the storage formations are composed of sandstone with appropriate permeability, and the cap rock consists of shale with lower permeability and is located at the middle of the conceptual domain. Each formation is homogeneous. The domain size is 10,100 m × 10,100 m × 220 m. The number of cells is 103 × 103 × 11 (116,699 grid blocks total). The conceptual model is completely saturated with brine and CO_2 is injected to induce transient release of leakage.

The XZ-planes on the left and right boundaries (Fig. 3.1 (a)) are assigned a constant head boundary condition, but other boundaries are assigned no flow boundary conditions. The assumed leakage pathway vertically penetrates the cap rock at (x, y) = (5250 m, 6050 m) from the origin. If CO_2 is injected into the storage formation, pressure buildup in the storage formation results, mobilizes brine and/or CO_2 into the overlying formation through the leakage pathway. Thus, the anomalies of pressure in the overlying formation are induced by leakage of brine/CO_2. This simulation is isothermal, so every grid was specified as 50 °C and constant. Initial salt mass fraction and CO_2 mass fraction were assigned as 0.05 (5.0 wt.-% NaCl) and 0.0, respectively. Initial pressures of all top grids and all bottom grids were specified as approximately 10 MPa and 12 MPa, respectively, and other cells are assigned a linear distribution with the same pressure gradient in the vertical direction, to keep hydrostatic conditions. In addition, injected CO_2 can sustain supercritical conditions in the simulation domain. The porosity of the overlying and storage formations and the leakage pathway is 0.2, and the porosity of the caprock is assigned as 0.02. The pore compressibility (Pa^{-1}) is assumed as 0 in the

conceptual domain so that porosity remains constant. Table 3.1 summarizes the dimensions of the model. Table 3.2 denotes assigned CO_2 injection conditions. Table 3.3 presents initial conditions. Table 3.4 details the point locations of monitoring "wells." In the model domain, four monitoring wells and an injection well are available for pressure observation. The four monitoring wells measure pressure data in both the overlying and storage formations, and the injection well observes in only the overlying formation. The pressure anomalies (due to brine/CO_2 leakage) observed in those monitoring wells will be used for the inverse analysis to estimate the leakage location.

Table 3.1 Dimensions of the conceptual model.

<table>
<tr><td>Cubic size (m)</td><td colspan="2">10,100 × 10,100 × 220</td><td rowspan="9">Permeability (m^2)</td><td>Storage formation</td><td>$k_x=k_y=k_z=10^{-13}$</td></tr>
<tr><td>Each normal cell size (m)</td><td colspan="2">100 × 100 × 20</td><td rowspan="2">Overlying formation</td><td rowspan="2">$k_x=k_y=k_z=10^{-15}$</td></tr>
<tr><td>Cell sizes including a leakage pathway (m)</td><td colspan="2">49.85 × 49.85 × 20, 0.3× 0.3 × 20 and 49.85 × 49.85 × 20</td></tr>
<tr><td>Number of cells</td><td colspan="2">103 × 103 × 11 (116,699 total)</td><td rowspan="3">Cap rock</td><td rowspan="3">$k_x=k_y=k_z=10^{-20}$</td></tr>
<tr><td>Leakage pathway location from origin</td><td colspan="2">(5250 m, 6050 m)</td></tr>
<tr><td>Simulation time (sec)</td><td colspan="2">0 – 3.16e8 (≈ 10 yrs)</td></tr>
<tr><td>Time step size (sec)</td><td colspan="2">10</td><td rowspan="3">Leakage pathway</td><td rowspan="3">$k_{lx}=k_{ly}=10^{-20}$ and $k_{lz}=10^{-10}$</td></tr>
<tr><td>Tolerance</td><td colspan="2">1.0e-7</td></tr>
<tr><td rowspan="2">Pore compressibility (Pa^{-1})</td><td>Both aquifers</td><td>0.0</td><td rowspan="2">Porosity</td><td>Both aquifers</td><td>0.2</td></tr>
<tr><td>Cap rock</td><td>0.0</td><td>Cap rock</td><td>0.02</td></tr>
</table>

Table 3.2 Rate of injected CO_2.

IW Location	CO_2 injection condition	
	Injection time (sec)	Injected mass (kg/s)
(5050 m, 5050 m, -190 m)	0	63.4
	3.16e8	63.4

Table 3.3 Initial conditions assigned in the model.

Initial conditions				
Locations	**Initial pressure**	**Salt mass fraction**	**CO_2 mass fraction**	**Temperature (°C)**
Top boundary	1.0e7	0.05	0.0	50
Bottom boundary	1.2e7	0.05	0.0	50

Table 3.4 Location points of the five monitoring wells.

	Distance from origin				
	1st well	**2nd well**	**3rd well (IW)**	**4th well**	**5th well**
Measurement point at overlying formation	6050m, 5050m, -30m	4050m, 5050m, -30m	5050m, 5050m, -30m	5050m, 4050m, -30m	5050m, 6050m, -30m
Measurement point at storage formation	6050m, 5050m, -170m	4050m, 5050m, -170m	-	5050m, 4050m, -170m	5050m, 6050m, -170m
	Distance from LW				
	800m, 1000m	1,200m, 1,000m	200m, 1,000m	200m, 2,000m	200m, 0m

The Van Genuchten (1980) and Corey (1954) functions were used for relative permeability. The van Genuchten-Mualem model was implemented for capillary pressure (Van Genuchten, 1980). The relative permeability for liquid phase (brine) from van Van Genuchten (1980) is

$$k_{rl} = \sqrt{S^*}\left(1 - \left(1 - \left(S^*\right)^{1/\lambda}\right)^{\lambda}\right)^2 \tag{3.1}$$

where $0 \le k_{rl} \le 1$, and $S^* = (S_l - S_{lr})/(1 - S_{lr})$. The relative permeability for gas phase (CO_2) due to Corey (1954) is

$$k_{rg} = (1 - \hat{S})^2\left(1 - \hat{S}^2\right) \tag{3.2}$$

where $0 \le k_{rg} \le 1$, and $\hat{S} = (S_l - S_{lr})/(1 - S_{lr} - S_{gr})$. The capillary pressure function is

$$P_{cap} = -P_0\left(\left(S^*\right)^{-1/\lambda} - 1\right)^{1-\lambda} \tag{3.3}$$

where $-P_{max} \le P_{cap} \le 0$, and $S^* = (S_l - S_{lr})/(1 - S_{lr})$. Table 3.5 details parameter values for the relative permeability and capillary pressure functions.

Table 3.5 Relative permeability and capillary pressure parameters.

Relative permeability	**Parameter values**
Liquid: van Genuchten function	
Irreducible water saturation (S_{lr})	0.20
Exponent (λ)	0.457
Gas: Corey curve	
Irreducible gas saturation (S_{gr})	0.05
Capillary pressure	**Parameter values**
van Genuchten function	
Irreducible water saturation (S_{lr})	0.20
Exponent (λ)	0.475
Strength coefficient (P_0)	19.61 kPa
Maximum capillary pressure (P_{max})	10^7 Pa

3.2 Sensitivity Analysis with Respect to Pressure Anomalies

For CCUS to be effective in reducing emissions, the amounts of CO_2 injected in the storage formation will be very large. Such injection will cause significant pressure buildup in the storage formation, and if CO_2 or brine leaks through cap rock discontinuities such as faults and abandoned wells, that leakage may influence pressures in the overlying formation. In fact, pressure anomalies in the overlying formation may vary much, depending on hydrogeological properties in the CO_2 storage system. In particular, pressure anomalies may be subject to flow rates of CO_2 or brine through leakage pathways based on effective permeability and cross-sectional area of the leakage pathways (Jung et al., 2012b). In addition, migration of CO_2 or brine through cap rock without leakage pathways can also increase pressures in the overlying formation. Such pressure effects may render indistinguishable the pressure perturbations due to CO_2 or brine leaks through discontinuities of cap rock, reducing efficiency of identification of leakage pathways by inverse analysis (Jung et al., 2012b). Thus, the sensitivity analysis here focuses on how details of hydrogeological properties affect the pressure signals at monitoring wells. It will also be a means for parameterization of hydrogeologic properties of the model domain.

The sensitivity of measurements of the hydrogeological properties is closely related to the accuracy of associated inverse solutions. If an unknown parameter of inverse analysis does not significantly affect the relevant dependent variable of forward analysis, such as pressure, the inverse analysis may not yield accurate solutions. For instance, if leakage pathways in the form of high permeability, the main unknown parameter of this inverse analysis, are so low that pressures at monitoring wells are

unaffected, leakage pathways cannot be identified through inverse simulation. This sensitivity analysis was carried out with respect to pressure perturbations in the overlying formation via examination of a sensitivity coefficient ($\partial P/\partial a$) where P: pressure, a: parameter value.

The difference of pressure (dP) is calculated from pressure in the overlying formation between (1) with the leakage pathway, and (2) without the leakage pathway in the domain of Fig. 3.1. The parameters influencing dP are the permeability of the pathway, permeability of the overlying formation, cap rock thickness, permeability of the cap rock, and so on. While Jung et al. (2012b) examined the sensitivity of dP to the permeability of the cap rock, they suggested that the most influential parameter is indeed the permeability of the cap rock. They concluded that a seal layer of 100 m thickness with permeability lower than 10^{-18} m^2 will not facilitate significant diffuse leakage through the seal layer. However, other parameters can also result in a marked difference of pressures, so examination of sensitivity to those parameters is necessary to improve leakage detection by inverse analysis. Therefore, this sensitivity analysis focuses on the permeability of the leakage pathway, the permeability of overlying formation, and the thickness of cap rock.

Firstly, the sensitivity analysis considers five different vertical permeability values of the leakage pathway (k_{lz}=10^{-10}, 10^{-13}, 10^{-15}, 10^{-17}, and 10^{-18} m^2) in the system. Fig. 3.2 presents the pressure differences in the overlying formation from each different permeability permutation of the leakage pathway, after 10 years simulated time. In Fig. 3.2, the dP values with contour lines may be analyzed to ascertain pressure anomalies at monitoring wells.

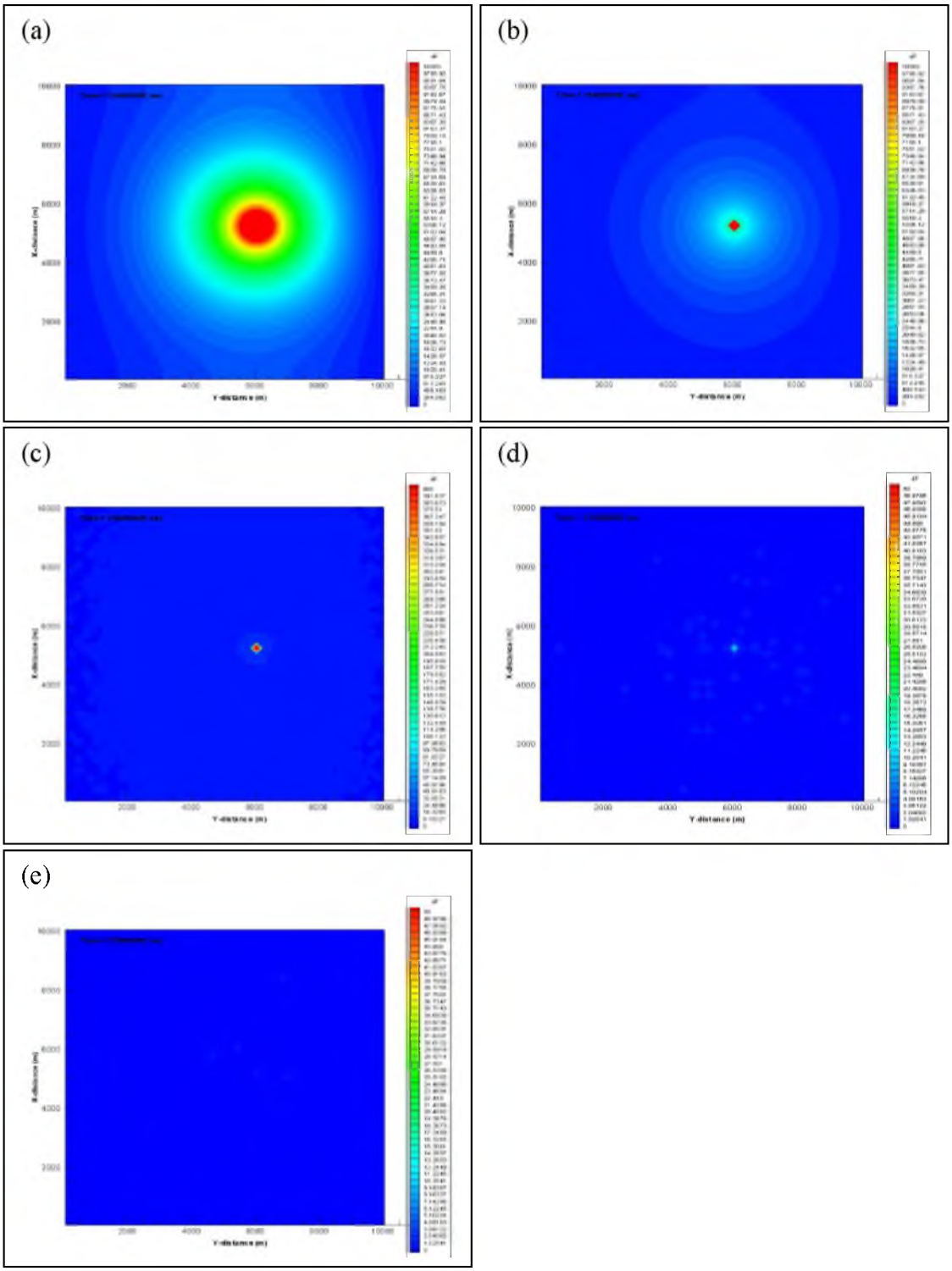


Fig. 3.2 Pressure differential results for five permeability values of the leakage pathway: (a) $k_{lz}=10^{-10}m^2$ (*dP* scale: 0 - 10,000 Pa), (b) $k_{lz}=10^{-13}m^2$ (*dP* scale: 0 - 10,000 Pa), (c) $k_{lz}=10^{-15}m^2$ (*dP* scale: 0 - 400 Pa), (d) $k_{lz}=10^{-17}m^2$ (*dP* scale: 0 - 50 Pa) and (e) $k_{lz}=10^{-18}m^2$ (*dP* scale: 0 - 50 Pa), after 10 years simulated time.

If the leakage pathway has permeability greater than 10^{-13} m^2, the monitoring wells can detect pressure anomalies due to leakage (Fig. 3.2 (a) and (b)). On the other hand, if the leakage pathway is assigned permeability between 10^{-15} and $10^{-17}m^2$, the area of pressure perturbations is significantly smaller than cases of higher permeability (10^{-10} and 10^{-13} m^2). For lower permeabilities, only the 5th monitoring well, which is 200 m away from the leakage pathway (see Table 3.4) can detect substantial pressure anomalies (Fig. 3.2 (c) and (d)). If the system does not have the 5th monitoring well, the leakage pathway with permeability values from 10^{-15} to 10^{-17} m^2 may not be identified through inverse simulation using pressure anomalies within 10 years. This implies that the possibility of leakage detection using inverse analysis can significantly depend on the relative distance between the monitoring wells and leakage pathways, at least for early leakage detection (within 10 years). Otherwise, for the pressure anomalies at the monitoring wells to be detected, the monitoring period has to be increased. In the lowest permeability case (10^{-18} m^2), the monitoring well cannot detect pressure perturbations within 10 years (Fig. 3.2 (e)), and that case is effectively a no-leak condition.

Secondly, this sensitivity analysis focuses on the permeability of overlying formation (kx = ky = kz = 10^{-13}, and 10^{-15} m^2). Fig. 3.3 represents the *dP* in the overlying formation from two different permeability permutations of the overlying formation, after 10 years. The sensitivity analysis quantified pressure perturbation in the overlying formation with 10^{-10} m^2 permeability assigned to the leakage pathway. For simulations assigning the overlying formation lower permeability (10^{-15} m^2) and for the 10^{-13} m^2 case, all monitoring wells can detect significant pressure anomalies due to leakage.

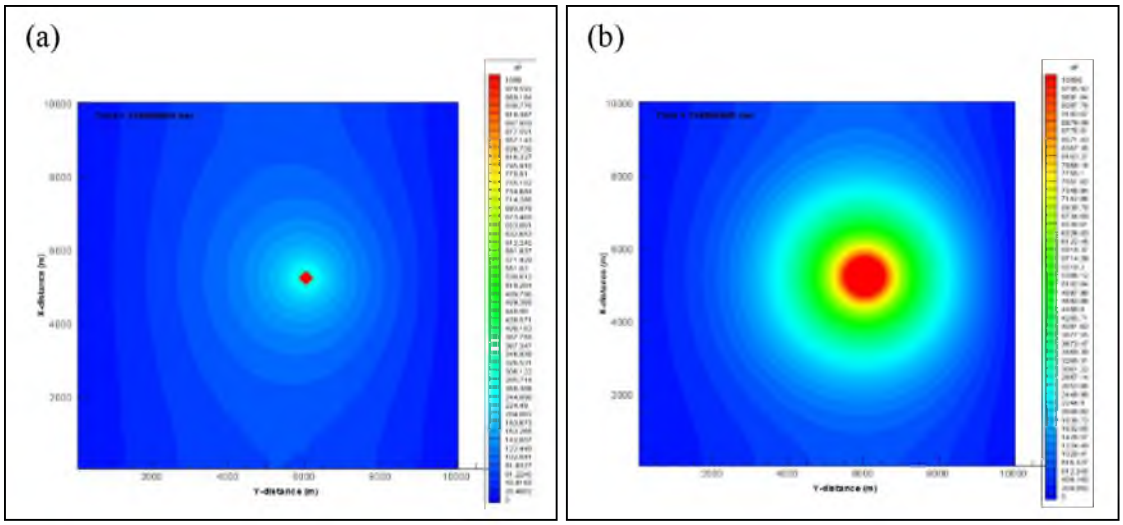


Fig. 3.3 Pressure differential results with overlying formation permeability: (a) 10^{-13} m^2 (dP scale: 0 - 1,000 Pa) and (b) 10^{-15} m^2 (dP scale: 0 - 10,000 Pa) after 10 years.

However, lower permeability of the overlying formation can increase pressure anomalies because the magnitude of dP is inversely proportional to permeability at least at a constant flow rate. Finally, the thickness of cap rock (60, 80, 100 and 120 m) was varied to examine how this parameter influences pressure in the overlying formation. Fig. 3.4 depicts the pressure differences in the overlying formation for each cap rock thickness, after 10 years in the case of both $k = 10^{-15}$ m^2 overlying formation permeability and $k_{lz} = 10^{-10}$ m^2 leakage pathway permeability. In Fig. 3.4, the pressure difference and the area significantly decrease as thickness of the seal layer decreases (after 10 years). The reason is that CO_2 or brine may diffuse through thinned cap rock (without leakage pathways) more quickly. As mentioned earlier, dP is difference of pressures between cases with and without a discrete leakage pathway, so dP may be reduced for cases of greater diffusion through cap rock. Diffusion process, therefore, may reduce efficacy of leakage pathway detection by inverse analysis.

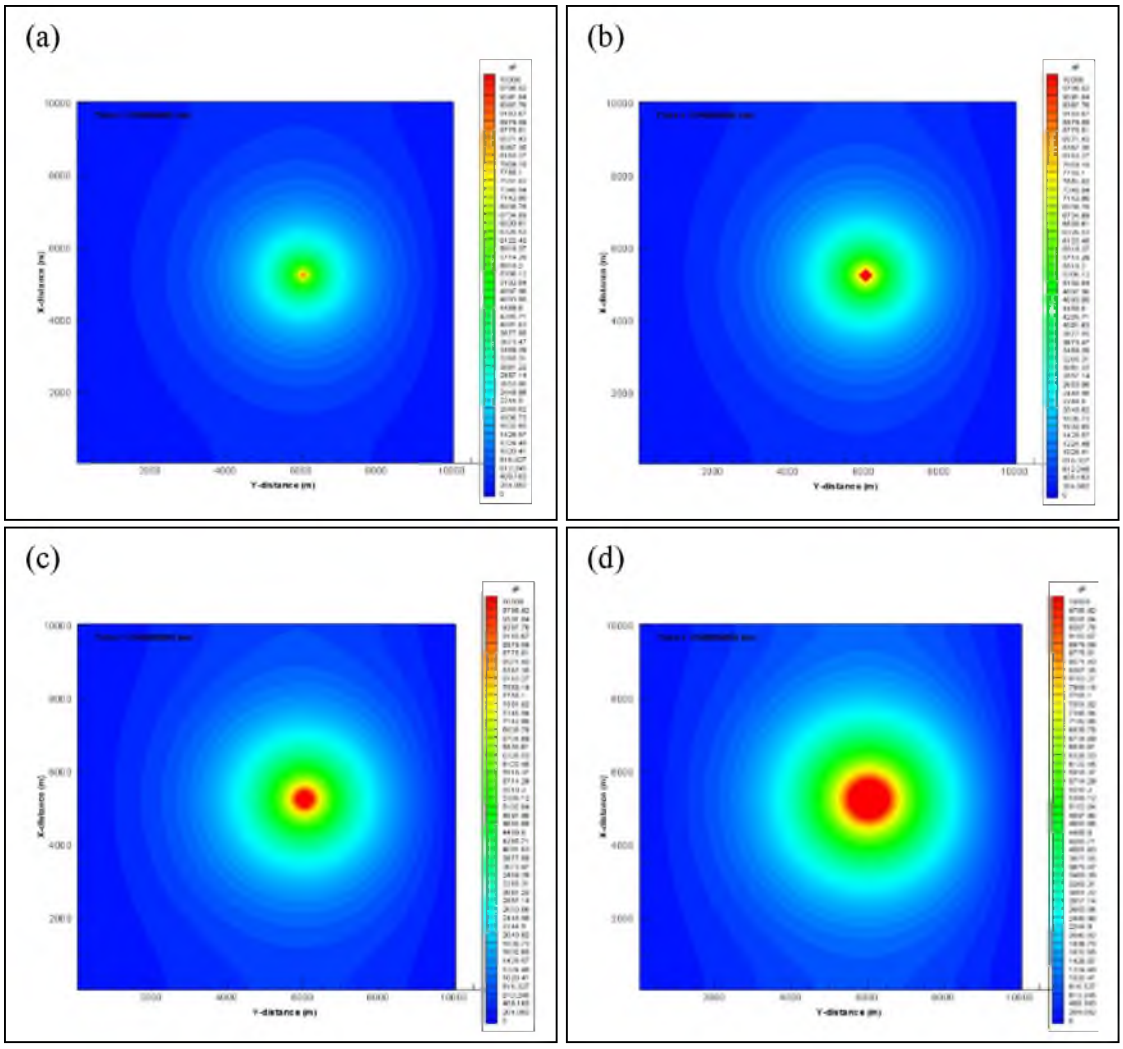


Fig. 3.4 Pressure differential results with cap rock thickness: (a) 60 m (*dP* scale: 0 - 10,000 Pa), (b) 80 m (*dP* scale: 0 - 10,000 Pa), (c) 100 m (*dP* scale: 0 - 10,000 Pa) and (d) 120 m (*dP* scale: 0 - 10,000 Pa), for the case of k = 10^{-15} m^2 after 10 years.

Fig. 3.5 illustrates sensitivity analysis results for cap rock thickness, but for all cases with permeability of overlying formation set to 10^{-13} m^2. The magnitude and area of pressure perturbations is substantially smaller than that for the overlying formation k = 10^{-15} m^2 results, after 10 years. In Fig. 3.5 (a) and (b), the 1st to the 4th monitoring wells cannot detect any value of dP over 200 Pa induced by leakage in the overlying formation.

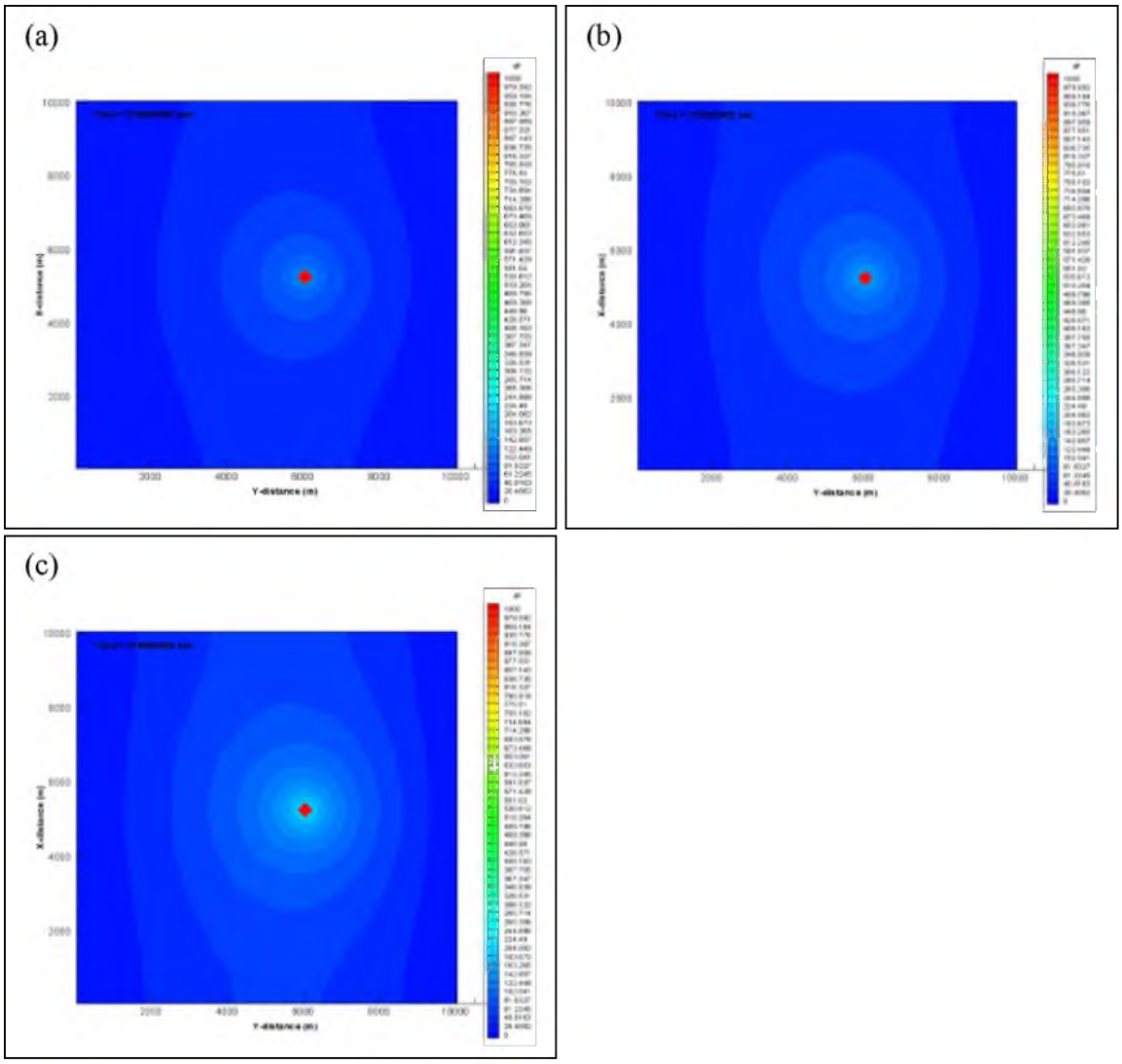


Fig. 3.5 Pressure differential results with cap rock thickness: (a) 60 m (*dP* scale: 0 - 1,000 Pa), (b) 80 m (*dP* scale: 0 - 1,000 Pa) and (c) 100 m (*dP* scale: 0 - 1,000 Pa), for the case of $k = 10^{-13}$ m^2 after 10 years.

Indeed, monitoring wells of a real site for CO_2 storage might exhibit much extraneous noise. The signals with excess extraneous noise can cause simulation errors and render it impossible to distinguish *dP* induced by leakage. That is, to increase possibility of early leakage detection, the *dP* must be large compared to the signal induced by noise.

Based on this sensitivity analysis of a generic system, hydrogeologic properties

like permeability of the overlying formation and the thickness of cap rock can have a significant impact on inverse analysis using pressure anomalies. To increase efficacy of leakage detection, it is recommended that the system has the following properties:

(1) Lower permeability (e.g., 10^{-15} m^2 or lower) of the overlying formation increases dP and thus can increase effectiveness of leakage detection through inverse simulation.

(2) Thicker caprock can reduce diffuse leakage and thus magnify pressure anomalies due to leakage pathways. If the overlying formation is of higher permeability ($k = 10^{-13}$ m^2), the cap rock thickness should probably be at least over 100 m.

(3) Leakage pathway permeability higher than at least 10^{-17} m^2 induces significant pressure anomalies through leakage pathways in the system.

3.3 Forward Simulation Results for Homogeneous Condition

Migration of CO_2 along a leakage pathway saturated by brine can be subject to buoyant and capillary effects, and may impact multidimensional flow in the formations (Pruess, 2005). The conceptual domain in Fig. 3.1 with given hydrogeological properties was employed to model the effects of CO_2 and brine migration through a leakage pathway. Fig. 3.6 illustrates CO_2 saturations during the simulation period (10 years) in the YZ-plane of the leakage pathway. CO_2 injected into the storage formation evolves a gas phase (CO_2). Buoyancy of CO_2 relative to brine slightly elevates the saturations of gaseous CO_2 until the CO_2 reaches cap rock (Fig. 3.6 (a)). CO_2 saturation builds up and laterally migrates, displacing some of the brine and partially dissolving in residual brine.

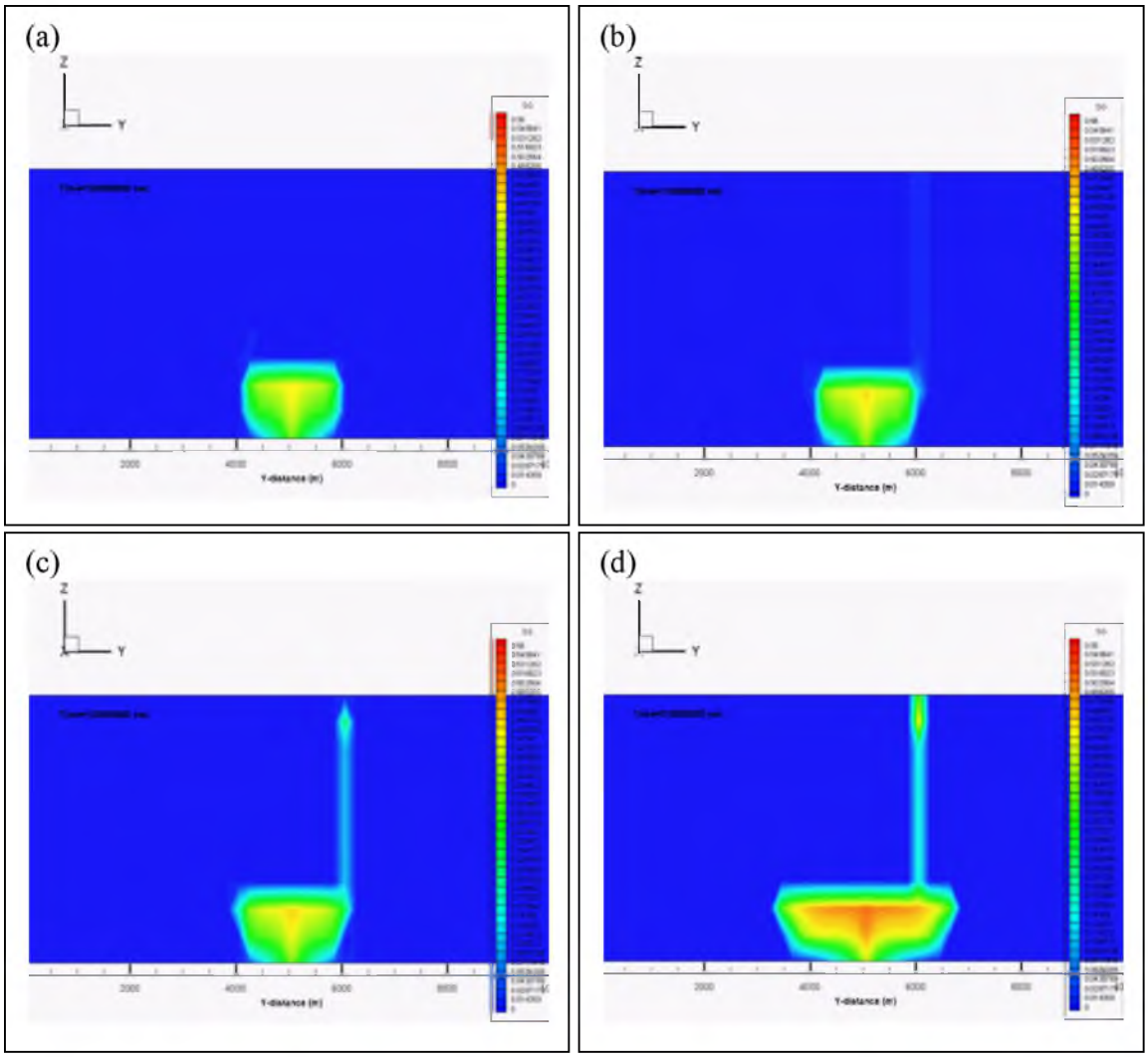


Fig. 3.6 Simulated CO2 saturations in the YZ-plane of the leakage pathway: (a) after 3.2 years, (b) after 3.5 years, (c) after 3.8 years and (d) after 10 years (SG scale: 0 - 0.56).

CO_2 reaches the bottom of the leakage pathway at approximately 1.1×10^8 seconds (3.5 years) and is rapidly transported into the overlying formation through the leakage pathway by pressurization and buoyant effects (Fig. 3.6 (b)). The magnitude of CO_2 saturation increases with increasing CO_2 leak rates into the overlying formation (Fig. 3.6 (c)). The buildup of CO_2 saturation in the overlying and storage formations continuously increases until 3.16×10^8 seconds (10 years). Fig. 3.7 presents the discharge rates of CO_2 and brine at the top of the leakage pathway.

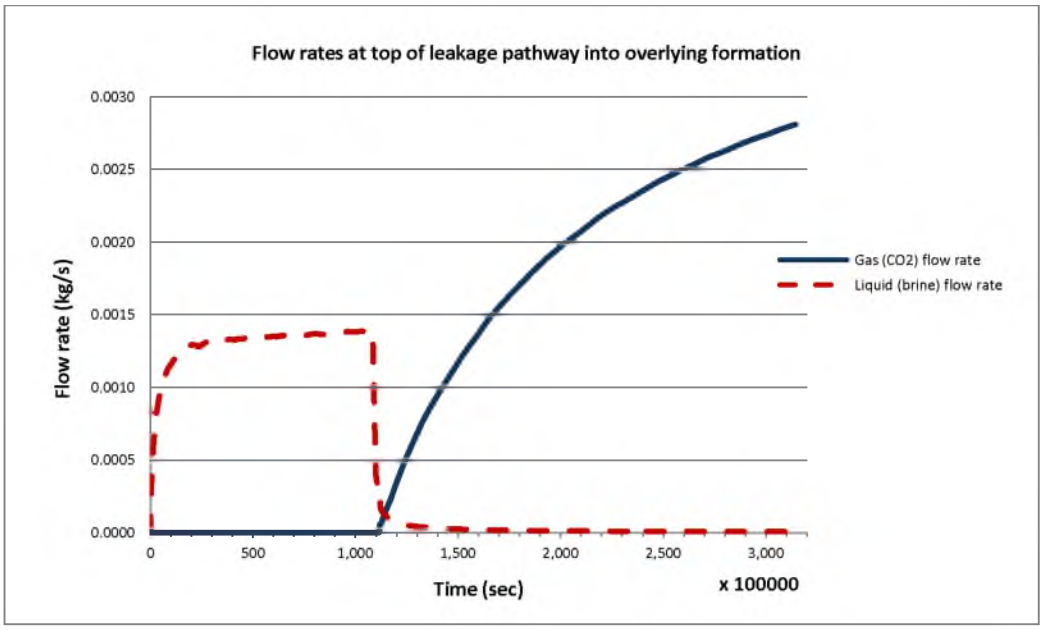


Fig. 3.7 Simulated CO2/brine discharge rates at the top of the leakage pathway.

When CO_2 injection starts, pressure buildup is propagated to the top of the leakage pathway, at which point it induces brine discharge. The outflow of brine lasts for approximately 1.1×10^8 seconds. However, after that time, CO_2 breaks through and starts to discharge from the leakage pathway; brine discharge is rapidly reduced because the relative permeability of brine is decreased and increased capillary pressure also reduces the pressure gradient of brine. In addition, flow rates of CO_2 increase due to higher relative permeability of CO_2. Even though brine flow rates rapidly decrease from 1.1×10^8 seconds, brine and CO_2 continue to discharge together from the leakage pathway until the end of the simulation.

Fig. 3.8 illustrates simulated pressure perturbations at the top of the leakage pathway. As shown in Fig. 3.7, outflows of brine increase before approximately 1.1×10^8 seconds, so the anomalies of pressure increase in the overlying aquifer as well as in the leakage pathway.

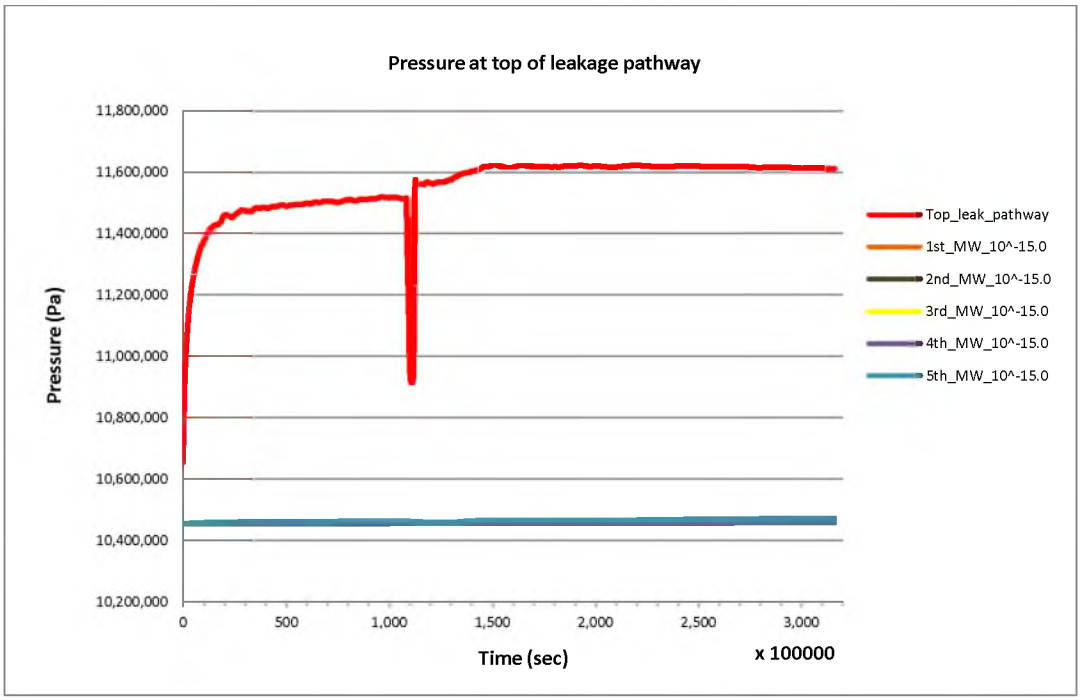


Fig. 3.8 Simulated pressure perturbations at the top of the leakage pathway.

However, at about 1.1×10^{8} seconds the flow rates of CO_2 reach the top of the leakage pathway, so brine saturation continues to drop, and capillary pressures are stronger. The pressures of brine are rapidly dropped at the top of leakage pathway, and then the pressures of CO_2 rapidly increase at the top of leakage pathway at approximately 1.12×10^{8} seconds because CO_2 relative permeability and saturation increase. In this dissertation the pressure anomalies by capillary pressure are defined as "capillary effects."

Fig. 3.9 illustrates simulated pressure perturbations in the YZ-plane of the leakage pathway. When flow rates of CO_2 reach the bottom of the leakage pathway at approximately 1.1×10^{8} seconds, brine leakage decreases in the leakage pathway, so pressure drops in the leakage pathway because of relative permeability reduction following reduced brine saturation. In addition, capillary pressure also causes brine pressures to decrease (Fig. 3.9 (b)).

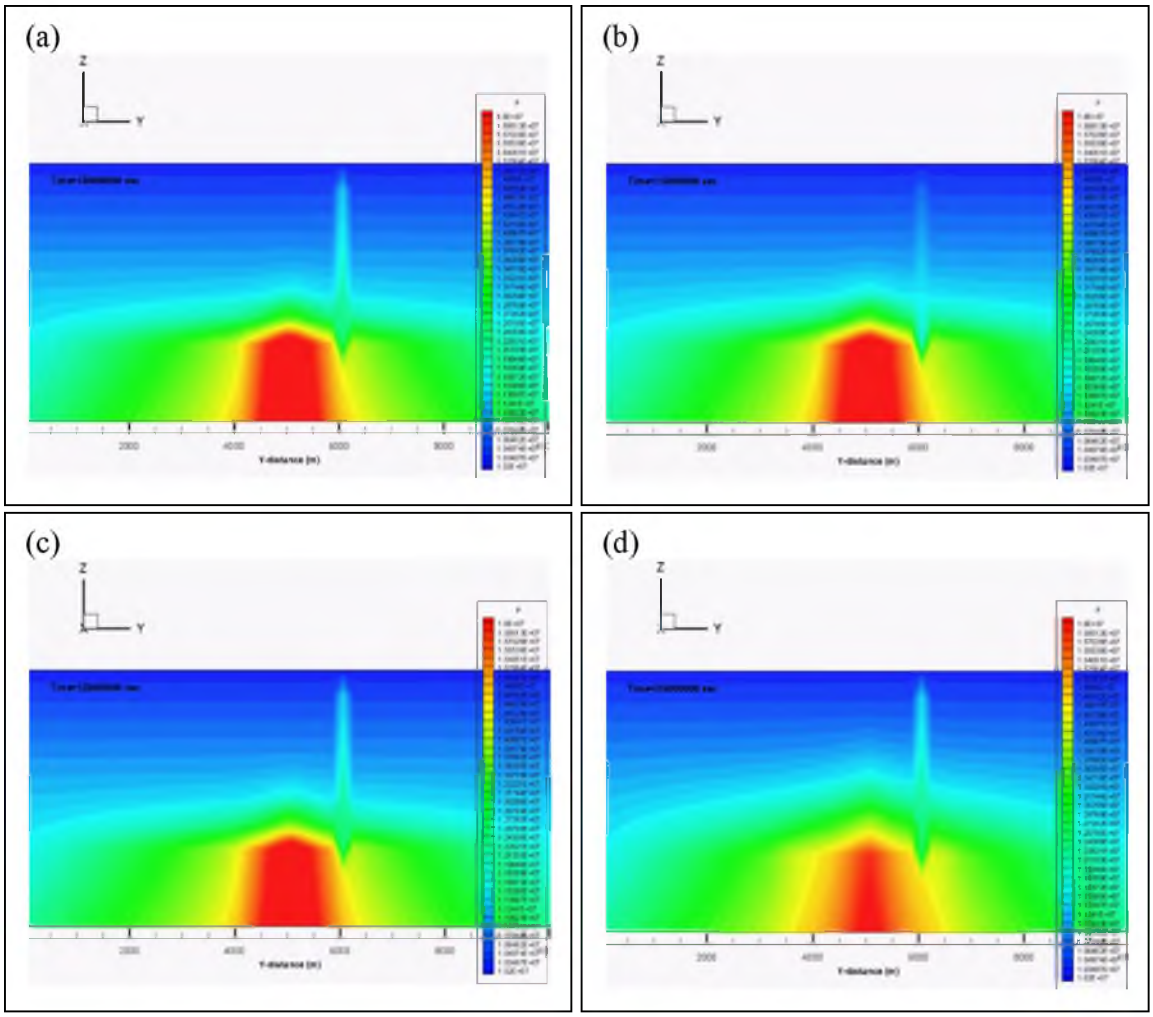


Fig. 3.9 Simulated pressure propagations in the YZ-plane of the leakage pathway: (a) after 3.2 years, (b) after 3.5 years, (c) after 3.8 years and (d) after 10 years.

After approximately 1.12×10^{8} seconds, outflow of CO_2 through the leakage pathway breaks through and then pressure increases in the leakage pathway and in the overlying formation (Fig. 3.9 (c) and (d)).

Fig. 3.10 illustrates simulated pressure propagation in the YZ-plane of the overlying formation. The overlying formation has 10^{-15} m^2 permeability. In the previous sensitivity analysis, the lower permeability of the overlying formation can increase pressure perturbations, so it can amplify leak source detection using inverse analysis.

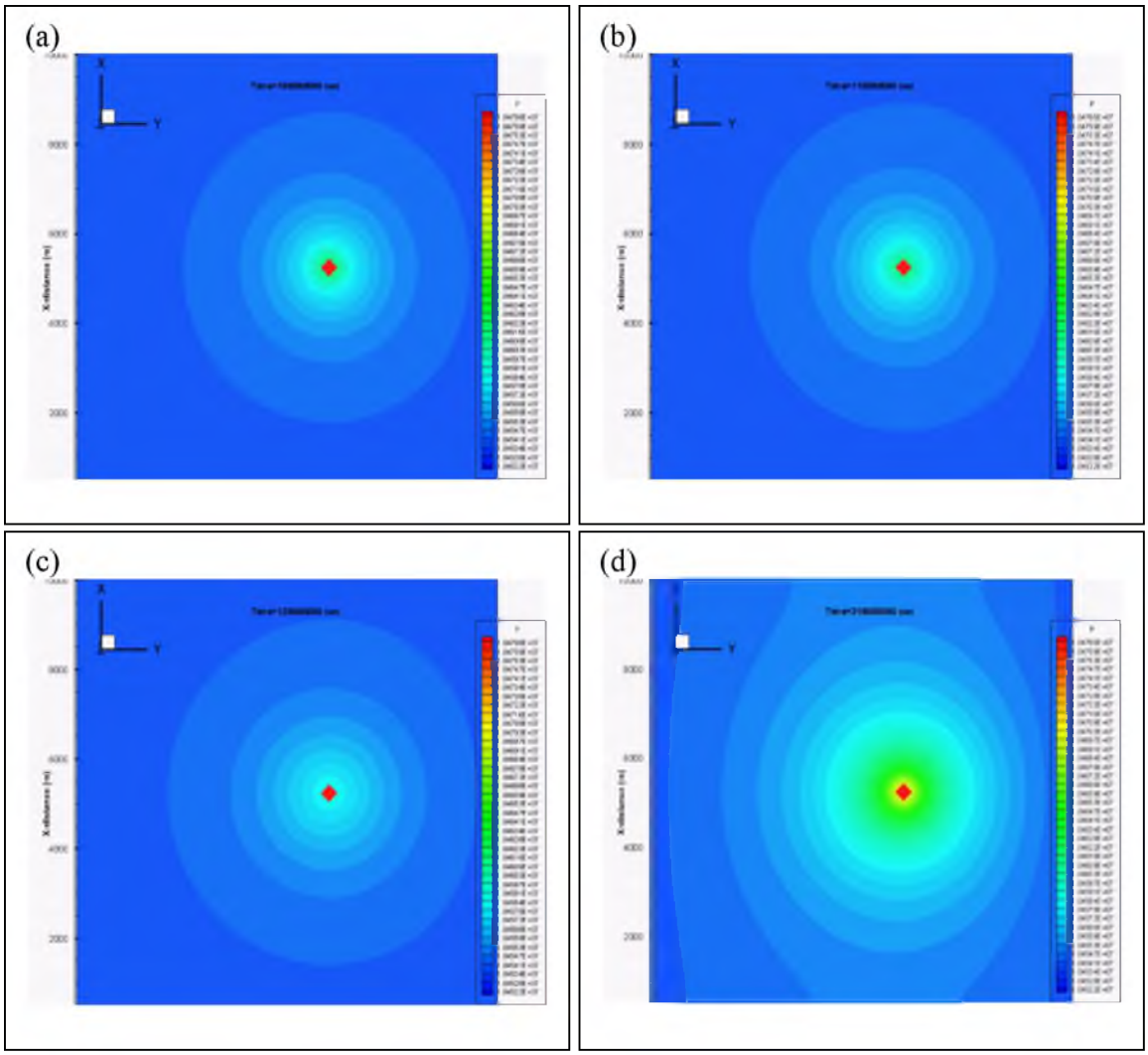


Fig. 3.10 Simulated pressure propagations in the XY-plane of the overlying formation: (a) after 3.2 years, (b) after 3.5 years, (c) after 3.8 years and (d) after 10 years.

Fig. 3.11 provides pressure profiles at monitoring "wells" in the overlying formation. Each pressure profile of a monitoring well exhibits a sudden change in pressure gradient. An explanation is that the pressure anomalies due to capillary pressures at the leakage pathway are propagated to the entire overlying aquifer. This has a significant effect on MW5 in the overlying aquifer, the closest of all monitoring wells to the leakage pathway. Such information can be very important for inverse analysis using pressure anomalies to estimate leakage pathways.

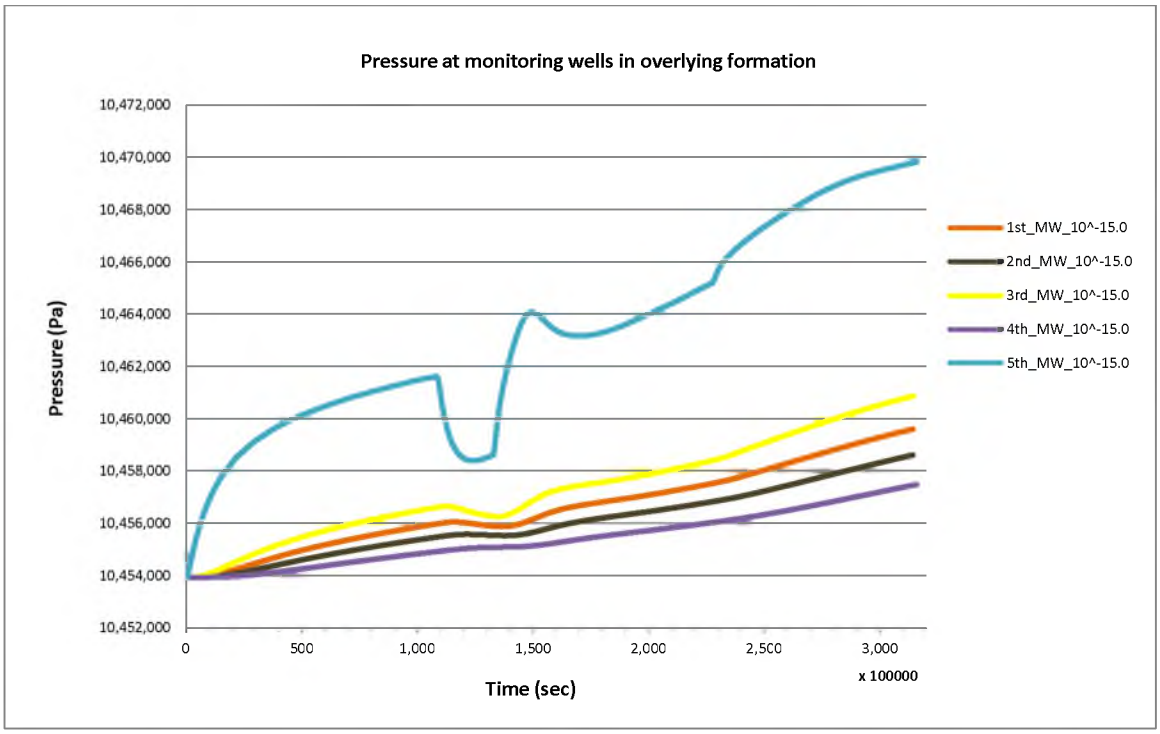


Fig. 3.11 Pressure anomalies at monitoring wells in the overlying formation.

3.4 Conceptual Domain for Heterogeneous Modeling

Field-scale studies always face difficulty in quantifying heterogeneity of the subsurface. Geostatistical and seismic methods are widely applied to generate realizations of heterogeneity in geological analyses (Finsterle, 2004). However, such is beyond the scope of this study. Rather than characterizing heterogeneity of a natural system, this study focuses on simulating migration of brine/CO_2 and associated pressure anomalies induced by discrete leakage pathways in a generic reservoir. Heterogeneity in the generic reservoir model was assigned to mimic the permeability distribution of the Scurry Area Canyon Reef Operations Committee (SACROC) unit. Fig. 3.12 depicts the characterized permeability distributions in the conceptual domain, including a caprock with permeability 10^{-20} m^2.

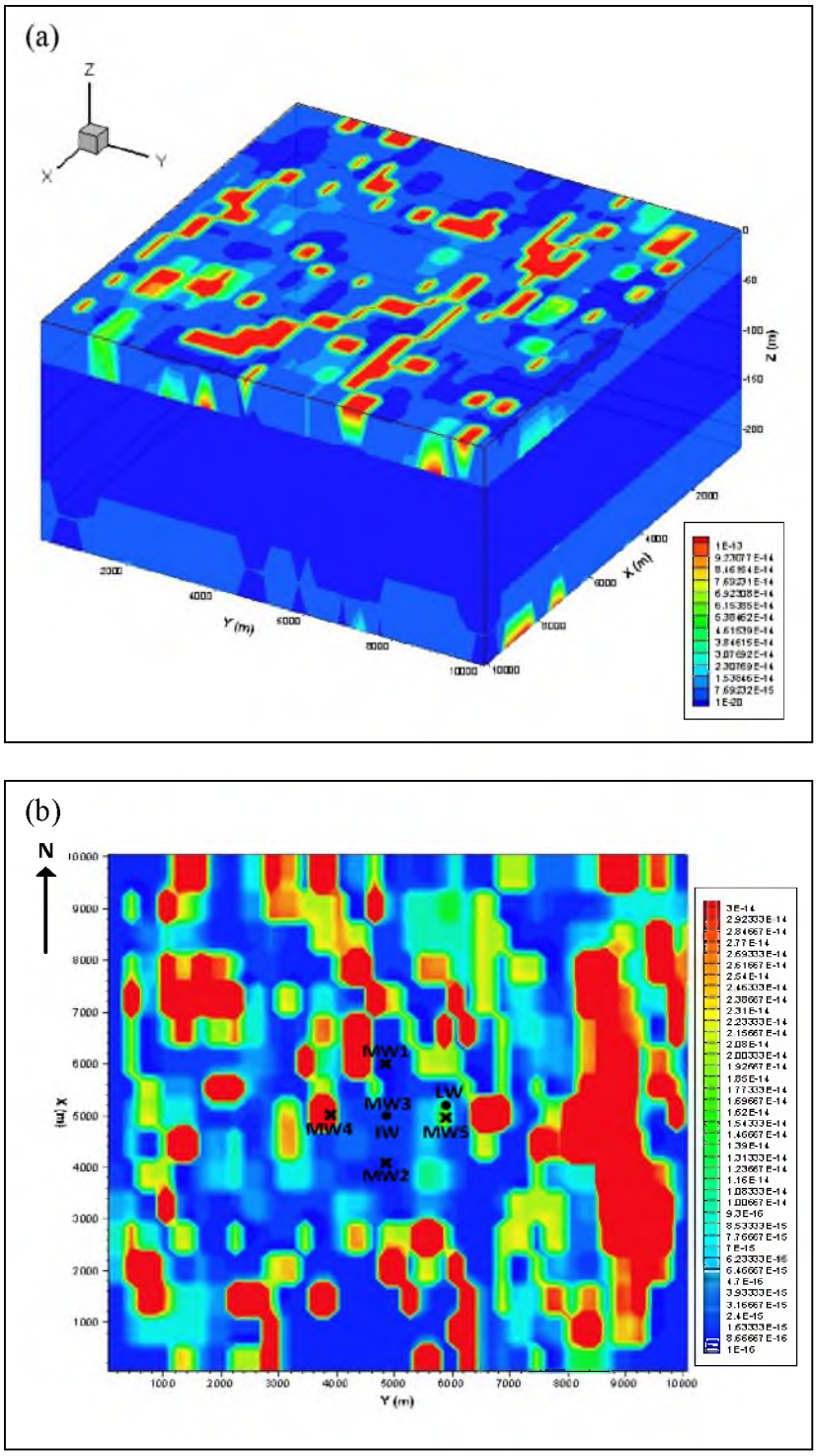
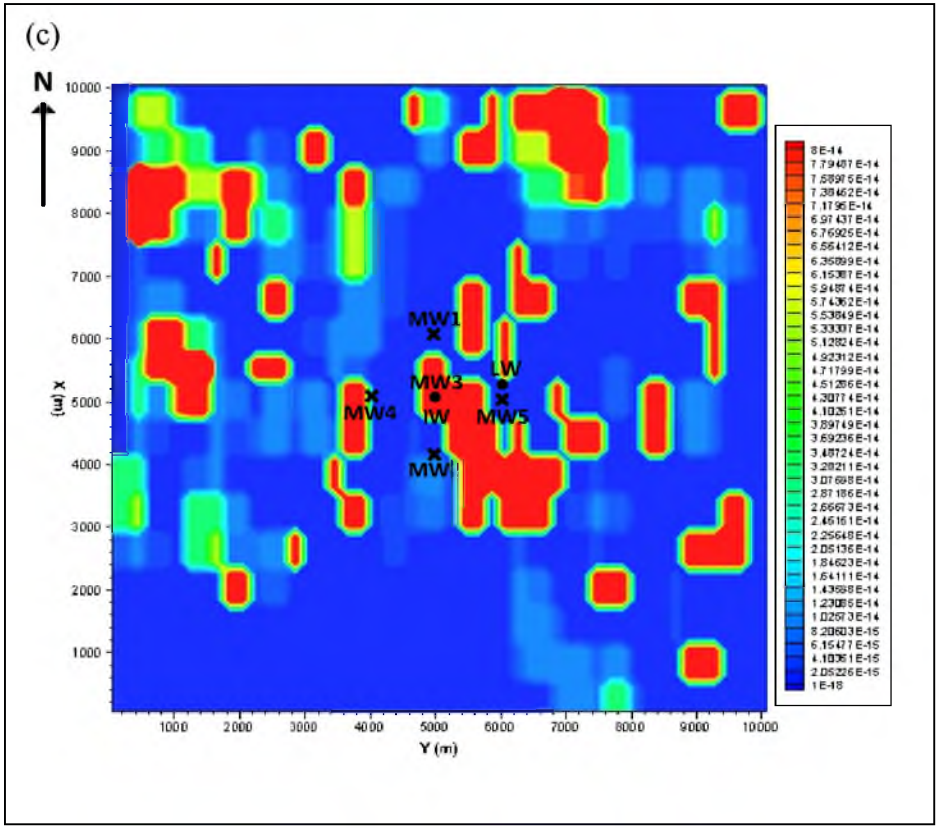


Fig. 3.12 Permeability distributions in the conceptual domain (LW: Leakage well, IW: Injection well, MW: Monitoring well, and IP: Injection point): (a) permeability distribution at external surface, (b) 2nd layer permeability distribution in the overlying formation, (c) 10th layer permeability distribution in the storage formation and (d) permeability distribution in the YZ-plane of the injection well.


(c)
N
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
X (m)
MW1
LW
MW3
MW4
IW
MW5
MW2
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Y (m)

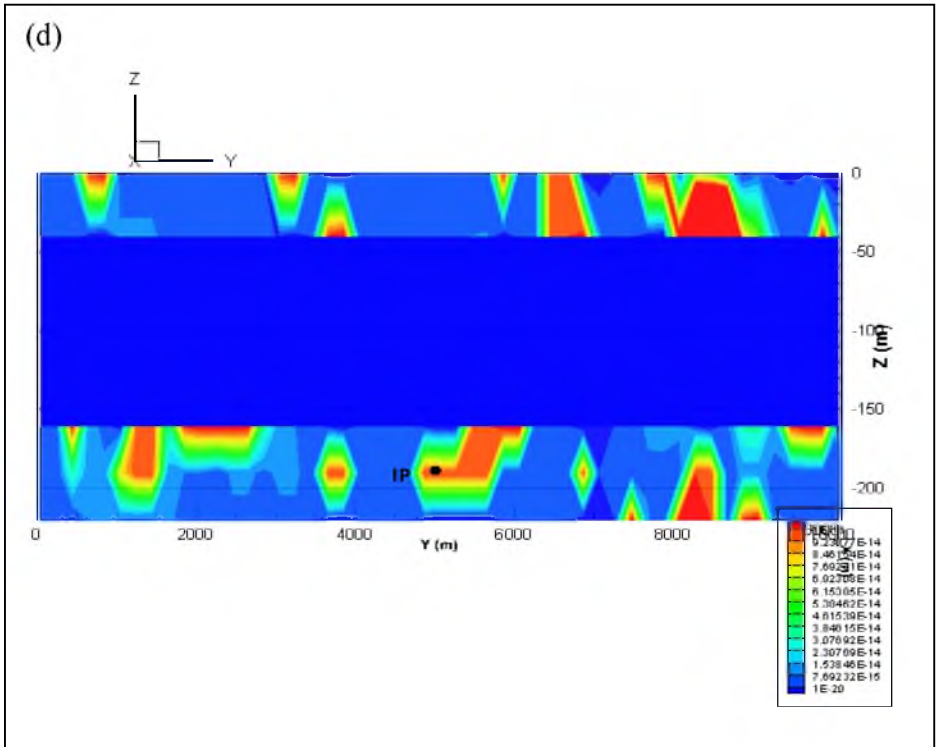


Fig. 3.12 Continued.

The SACROC unit is located in the southeastern segment of the Horseshoe Atoll within the Midland basin, western Texas. Over 93 million metric tons of CO_2 were injected for the purpose of enhanced oil recovery since 1972 (Han et al., 2010). Han et al. (2010) acquired a three-dimensional high-resolution geocellular model from the Texas Bureau of Economic Geology, including a detailed characterization of heterogeneity. The permeability data set in the SACROC unit, Han et al. (2010) used, was assigned to both the storage and the overlying formations and the cap rock was assigned a homogeneous permeability of 10^{-20} m^2 in the generic domain of Fig. 3.1.

3.5 Forward Simulation Results for the Heterogeneous Model

The simulation conditions for the heterogeneous model are the same as those of the homogeneous model except for CO_2 injection rate. In the homogeneous simulations, the total amount of CO_2 injection was about 20 million tons over 10 years at one injection well. This huge amount was considered to induce migration of CO_2 to and through the simulated leakage pathway. However, in the heterogeneous model the CO_2 injection rate was reduced to approximately 2 million tons over 10 years. That condition is more practical (realistic) and also improved model convergence. Table 3.6 presents the CO_2 injection details for the heterogeneous simulations.

Fig. 3.13 illustrates simulated CO_2 saturations in the YZ-plane of the leakage pathway in the heterogeneous system. Gaseous CO_2 injected into the storage formation migrates through higher permeability zones around the CO_2 injection point (refer to Fig. 3.12 (d)) with time. In Fig. 3.13, however, it can be inferred that CO_2 does not pass through the leakage pathway until the end of the simulation (10 years).

Table 3.6 CO_2 injection rate for heterogeneous simulation.

IW Location	CO_2 injection condition	
	Injection time (sec)	Injected mass (kg/s)
(5050 m, 5050 m, -190 m)	0	6.34
	3.16e8	6.34

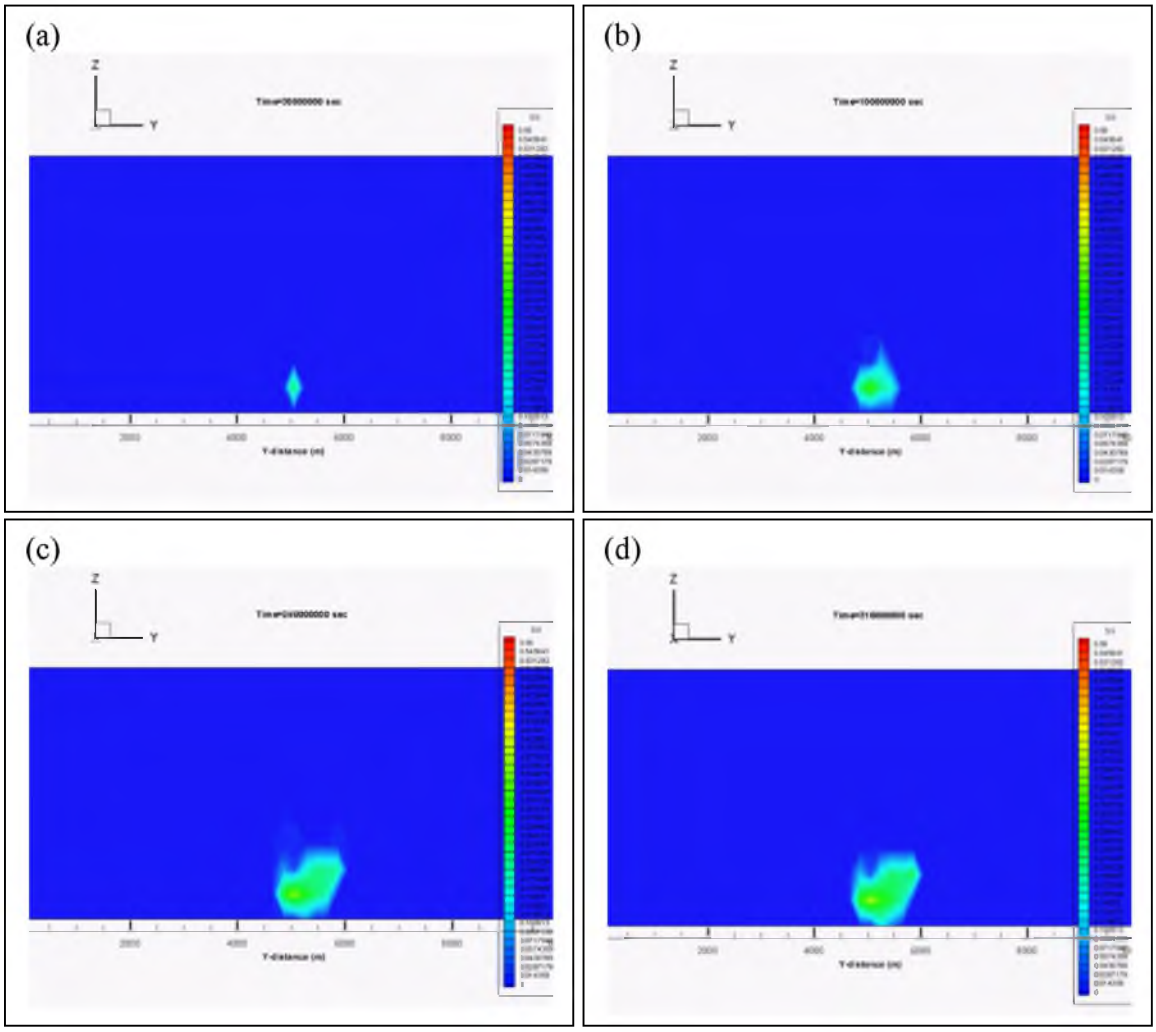


Fig. 3.13 Simulated CO2 saturations in the YZ-plane of the leakage pathway in the heterogeneous system, (a) after 0.95 years, (b) after 3.17 years, (c) after 7.61 years and (d) after 10 years (SG scale: 0- 0.56).

Fig. 3.14 presents the discharge rates of CO_2 and brine at the top of the leakage pathway in the cap rock. Fig. 3.15 depicts pressure perturbations at four points between the top and the bottom of leakage pathway in the cap rock. A discontinuity in fluid pressure due to the capillary pressure is not invoked. As shown in Fig. 3.14, the outflow of brine lasts for the entire simulation period, but CO_2 does not discharge from the leakage pathway.

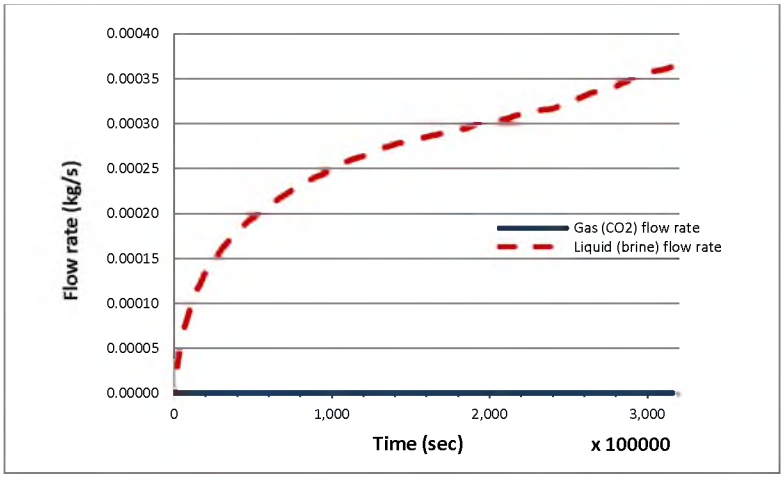


Fig. 3.14 Simulated CO2/brine discharge rates at the top of the leakage pathway.

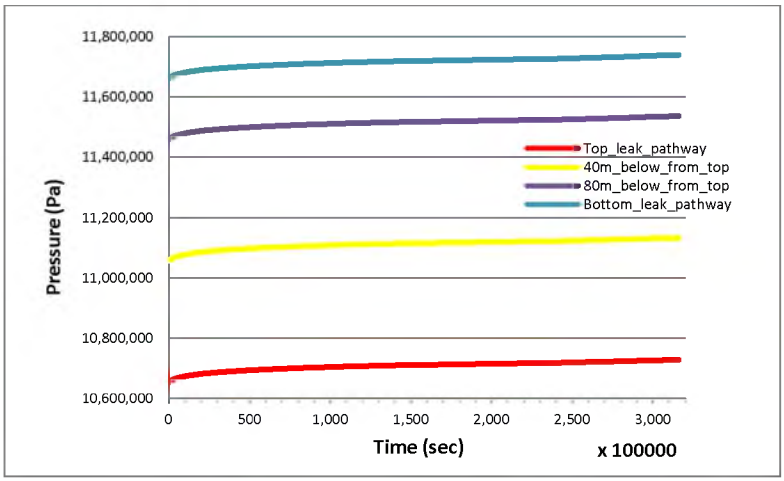


Fig. 3.15 Simulated pressure perturbations at four points of the leakage pathway.

Thus, in the heterogeneous system (with immiscible flow) the pressure anomalies in the leakage pathway are induced by only brine discharge. Capillary pressure through the leakage pathway does not impact pressure anomalies in the leakage pathway (because capillary pressures occur across an interface between two immiscible fluids).

Fig. 3.16 presents pressure perturbations in the YZ-plane of the leakage pathway. The increased pressure gradient by CO_2 injection causes brine discharge into the overlying formation through the leakage pathway, so pressure buildup is not only in the storage formation, but also in both the leakage pathway and the overlying formation.

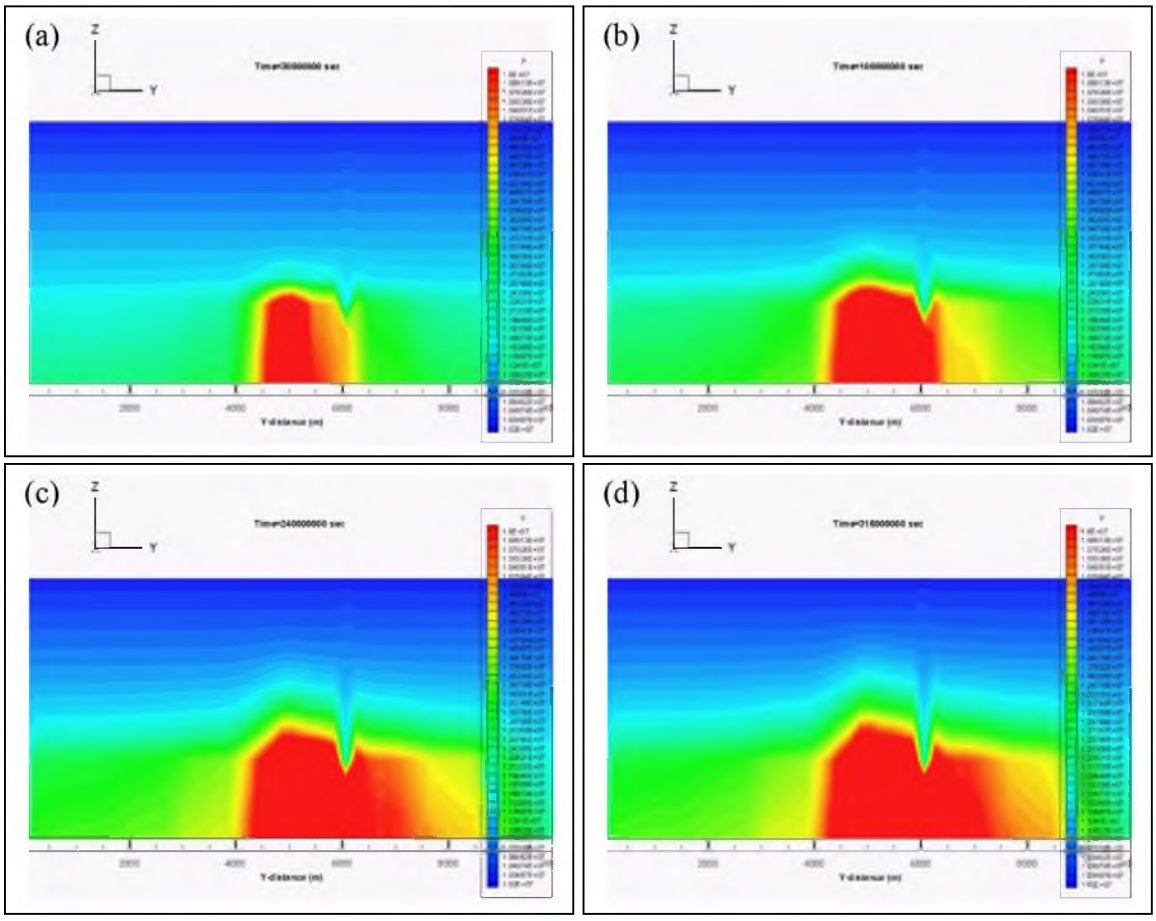


Fig. 3.16 Simulated pressure propagations in the YZ-plane of the leakage pathway, (a) after 0.95 years, (b) after 3.17 years, (c) after 7.61 years and (d) after 10 years (Pressure scale: 10.2 MPa – 16.0 MPa).

On the other hand, pressure increases exhibited in the vicinity of the bottom of the leakage pathway are relatively reduced because of outflows of brine into the leakage pathway. In the homogeneous model results, pressure propagated based on past experience with constant velocity in all directions (see section 3.3). In the heterogeneous system, however, simulated pressure propagation can be different in different directions. The pressures are further propagated into higher permeability zones by migrations of fluids. For reference, pressure propagation length and time can also be identified by an analytical relationship between decay length and decay time of a pressure pulse (Deming, 1994), as follows:

$$l \approx \sqrt{(\alpha \cdot t)} \tag{3.4}$$

$$t \approx l^2/\alpha \tag{3.5}$$

where l: decay length of pressure pulse, t: pressure decay time, and α: hydraulic diffusivity ($=K/Ss$, where K: hydraulic conductivity and Ss: specific storage).

Fig. 3.16 (c) and (d) might create an illusion that pressures in the leakage pathway decrease or the leaks in the leakage pathway flow up to down. However, the pressures in the leakage pathway continue to buildup due to brine leakage with time, as shown in Fig. 3.15, and the leakage consistently flows from the bottom to the top of the leakage pathway (Fig. 3.14). The illusion is an artifact of the visualization (contouring) software. The magnitude of pressure buildup at the bottom of the cap rock (with lower permeability, 10^{-20} m^2) is substantially larger even if the length of pressure propagations is relatively shorter over the simulation time; pressure buildup in the leakage pathway with higher permeability (10^{-10} m^2) is relatively decreased but the length of propagations is relatively longer.

Fig. 3.17 illustrates the development of pressure propagation in the YZ-plane of the 2nd layer of overlying formation. Fig. 3.18 represents the pressure difference between (1) a model with a leakage pathway and (2) a model without a leakage pathway, in the YZ-plane of the 2nd layer of the overlying formation. The pressure anomalies induced by brine leakage are transmitted to the vertical direction (X-axis direction) in the figure. We can infer that the overlying formation is likely to have higher permeability zones or networks in the X-axis direction (refer to Fig. 3.12 (b)).

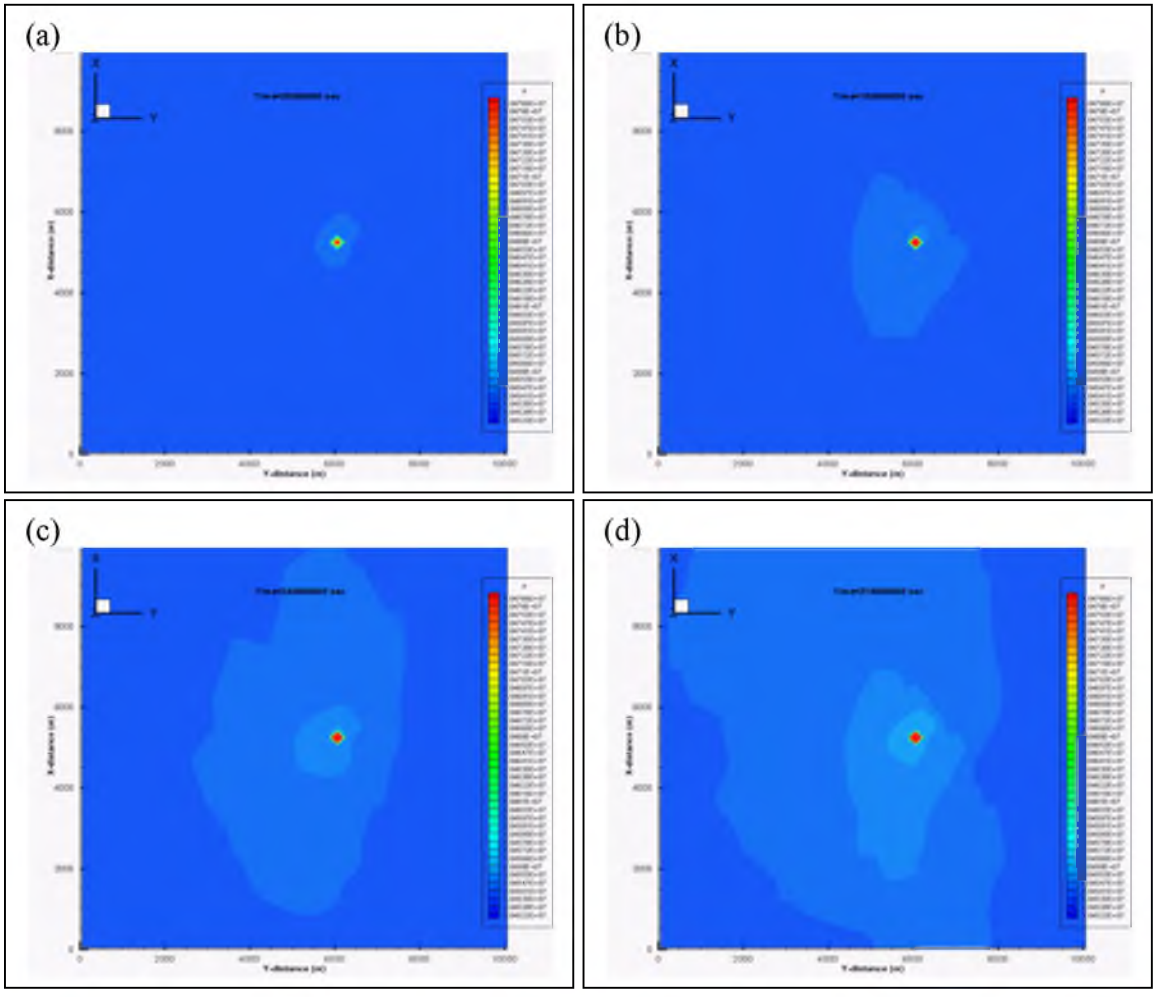


Fig. 3.17 Simulated pressure propagations in the XY-plane (2nd layer) of the overlying formation, (a) after 0.95 years, (b) after 3.17 years, (c) after 7.61 years and (d) after 10 years (Pressure scale: 10.4522 MPa – 10.4766 MPa).

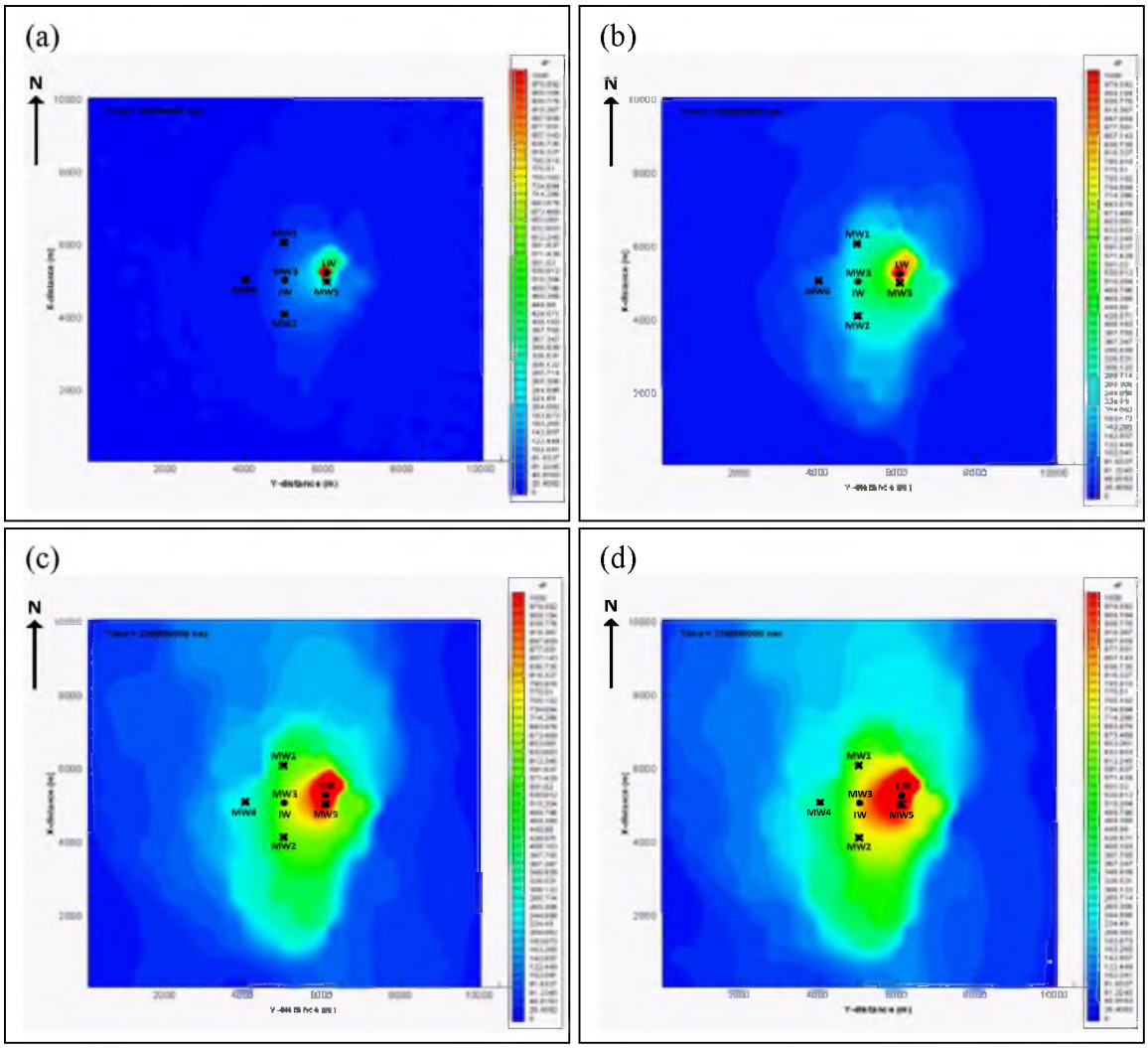


Fig. 3.18 Simulated pressure differences in the XY-plane (2nd layer) of the overlying formation (*dP* scale: 0 - 1,000 Pa): (a) after 0.95 years, (b) after 3.17 years, (c) after 7.61 years and (d) after 10 years (*dP* scale: 0 – 1000 Pa).

Fig. 3.19 illustrates pressure signals at monitoring "wells" in the overlying formation. The pressure profiles in the overlying formation do not exhibit rapid changes in slopes. This is a different result than that of the homogeneous simulation (see Fig. 3.11). As mentioned earlier, capillary pressure does not influence pressure anomalies in the leakage pathway, so the discontinuity in pressure induced by capillary effects is not transmitted to the pressure distribution in the overlying formation.

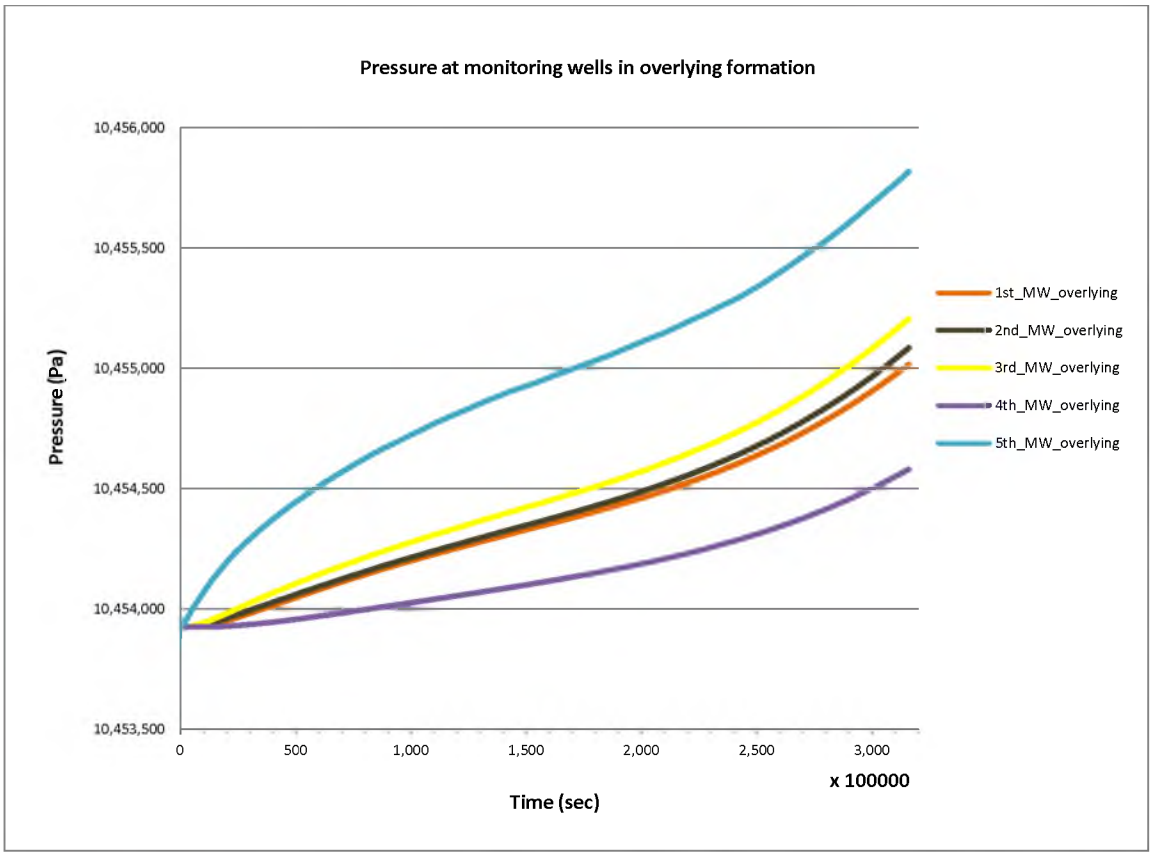


Fig. 3.19 Simulated pressure anomalies at monitoring wells in the overlying formation.

Another result to note is that the brine leakage generates faster pressure buildup at MW2 than MW1 even though LW is more closely located to MW1 (see Fig. 3.12). In the case of the homogeneous model, the pressure buildup at MW1 progresses faster and becomes larger (see Fig. 3.11). On the contrary, in the case of the heterogeneous model, the pressure buildup more rapidly reaches MW2. The heterogeneous permeability distribution can be attributed to the anomalous pressure propagation in the system. In the inverse analysis, that kind of pressure anomalies through highly heterogeneous media might make it difficult to estimate leakage pathways because of the uncertainty of permeability distributions.

3.6 Summary and Conclusion

The sensitivity analysis identified the effect of hydrogeological properties on the pressure signals at monitoring wells, and forward simulations were performed to realize CO_2 or brine leaks in homogeneous and heterogeneous conceptual domains.

The sensitivity of measurements with respect to the hydrogeological properties will be closely related to the accuracy of the inverse solutions estimated through measured pressure anomalies induced by leaks. The pressure anomalies are subject to the flow rates of CO_2 or brine through leakage pathways based on the effective permeability and cross-sectional area of the leakage pathways. On the other hand, the migrations of CO_2 or brine through the cap rock without the leakage pathways can damp the pressure anomalies in the overlying formation. That can affect the purity of pressure perturbations due to CO_2 or brine leaks through the leakage pathway in the cap rock, decreasing the detectability of leakage pathways by inverse analysis. The sensitivity analysis focused on the influences of three parameters: the permeability of the leakage pathway, the permeability of the overlying formation, and the thickness of cap rock. The difference of pressure (dP) is calculated from pressure in the overlying formation between with the leakage pathway and without the leakage pathway in the domain of Fig. 3.1. The results of the sensitivity analysis in terms of the detectability of leakage pathways by inversion are as follows:

(1) Leakage pathway permeability has a large impact on the magnitude of dP in the overlying formation because the leakage rates are subject to the permeability of the leakage pathway into the overlying formation. The permeability of the leakage pathway has to be higher than 10^{-17} m^2 to induce pressure anomalies by

CO_2 or brine leaks through the leakage pathway in the system.

(2) The permeability of the overlying formation influences the magnitude of *dP* in the overlying formation. In the case of the lower permeability (10^{-15} m^2), the monitoring wells can detect larger significant pressure anomalies and area induced by leaks through the leakage pathway than that of the higher permeability (10^{-13} m^2) because the lower permeability enlarges the magnitude of pressures. The lower permeability of the overlying formation can increase the possibility of leakage detection through inverse simulation using pressure anomalies in the system.

(3) Cap rock thickness also affects the magnitude of *dP* in the overlying formation. The reason is that it can contribute to reduce diffuse leakage through the cap rock without the leakage pathway. As a result, thicker cap rock is able to magnify pure pressure anomalies due to CO_2 or brine leaks through leakage pathways. If the overlying formation has higher permeability ($k = 10^{-13}$ m^2), the cap rock thickness might be at least over 100 m with respect to increasing detectability of inverse simulation in the system.

The conceptual domain with a homogeneous distribution was applied to model the effects of migrations of CO_2 and brine through the leakage pathway. In the modeling scenario, 20 million tons of CO_2 is injected for 10 years into the storage formation and the injected CO_2 evolves gas phase saturation. The increased pressure gradient by CO_2 injection continuously induces brine discharges through the leakage pathway before the CO_2 leaks. When CO_2 reaches the bottom of the leakage pathway after approximately 3.5 years, it is rapidly transported into the overlying formation through the leakage pathway

by pressurization and buoyancy effects. CO_2 migration along a leakage pathway saturated by brine induces capillary effects. The capillary effects at the leakage pathway are propagated into the whole overlying aquifer. Each pressure profile at the five monitoring points in the overlying aquifer has a sudden change in pressure gradient due to capillary effects at the leakage pathway. It has a significant effect on MW5 in the overlying aquifer, which is the closest monitoring well to the leakage pathway.

The heterogeneous modeling was used to determine the migration of CO_2/brine leaks and pressure anomalies induced by the leaks in the overlying formation of the heterogeneous field. The heterogeneity from the SACROC unit was introduced into the conceptual domain. In the heterogeneous simulation, the CO_2 injection rate is approximately 2 million tons for 10 years. The outflow of brine through the leakage pathway lasts for the simulation period but CO_2 does not leak through the leakage pathway. Thus in the overlying formation, the pressure anomalies are induced by only brine discharge and the discontinuity in fluid pressure due to capillary effects is not distinct from that of the homogeneous simulation. In the heterogeneous system, the magnitude and travel time of pressure anomalies induced by CO_2 and brine leaks can be various. The pressures are faster and further propagated into the higher permeability zone by an increased pressure gradient due to migrations of fluids. The variable pressure anomalies may make it difficult to estimate the leakage pathways because of the uncertainty of permeability distributions.

3.7 References

Altevogt, A. S., and Celia, M. A. (2004). "Numerical modeling of carbon dioxide in unsaturated soils due to deep subsurface leakage." *Water Resources Research*, 40(3).

Altunin, V. (1975). *Thermophysical properties of carbon dioxide*, Publishing House of Standards, Moscow, Russia.

Anderson, M. P., and Woessner, W. W. (1992). *Applied groundwater modeling: simulation of flow and advective transport*, Elsevier Science, Oxford, UK.

Babbar, M., and Minsker, B. (2006). "Groundwater remediation design using multiscale genetic algorithms." *Journal of Water Resources Planning and Management*, 132(5), 341-350.

Beckford, O., Hilton, A. C., and Liu, X. "Development of an enhanced multi-objective robust genetic algorithm for groundwater remediation design under uncertainty." *Proc., Proc. Am. Soc. Civ. Eng*, 98.

Benjamin, J. R. (1970). *Probability, statistics, and decision for civil engineers*, McGraw-Hill, New York, NY.

Bennett, G. D. (1976). *Introduction to groundwater hydraulics*, U.S. Geological Survey, Denver, CO.

Bevington, P., and Robinson, D. (1969). *Data reduction and data analysis for the physical sciences*, McGraw-Hill, New York, NY.

Carroll, S., Hao, Y., and Aines, R. (2009). "Transport and detection of carbon dioxide in dilute aquifers." *Energy Procedia*, 1(1), 2111-2118.

Chan-Hilton, A. B., Iyer, S. K., Magar, V., and Kelley, M. "Optimization of natural attenuation with active remediation under uncertainty." *Proc., Seventh International In Situ and On-Site Bioremediation Symposium, Orlando, Florida, USA, 2-5 June 2003. Part H. Natural Attenuation, Long-Term Monitoring, and Site Closure*, Battelle Press, Columbus, OH.

Chen, C. S. (1989). "Solutions approximating solute transport in a leaky aquifer receiving wastewater injection." *Water Resources Research*, 25(1), 61-72.

Chen, Z., Huan, G., and Ma, Y. (2006). *Computational methods for multiphase flows in porous media*, Siam, Philadelphia, PA.

Christensen, S., and Cooley, R. (1999). "Evaluation of confidence intervals for a steady-state leaky aquifer model." *Advances in Water Resources*, 22(8), 807-817.

Cihan, A., Zhou, Q., and Birkholzer, J. T. (2011). "Analytical solutions for pressure perturbation and fluid leakage through aquitards and wells in multilayered-aquifer systems." *Water Resources Research*, 47(10), W10504.

Cobb, P., McElwee, C., and Butt, M. (1982). "Analysis of leaky aquifer pumping test data: An automated numerical solution using sensitivity analysis." *Groundwater*, 20(3), 325-333.

Committee, I. F. (1967). "A formulation of the thermodynamic properties of ordinary water substance." *IFC Secretariat, Düsseldorf, Germany*, 26.

Corey, A. T. (1954). "The interrelation between gas and oil relative permeabilities." *Producers Monthly*, 19(1), 38-41.

Deming, D. (1994). "Factors necessary to define a pressure seal." *AAPG bulletin*, 78(6).

Doughty, C., and Pruess, K. (2004). "Modeling supercritical carbon dioxide injection in heterogeneous porous media." *Vadose Zone Journal*, 3(3), 837-847.

Espinoza, F. P., Minsker, B. S., and Goldberg, D. E. (2005). "Adaptive hybrid genetic algorithm for groundwater remediation design." *Journal of Water Resources Planning and Management*, 131(1), 14-24.

Finsterle, S. (2004). "Multiphase inverse modeling." *Vadose Zone Journal*, 3(3), 747-762.

Finsterle, S. (2007a). "iTOUGH2 user's guide." *Report LBNL-40040*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007b). "iTOUGH2 command reference." *Report LBNL-40041*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007c). "iTOUGH2 sample problems." *Report LBNL-40042*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2010). "iTOUGH2 V3. 2, verification and validation report." *Report LBNL-42002*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S., Moridis, G., and Pruess, K. (1994). "A tough2 equation-of-state module for the simulation of two-phase flow of air, water, and a miscible gelling liquid." *Report LBL-36086*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Garcia, J. E. (2001). "Density of aqueous solutions of CO2." *Report LBNL-49023*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Gasda, S. E., Bachu, S., and Celia, M. A. (2004). "Spatial characterization of the location of potentially leaky wells penetrating a deep saline aquifer in a mature sedimentary basin." *Environmental Geology*, 46(6-7), 707-720.

Gasda, S. E., Wang, J. Z., and Celia, M. A. (2011). "Analysis of in-situ wellbore integrity data for existing wells with long-term exposure to CO2." *Energy Procedia*, 4, 5406-5413.

Han, W. S., McPherson, B. J., Lichtner, P. C., and Wang, F. P. (2010). "Evaluation of trapping mechanisms in geologic CO2 sequestration: Case study of SACROC northern platform, a 35-year CO2 injection site." *American Journal of Science*, 310(4), 282-324.

Hou, Z., Murray, J. C., and Rockhold, L. M. (2012). "CO2 migration in intact caprock and leakage risk in three-dimensional heterogeneous formations." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittssburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012a). "Early detection of brine or CO2 leakage through high-permeability pathways using pressure-based monitoring data." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittsburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012b). "Impact of data uncertainty on identifying leakage pathways in CO2 geologic storage systems and estimating their hydrogeological properties by inverse modeling." *TOUGH Symposium 2012,* Lawrence Berkeley National Laboratory, Berkeley, CA.

Ko, N.-Y., and Lee, K.-K. (2008). "Reliability and remediation cost of optimal remediation design considering uncertainty in aquifer parameters." *Journal of Water Resources Planning and Management*, 134(5), 413-421.

Krevor, S., Perrin, J.-C., Esposito, A., Rella, C., and Benson, S. (2010). "Rapid detection and characterization of surface CO2 leakage through the real-time measurement of δ13 C signatures in CO2 flux from the ground." *International Journal of Greenhouse Gas Control*, 4(5), 811-815.

Liggett, J. A., and Chen, L.-C. (1994). "Inverse transient analysis in pipe networks." *Journal of Hydraulic Engineering*, 120(8), 934-955.

Metz, B., Davidson, O., de Coninck, H., Loos, M., and Meyer, L. (2005). *IPCC special report on carbon dioxide capture and storage*, Cambridge University Press, New York, NY.

Nogues, J. P., Nordbotten, J. M., and Celia, M. A. (2011). "Detecting leakage of brine or CO2 through abandoned wells in a geological sequestration operation using pressure monitoring wells." *Energy Procedia*, 4, 3620-3627.

Nordbotten, J. M., Celia, M. A., and Bachu, S. (2004). "Analytical solutions for leakage rates through abandoned wells." *Water Resources Research*, 40(4), W04204.

Nordbotten, J. M., Kavetski, D., Celia, M. A., and Bachu, S. (2008). "Model for CO2 leakage including multiple geological layers and multiple leaky wells." *Environmental Science & Technology*, 43(3), 743-749.

Onuma, T., and Ohkawa, S. (2009). "Detection of surface deformation related with CO2 injection by DInSAR at In Salah, Algeria." *Energy Procedia*, 1(1), 2177-2184.

Phillips, S. L., Igbene, A., Fair, J., Ozbek, H., and Tavana, M. (1981). *A technical databook for geothermal energy utilization*, Lawrence Berkeley Laboratory, University of California, Berkeley, CA.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in Fortran 77: The art of scientific computing second edition*, Cambridge University Press, New York, NY.

Pruess, K. (1987). "TOUGH user's guide, nuclear regulatory commission report NUREG/CR-4645." *Report LBL-20700*, Lawrence Berkeley Laboratory Berkeley, CA.

Pruess, K. (1991). "TOUGH2: A general-purpose numerical simulator for multiphase fluid and heat flow." *Report LBL-29400*, Lawrence Berkeley Laboratory, Berkeley, CA.

Pruess, K. (2004). "Numerical simulation of CO2 leakage from a geologic disposal reservoir, including transitions from super-to subcritical conditions, and boiling of liquid CO2." *Spe Journal*, 9(2), 237-248.

Pruess, K. (2005). "ECO2N: A TOUGH2 fluid property module for mixtures of water, NaCl, and CO2." *Report LBNL-57952*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Pruess, K., and García, J. (2002). "Multiphase flow dynamics during CO2 disposal into saline aquifers." *Environmental Geology*, 42(2-3), 282-295.

Pruess, K., Moridis, G., and Oldenburg, C. (1999). "TOUGH2 user's guide, version 2.0." *Report LBNL-43134*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Rao, S. S. (2009). *Engineering optimization: Theory and practice*, John Wiley & Sons, Hoboken, New Jersey.

Singh, S. K. (2009). "Simple method for quick estimation of leaky-aquifer parameters." *Journal of Irrigation and Drainage Engineering*, 136(2), 149-153.

Spycher, N., and Pruess, K. (2005). "CO2-H2O mixtures in the geological sequestration of CO2. II. Partitioning in chloride brines at 12–100° C and up to 600 bar." *Geochimica et Cosmochimica Acta*, 69(13), 3309-3320.

Sun, A. Y., Zeidouni, M., Nicot, J.-P., Lu, Z., and Zhang, D. (2013). "Assessing leakage detectability at geologic CO2 sequestration sites using the probabilistic collocation method." *Advances in Water Resources*, 56, 49-60.

Van Genuchten, M. T. (1980). "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils." *Soil Science Society of America Journal*, 44(5), 892-898.

Zheng, C., Bennett, G. D., Melton, J., and Simon, A. R. (2002). "Applied contaminant transport modeling." *Industrial and Commercial Training*, 34(7), 256-262.

Zhou, Q., Birkholzer, J. T., and Tsang, C.-F. (2009). "A semi-analytical solution for large-scale injection-induced pressure perturbation and leakage in a laterally bounded aquifer–aquitard system." *Transport in Porous Media*, 78(1), 127-148.

CHAPTER 4

INVERSE SIMULATION USING ITOUGH2

Parameter estimation by inversion (using the iTOUGH2 code) is applied to detect locations of leakage pathways by calibrating the absolute permeability of initial guesses of leakage pathways in homogeneous and heterogeneous conceptual domains as discussed in Chapter 3. This chapter focuses on similar objectives, but through inverse simulation. In CCUS, early detection is very important to provide an early warning. If not detected early on, CO_2 leaks may reach freshwater aquifers or the land surface. To reduce CO_2 leakage risk, early leakage detection should be applied for leakage pathway estimation (Jung et al., 2012b).

Hydrogeological parameters measured from some techniques may include errors. In this context, error means a deviation between simulation results and exact solutions. In general, there are two types of errors: (1) systematic errors, which are predictable errors from measuring devices or observers' bias, and (2) random errors, unpredictable errors like noises in measurement data when a measurement is repeated. It is practically impossible to obtain exact values of model parameters from the real world because the errors from some techniques can never be removed (Finsterle, 2007a). The uncertainties (systematic errors) of parameters can influence calculated pressures and outcomes of forward modeling. Uncertainties may decrease the accuracy of results in inverse

simulations, i.e., calibrating the permeability of leakage pathways, because the parameters that exhibit these errors are given as "known values" in the inverse modeling. Therefore, the impact of parameter uncertainties has to be examined, and associated uncertainties should be estimated through the inverse modeling to improve accuracy of estimating locations of leakage pathways (Jung et al., 2012b).

Input parameter uncertainty is associated with absolute permeability, porosity, relative permeability, and capillary pressure. Uncertainties are inherent to information on the geologic and hydrologic boundaries, and the thicknesses of geologic layers (Finsterle, 2007a). However, it is impossible to examine or estimate all uncertainties of those parameters through the inverse modeling. In practice, a priori investigation of those parameters should be performed to reduce associated error. Jung et al. (2012b) defined that cap rock permeability is the main parameter affecting pressure anomalies (by leaks). On the other hand, this study, and this chapter in particular, focuses on reducing the effects of uncertainty of permeability distribution of reservoirs, one of the most important factors in understanding the ability of a reservoir to transmit fluids. Using homogeneous models, the inverse analysis identifies impact of the uncertainty of permeability of overlying formation as a systematic error. To reduce its impact, the inversion simultaneously calibrates the permeability of overlying formation during the estimation of the leakage pathway. Inverse modeling of permeability heterogeneity examines the effect of systematic error associated with renormalization (upscaling) and estimates the renormalized heterogeneous permeability. All simulation periods are 10 years, to represent early project stages. The Levenberg-Marquardt method, which is known for performing well for the inverse modeling in subsurface fields, is applied in this study.

The inverse modeling using iTOUGH2 is conducted based on the procedures described in Table 2.7.

4.1 Leakage Detection in the Homogeneous Domain

Jung et al. (2012b) estimated the location of a leakage well by calibrating the permeability of that well through inverse analysis of pressure data, limited to homogeneous aquifers with single-phase conditions. This section (4.1) examines inverse analysis for estimation of a leakage pathway in homogeneous domain, but with multiphase flow. The same model domain used for forward simulation (Fig. 3.1) is used for this inverse modeling. As shown in Table 3.4, nine pressure observation points "measure" the pressure perturbation data induced by leaks; eight points in both the storage and overlying formations at four monitoring wells and one point in the overlying formation at an injection well. In this inverse analysis, initial "guesses" of locations of leakage pathways are assigned a necessary aspect of the algorithm. Initial guesses are improved by better a priori characterization information, such as locations of abandoned wells or otherwise. The measured pressure observation points are used for calibrating the vertical permeability of each different initial guess of the leakage pathway; note the location of the actual leakage pathway at (x, y) = (5250 m, 6050 m) from the model origin. The pressure data illustrated in Fig. 3.11 are used as the measured pressure data in the overlying formation.

The inverse modeling not only estimates the leakage pathway location through estimating vertical permeability of each initial guess, but may also factor in the estimated error in permeability of the overlying formation. Three scenarios were evaluated for the

modeling. First, inverse modeling of only the location of the actual leakage pathway from the initial guesses based on the exact homogeneous permeability (10^{-15} m^2) of the overlying formation. This scenario can be called the "idealized case." The second case is inverse analysis for only the leakage location based on an assumed error of the permeability field (in this analysis, the assumed degree or error is $10^{-14.5}$ m^2 to $10^{-15.5}$ m^2) of the overlying formation. In the third scenario, both the location of the leakage pathway and the permeability of the overlying formation are estimated. As mentioned in section 2.3.2, because reservoir pressures may vary over two orders of magnitude, weighting factors can be used to scale measurement data. Weighting factors of 1 Pa and 100 Pa were assigned to scale measurements of the overlying and storage formations, respectively, such that pressure in one formation will not obscure pressure of the other formation in the analysis. The given weighting factor values were evaluated from residual analysis, as described in section 4.1.1.

The first inverse modeling ("case 1") is applied to an idealized case here. The inversion estimates only the location of the actual (one) leakage pathway from the initial guesses of the leakage pathway. As mentioned earlier, the leakage pathway location is estimated by calibrating the vertical permeability of each different initial guess (location) of the leakage well.

Fig. 4.1 illustrates a two-dimensional model domain. First of all, possible areas with the presence of a leakage well can be roughly estimated from the travel time of pressure measured in monitoring wells. In Fig. 3.11, the location of a leakage well can be approximated at least for the homogeneous conditions, because the propagation length of pressure is proportional to its travel time.

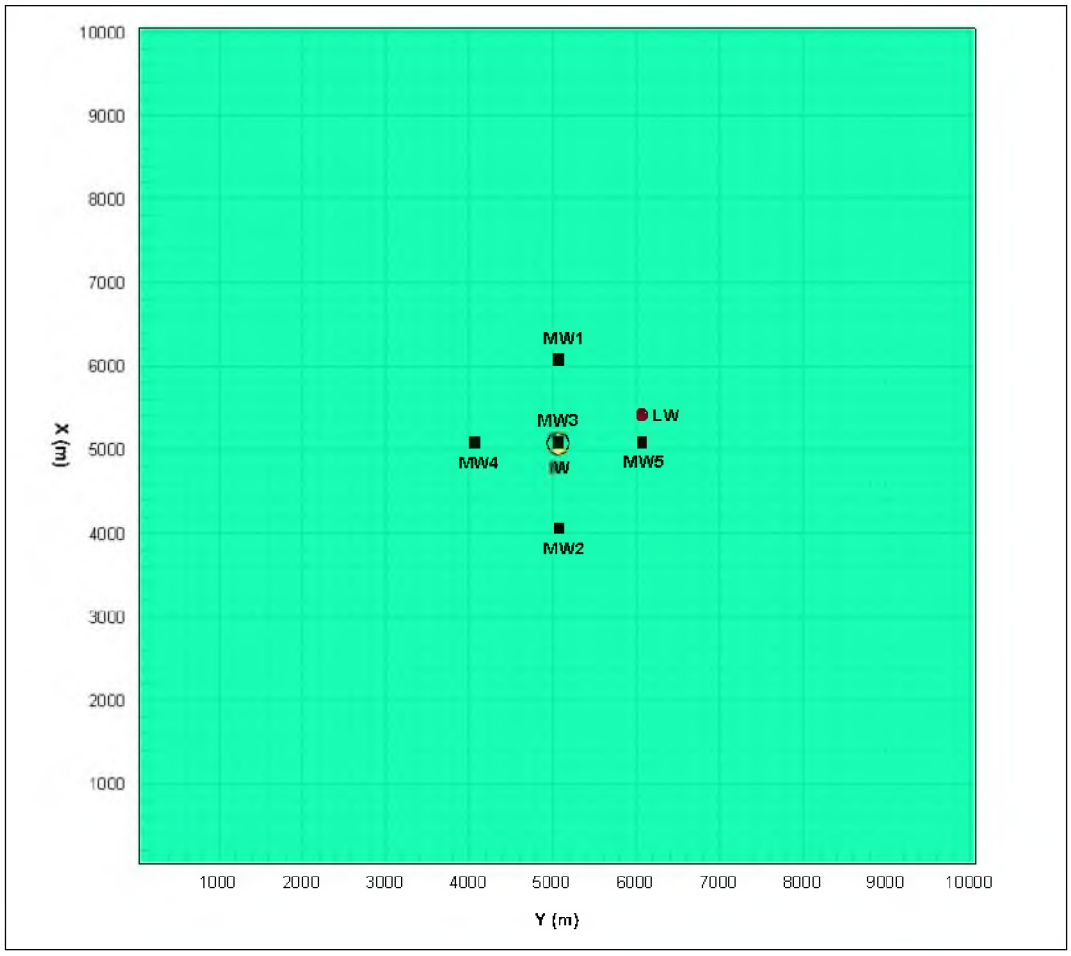


Fig. 4.1 Two-dimensional model domain.

Specifically, the length that a pressure perturbation migrates, l, over a specific time, t, is $l \approx \sqrt{(\alpha \cdot t)}$, where α = hydraulic diffusivity. Thus, with data about relative arrival times of new pressure anomalies, we can expect to approximate an area with the presence of a leakage well. From Fig. 3.11 the pressure perturbations induced by leaks in the overlying formation reach the 5th monitoring well first. Also, the magnitude of the pressure perturbation is largest there, so the leakage well may be closest to the 5th monitoring well. Based on the observation, initial guesses of the leakage pathway are assigned around the 5th monitoring well with more points.

A total of 48 initial guesses of leakage pathways are assigned in the model domain and the logarithm of absolute vertical permeability (k_{lz}) for each initial guess becomes an unknown parameter. The 48 inverse simulations estimate each initial guess.

The inversion should be effectively designed to estimate the optimum leakage well location through the parameter values to minimize the objective function. This inverse modeling approach iteratively runs the forward simulator to generate model output (pressures) so that a minimum number of grid blocks is needed (to reduce computational expense). For effective inverse modeling, the number of grid blocks in the model domain is 21×31×11 (7,161 grid blocks total). Grid blocks were meshed to 0.3 m × 0.3 m according to the specific geometry of the 48 initial guesses. The logarithm of the absolute vertical permeability of each initial guess was iteratively estimated by the inverse model. Fig. 4.2 shows a contour plot of the objective function from the inversion performed to estimate the logarithm of vertical permeability of each initial guess. The shape and convexity of the objective function indicates both uniqueness and whether an inverse analysis is stable (or well-posed). Since the objective function is a sum of squares of residuals, a well-posed inversion displays parabolic with elliptical contour lines in a 3-dimensional plot. In the case of a two-dimensional plot, the objective function near the global minimum exhibits elliptical contour lines (Finsterle, 2007a). Fig. 4.2 shows that the inversion is stable and has a unique solution because the objective function near the global minimum exhibits close to elliptical contour lines and the topography presents only one global minimum. The initial guess with a minimum objective function is considered to be the best estimation, i.e., the most possible location of the leakage well (black filled circle).

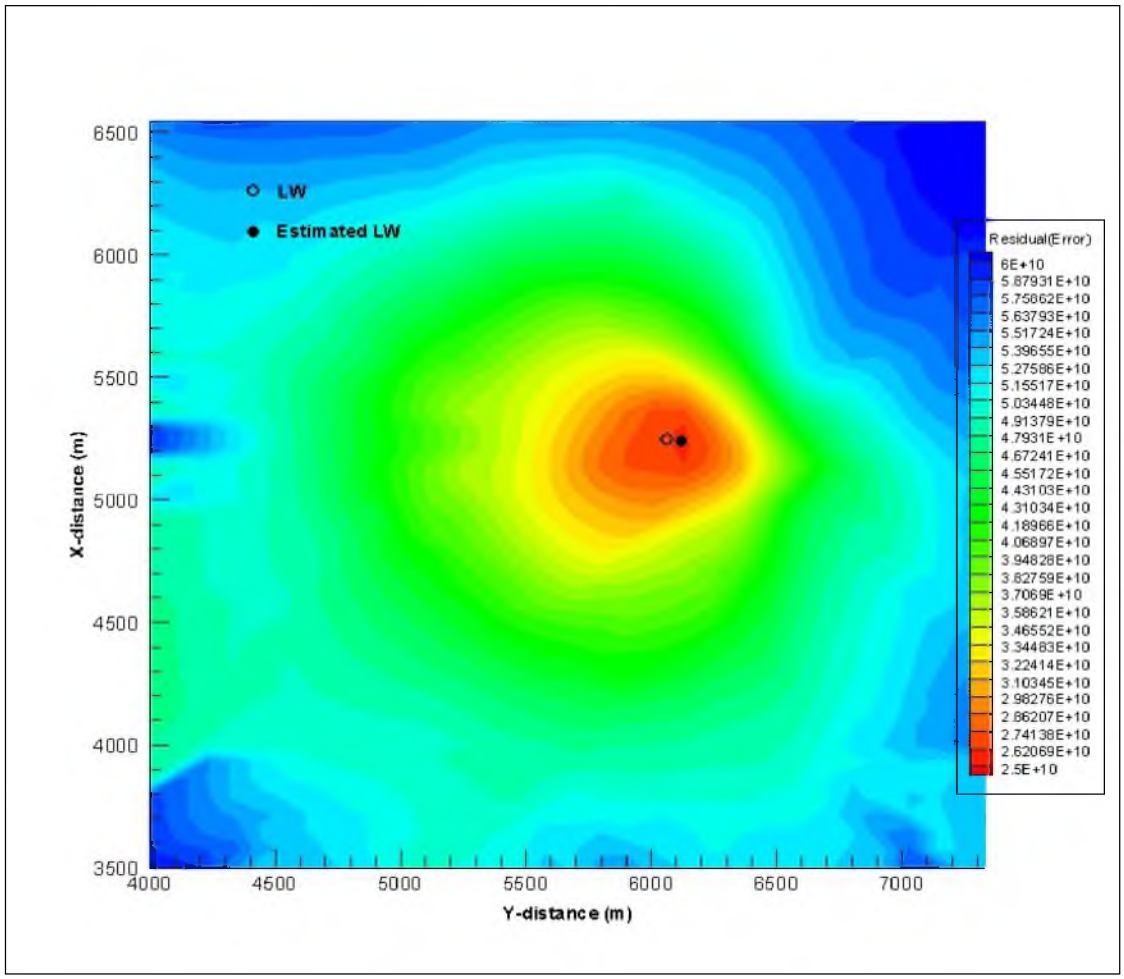


Fig. 4.2 Estimated leakage well location from objective function in case 1.

The minimum objective function value expresses the best fit between the measured and simulated pressures. The inversion estimated a coordinate (x, y) = (5250 m, 6150 m) as the most possible location. The estimated leakage well has an inherent deviation of 100 m against the actual leakage well location (void circle) at (x, y) = (5250 m, 6050 m). Even with the deviation of 100 m, the deviation is not significantly large with respect to the whole system, suggesting that the inversion results are qualitatively good. Table 4.1 denotes the objective function values of some initial guesses estimated from case 1.

Table 4.1 Objective function values in case1.

Initial guess		**Objective function (Pa^2)**
Number	**Coordinate (m)**	
True	**(5250, 6050)**	**-**
5	(4950, 5550)	0.3365e+11
12	(5150, 6050)	0.2687e+11
14	(5150, 6350)	0.2596e+11
15	(5150, 6550)	0.3814e+11
16	(5150, 7050)	0.5042e+11
23 (True)	**(5250, 6050)**	**0.2669e+11**
24 (Best)	**(5250, 6150) (Deviation: 100 m from true)**	**0.2558e+11**
25	(5250, 6350)	0.3112e+11
26	(5250, 6550)	0.4085e+11
31	(5350, 5550)	0.3408e+11
34	(5350, 6050)	0.2634e+11
35	(5350, 6150)	0.2687e+11

The second scenario ("case 2") consists of estimating the leakage location based only on underestimated and overestimated permeability, (a) $10^{-14.5} \approx 3.16 \times 10^{-15}\ m^2$ and (b) $10^{-15.5} \approx 3.16 \times 10^{-16}\ m^2$. The inverse modeling is conducted with the given incorrect permeability of the overlying formation as a known value for the leakage pathway estimation. This simulation is intended to examine the impact of uncertainty of the overlying formation permeability on leakage pathway detection. Fig. 4.3 illustrates a simple uncertainty propagation analysis for the incorrect permeability of the overlying formation.

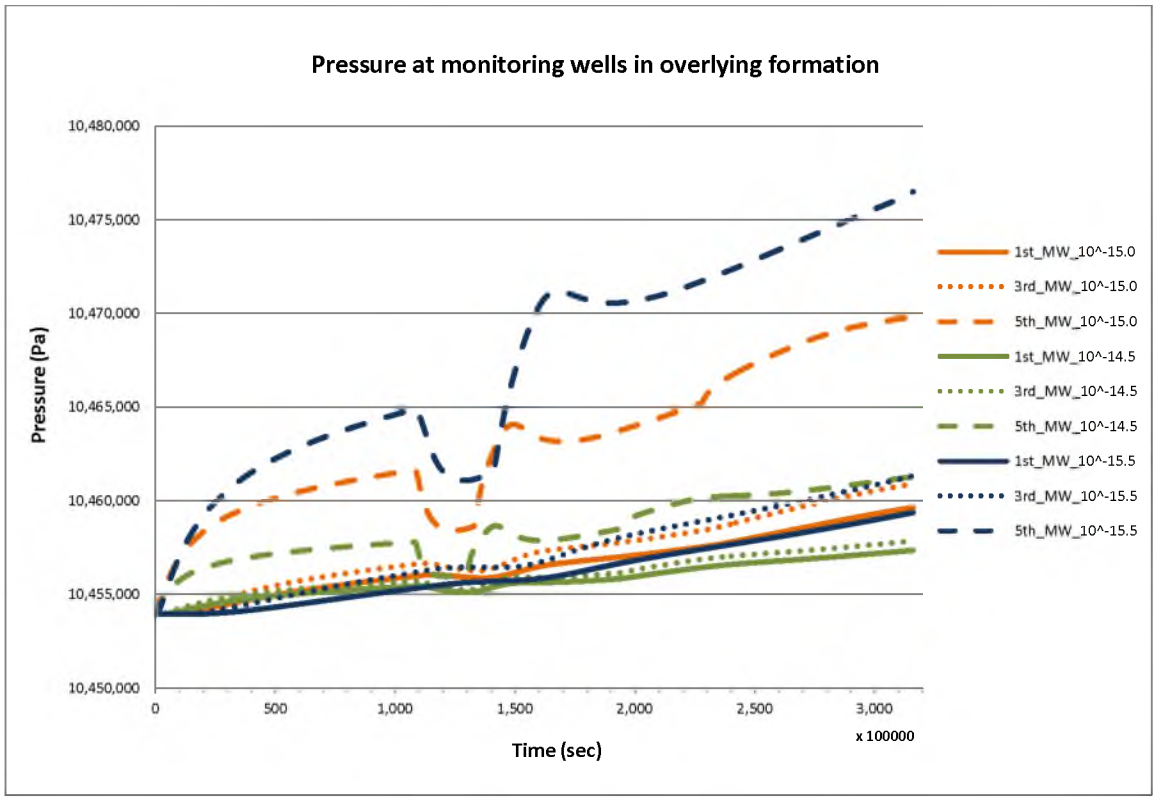


Fig. 4.3 Pressure drifts among 10^{-15}, $10^{-14.5}$ and $10^{-15.5}$ m^2 permeability of the overlying formation.

In Fig. 4.3 the solid, dashed and dotted lines represent pressures at three monitoring points of MW1, MW3 and MW5 in the overlying formation, respectively. The orange line represents the pressure profiles for the exact permeability (10^{-15} m^2), whereas the green and the blue lines describe the pressure signals from the incorrect permeabilities, (a) $10^{-14.5}$ m^2 and (b) $10^{-15.5}$ m^2, respectively, at three monitoring points. Pressures induced with included uncertainty of the overlying formation permeability significantly deviate from the actual pressure solutions. Fig. 4.3 indicates that the wrong information for formation permeability propagates error to calculated pressures at the monitoring wells, and that influences the inversion results. The inversion is conducted with the same methodology as the first simulation case.

Fig. 4.4 presents a contour plot of the objective function from the inversion performed to estimate the logarithm of vertical permeability of each initial guess based on an overestimated permeability $10^{-14.5}$ m^2 (a). In Fig. 4.4 the inversion also exhibits a global minimum and stability. The most likely leakage well location is at (x, y) = (5150 m, 5950 m). The deviation with respect to the actual leakage well is 141 m. Fig. 4.5 represents a contour plot of the objective function for the inversion with an underestimated permeability of $10^{-15.5}$ m^2 (b).

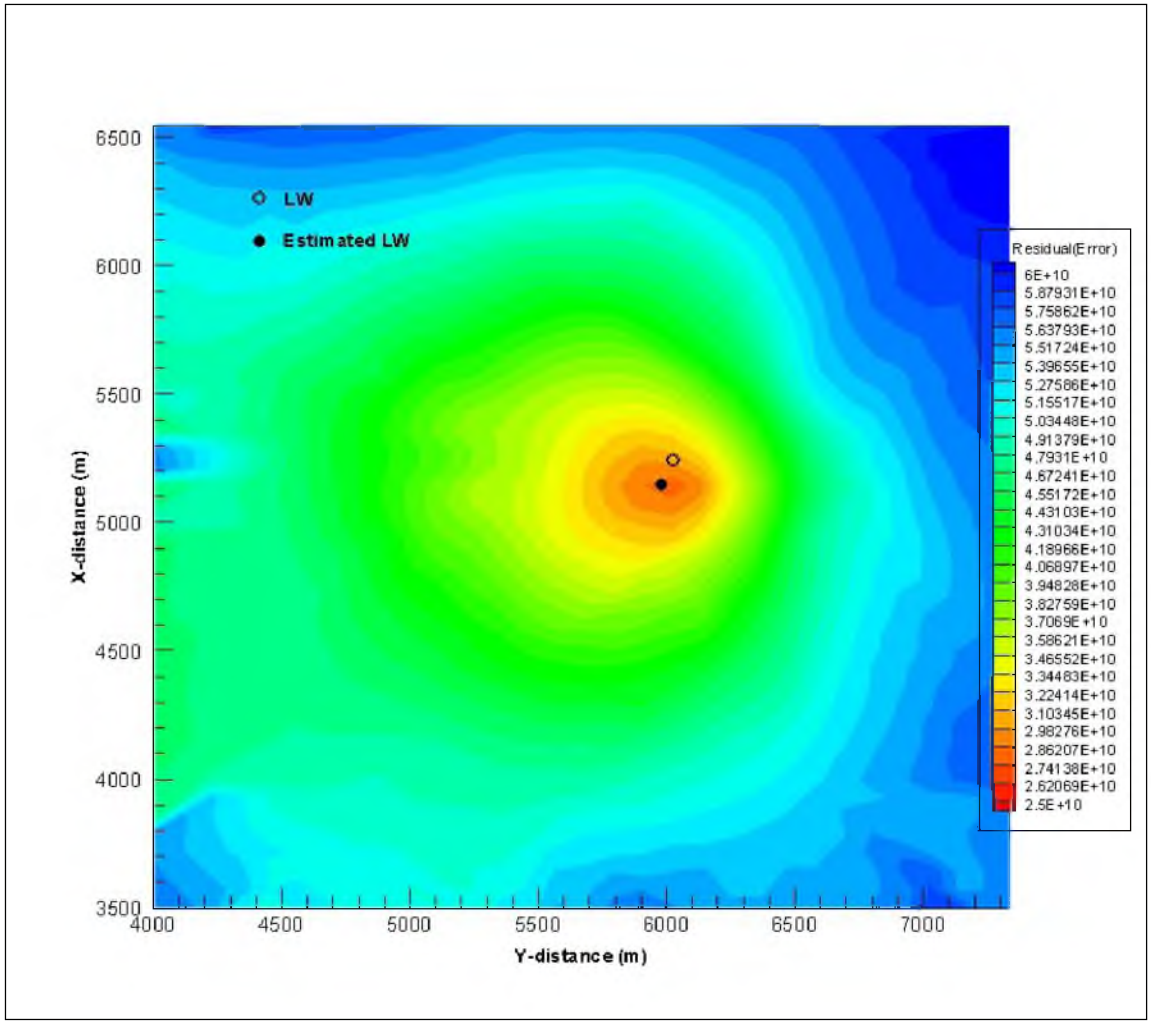


Fig. 4.4 Estimated leakage well location based on the objective function for the simulation with overestimated permeability $10^{-14.5}$ m^2.

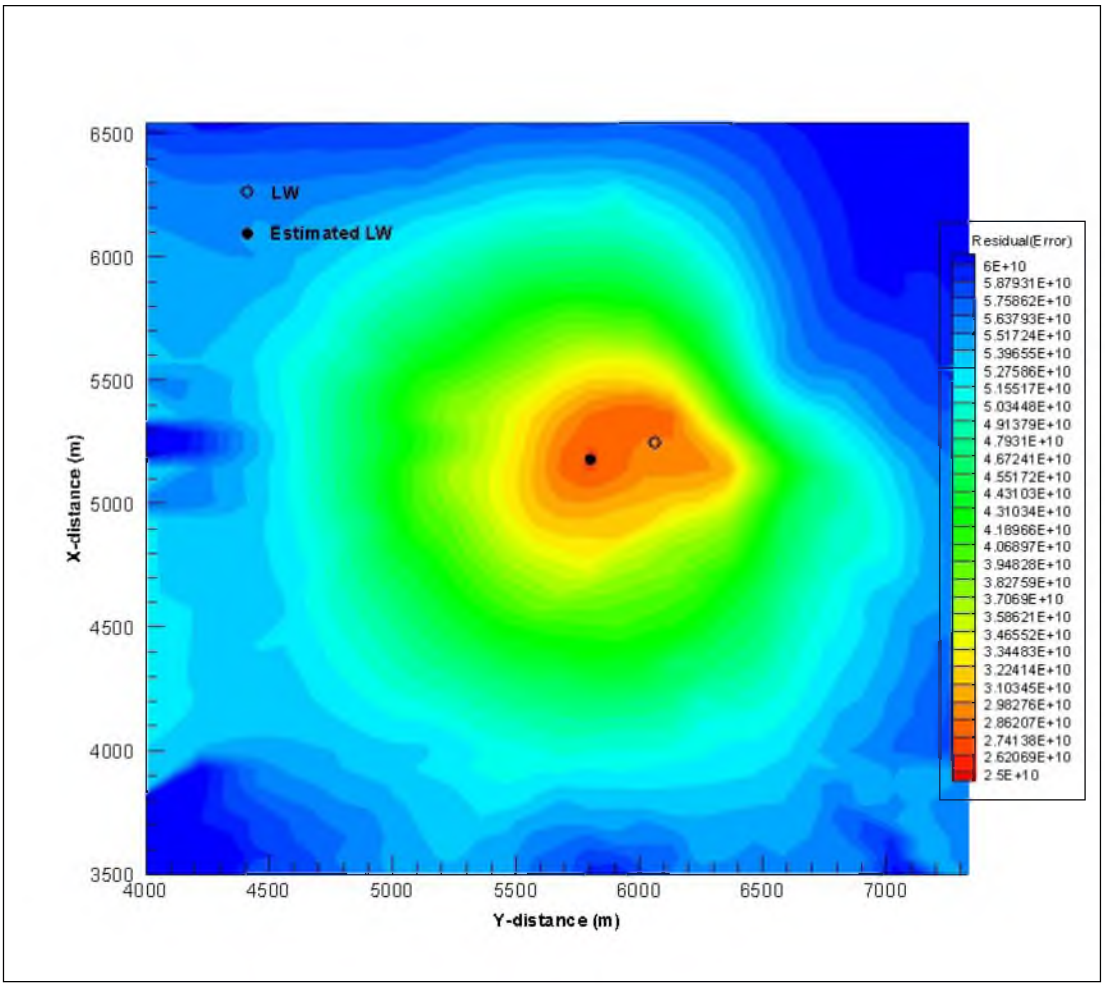


Fig. 4.5 Estimated leakage well location based on the objective function for the simulation with underestimated permeability $10^{-15.5}$ m^2.

This simulation resulted in a global minimum at (x, y) = (5150 m, 5750 m) as the predicted leakage well location. The deviation from the actual leakage well location at (x, y) = (5250 m, 6050 m) is about 316 m. Specific objective function values of case 2 (a) and (b) are shown in Table 4.2. In case 2, the two estimations from the overestimated and underestimated permeability model permutations do not significantly deviate from the actual leakage well location. However, we can still establish that the uncertainty of the overlying formation permeability influences the accuracy of the inversion for leakage pathway detection.

Table 4.2 Objective function values of case 2 (a) and (b).

Initial guess		**Objective function (Pa^2)**	
Number	**Coordinate (m)**	**Case 2 (a)**	**Case 2 (b)**
True	**(5250, 6050)**	-	-
5	(4950, 5550)	0.3435e+11	0.3185e+11
9	(5150, 5550)	0.3420e+11	0.3154e+11
10	(5150, 5750)	0.3073e+11	**0.2695e+11 (Deviation: 316 m)**
11	(5150, 5950)	**0.2768e+11 (Deviation: 141 m)**	0.2827e+11
12	(5150, 6050)	0.2791e+11	0.2978e+11
13	(5150, 6150)	0.2952e+11	0.2894e+11
22	(5250, 5950)	0.3019e+11	0.2799e+11
23 (True)	**(5250, 6050)**	**0.3077e+11**	**0.2954e+11**
24	(5250, 6150)	0.3230e+11	0.2863e+11
25	(5250, 6350)	0.3899e+11	0.3350e+11
26	(5250, 6550)	0.4606e+11	0.4395e+11
34	(5350, 6050)	0.3280e+11	0.2739e+11
35	(5350, 6150)	0.3465e+11	0.2826e+11

The third simulation ("case 3") includes the overlying formation permeability as an additional unknown parameter. This inversion estimates an optimum combination of both the vertical permeability of the initial guesses of the leakage pathway and the overlying formation permeability. This case identifies that estimating the uncertain permeability of the overlying formation can improve the accuracy of the leakage pathway estimation. The methodology of inverse modeling is the same as that described at the first and second cases.

The contour plot of the objective function from the third inversion is shown in Fig. 4.6. Table 4.3 presents the objective function values of case 3. In this inversion, the leakage pathway was similar to case 1, the idealized case. Results indicate that estimating the combination of both the vertical permeability of the initial guesses and the overlying formation permeability reduces the impact of the uncertainty of the overlying formation permeability and increases the accuracy of detection of the leakage pathway location. The increased accuracy of this case can be identified by the residual analysis.

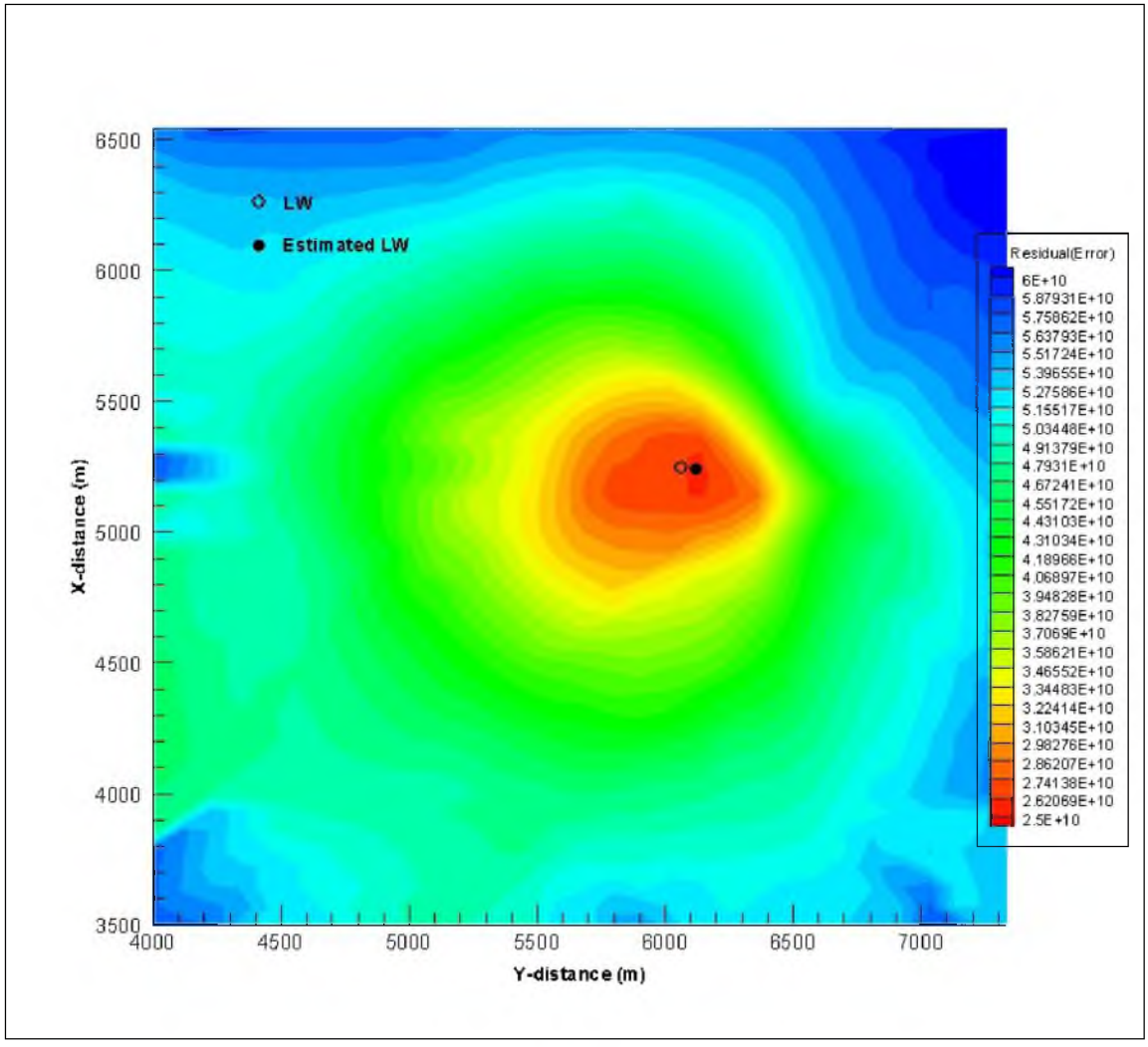


Fig. 4.6 Estimated leakage well location from objective function in case 3.

Table 4.3 Objective function values of case 3.

Initial guess		**Objective function (Pa^2)**
Number	**Coordinate (m)**	
True	**(5250, 6050)**	-
5	(4950, 5550)	0.3185e+11
12	(5150, 6050)	0.2637e+11
14	(5150, 6350)	0.2775e+11
15	(5150, 6550)	0.3768e+11
16	(5150, 7050)	0.5036e+11
23 (True)	**(5250, 6050)**	**0.2673e+11**
24 (Best)	**(5250, 6150)** **(Deviation: 100 m from true)**	**0.2559e+11**
25	(5250, 6350)	0.3082e+11
26	(5250, 6550)	0.4071e+11
31	(5350, 5550)	0.3252e+11
34	(5350, 6050)	0.2628e+11
35	(5350, 6150)	0.2657e+11

The inverse analysis estimates an optimum parameter set based on given conditions of geologic and hydrologic properties in the system. Thus, the errors of the properties result in deviated estimations. The main parameter which can influence the inversion results has to be estimated to reduce the impact from uncertainty of the parameter (Finsterle, 2004). Table 4.4 denotes the statistics of estimated parameters in each simulation case. The arithmetic means and standard deviations described in Table 4.4 are from model results for estimated permeability values of 48 initial guesses with minimum objective function values. The residual analysis associated with these simulation results is described in the next section.

Table 4.4 Statistics of estimated parameters in each simulation case.

		Estimated LW locations (m)			Logarithm of estimated permeability of initial guesses (m^2)		Logarithm of estimated permeability of overlying formation (m^2)	
		X	Y	Deviation	Average	Std. dev.	Average	Std. dev.
True		5250	6050	-	-10.0	-	-15.0	-
1st Case		5250	6150	100	-10.82	1.12	-	-
2nd Case	(a)	5150	5950	141	-10.25	0.63	-	-
	(b)	5150	5750	316	-10.94	1.29	-	-
3rd Case		5250	6150	100	-10.58	1.03	-15.06	0.25

4.1.1 Residual Analysis

Fig. 4.7 presents residuals (vector $\mathbf{r} = \mathbf{z}^* - \mathbf{z(p)}$) of the best estimates of the disparity between measured pressures (vector $\mathbf{z}^*$) and calculated pressures (vector $\mathbf{z(p)}$) in the storage formation for all three cases. Fig. 4.8 illustrates the residuals for the overlying formation from each simulation. In Fig. 4.8 the residuals for case 2 ((b) and (c)) are larger than those for case 1 (a) and case 3 (d), indicating that the inversion of case 2 has lower accuracy and the uncertainty of the overlying formation permeability affected the results. On the other hand, in Fig. 4.7, the residuals for the storage formation in each case are almost identical although the residuals are calculated from the different estimated results of each case. This result implies that the pressure anomalies in the storage formation may not be suitable for estimating leakage pathway locations. The reason is that the large amount of injected CO_2 can damp the pressure anomalies in the storage formation. However, pressure anomalies induced by leaks into the overlying formation are sufficient to estimate possible leakage pathway locations by inversion. To demonstrate these results, two inverse models were conducted based on the "idealized case."

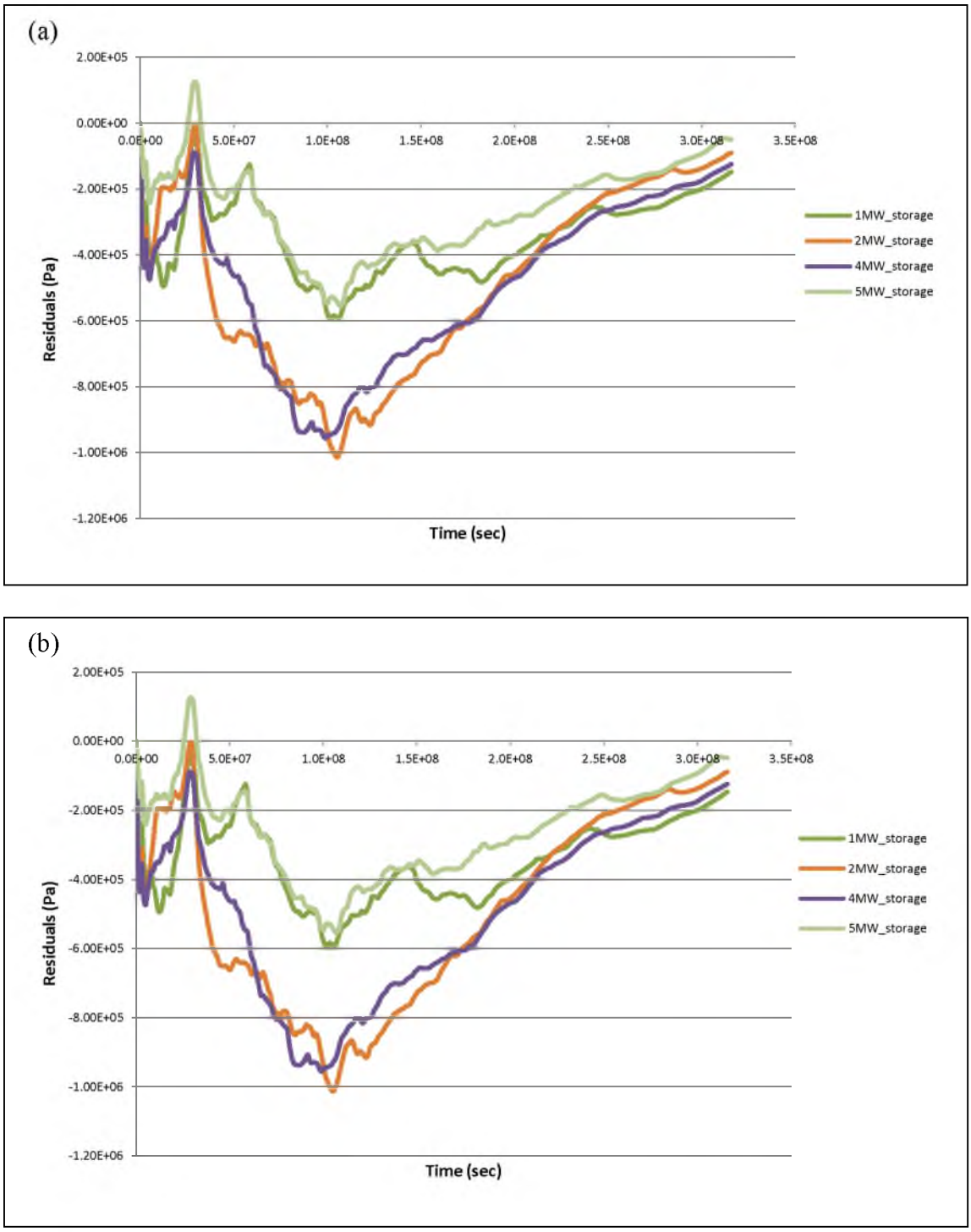


Fig. 4.7 Residuals between measured and calculated pressures in the storage formation in each simulation case for (a) case 1, (b) case 2 (permeability $10^{-14.5}$ m^2), (c) case 2 (permeability $10^{-15.5}$ m^2) and (d) case 3.

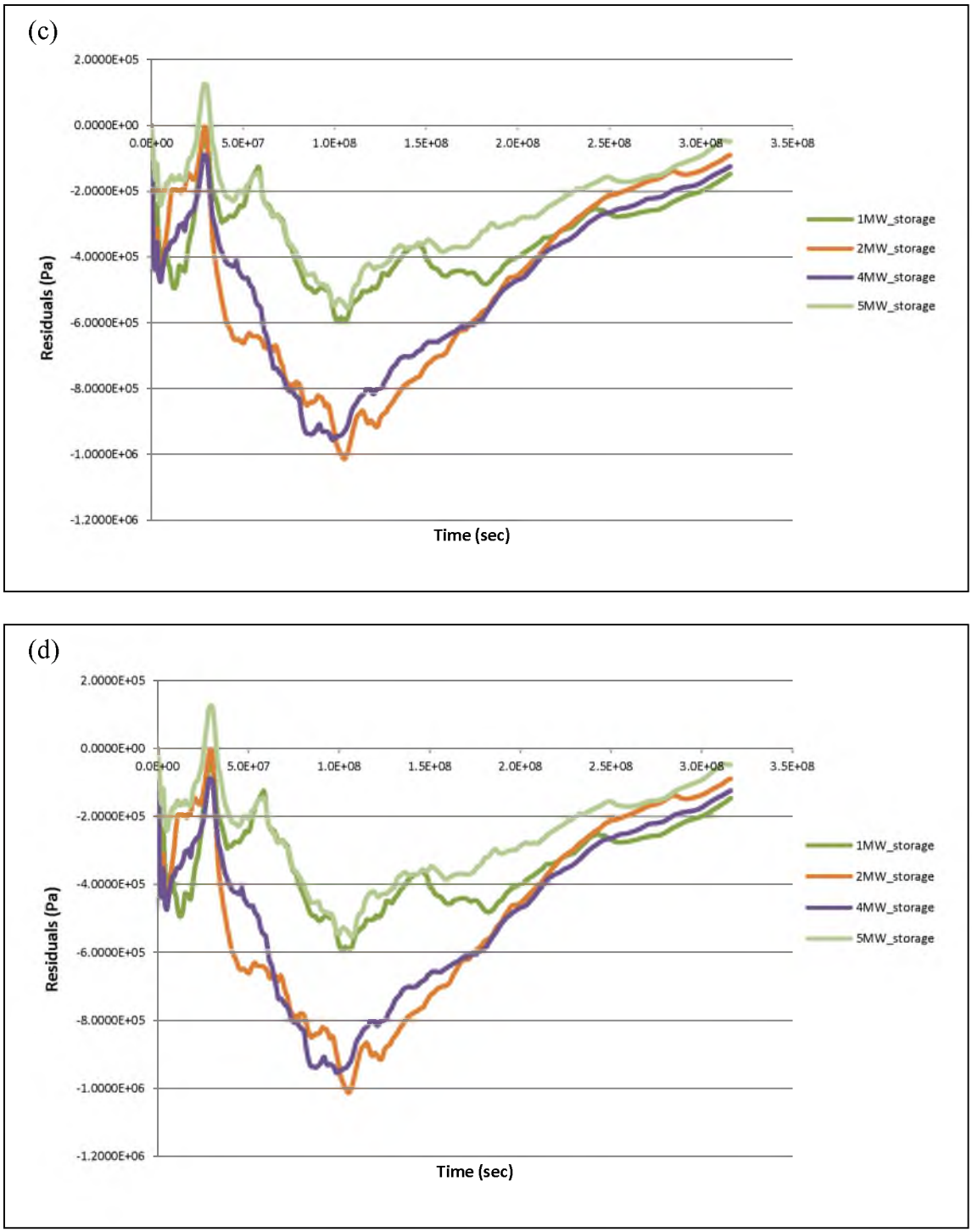


Fig. 4.7 Continued.

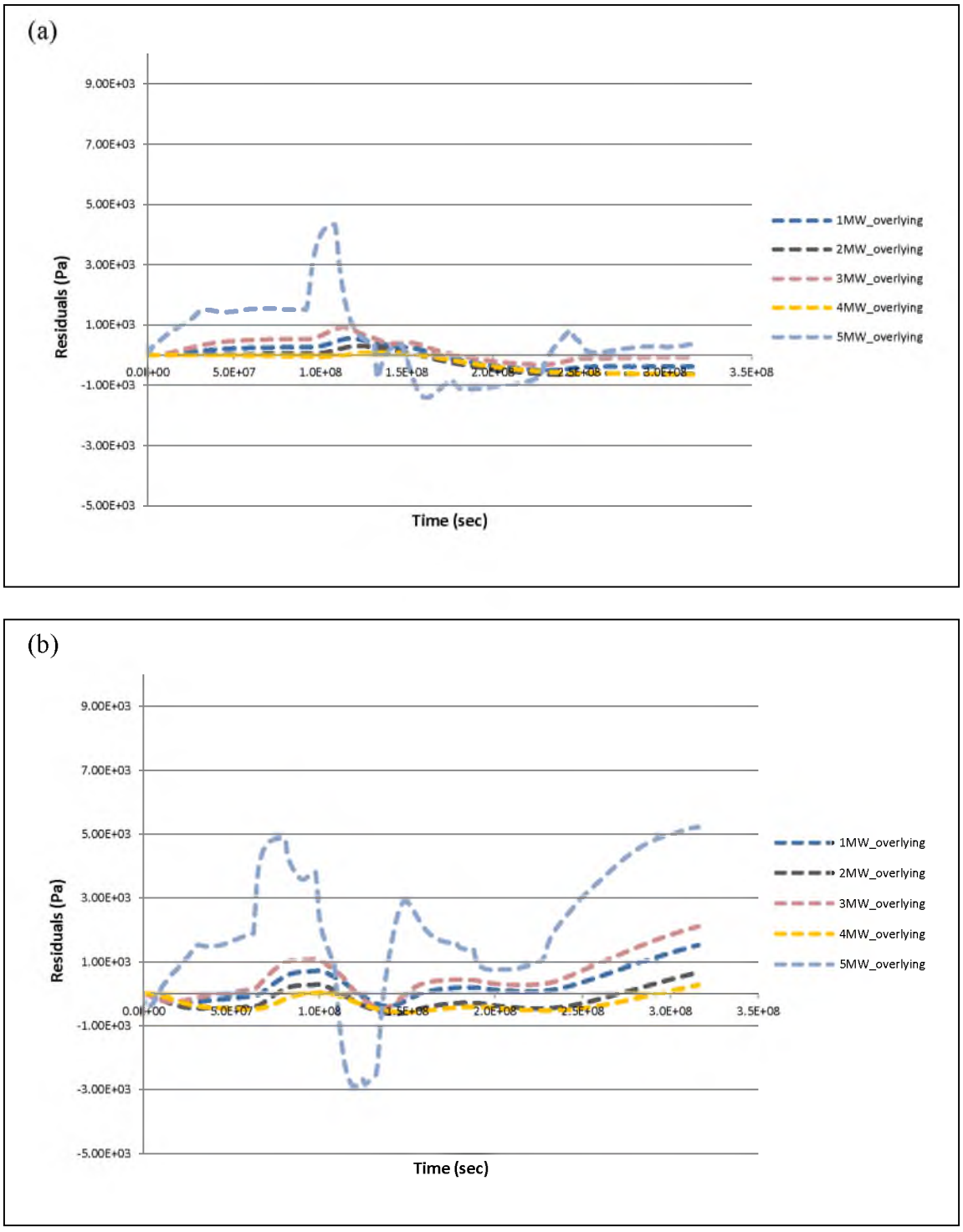


Fig. 4.8 Residuals in the overlying formation in each simulation case for (a) case 1, (b) case 2 (permeability $10^{-14.5}$ m^2), (c) case 2 (permeability $10^{-15.5}$ m^2) and (d) case 3.

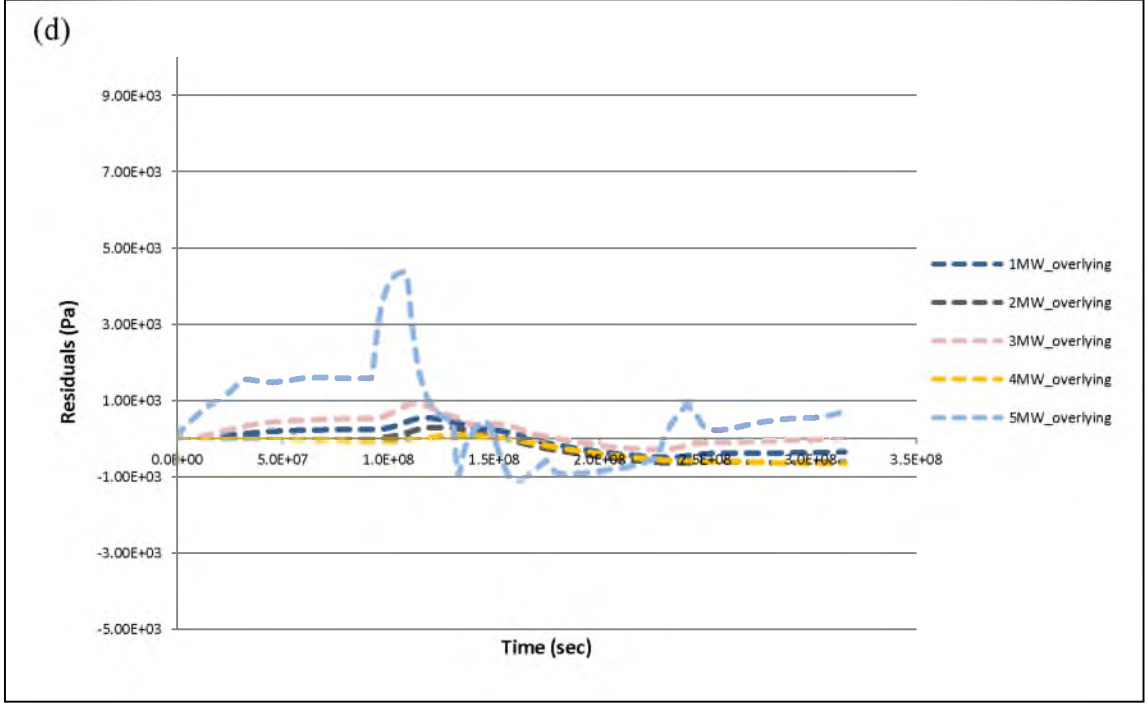
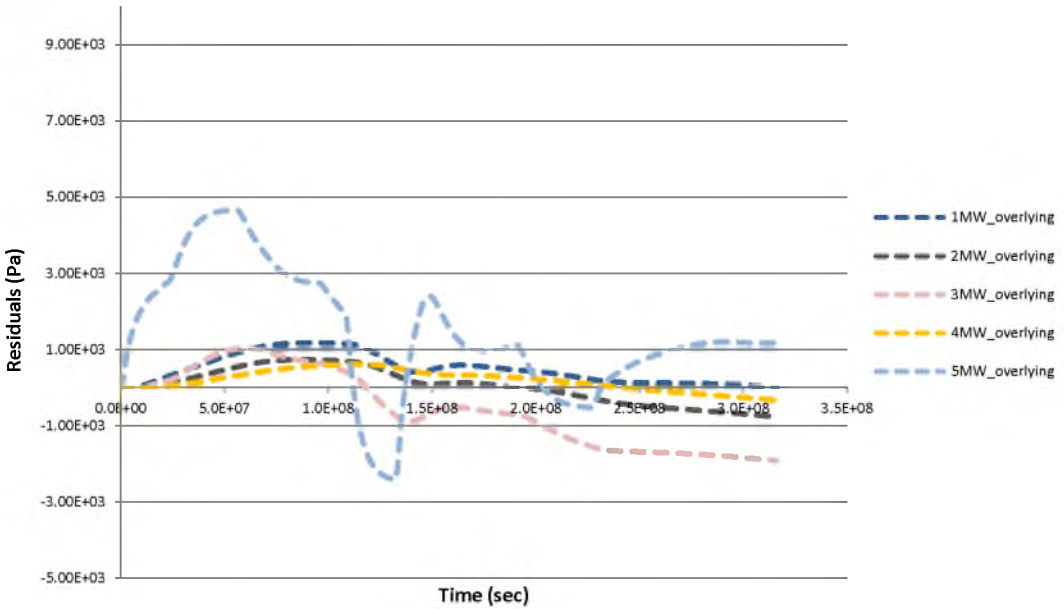


Fig. 4.8 Continued.

(c)


9.00E+03
7.00E+03
5.00E+03
3.00E+03
1.00E+03
-1.00E+03
-3.00E+03
-5.00E+03
Residuals (Pa)
0.0E+00
5.0E+07
1.0E+08
1.5E+08
2.0E+08
2.5E+08
3.0E+08
3.5E+08
Time (sec)
1MW_overlying
2MW_overlying
3MW_overlying
4MW_overlying
5MW_overlying

First, inverse analysis estimated the leakage pathway location based on only measurement data in the overlying formation (Fig. 4.9 (a)). The second inversion estimated the leakage location based on only measurement data in the storage formation (Fig. 4.9 (b)). In Fig. 4.9 (a) the leakage pathway was estimated similarly to case 1 but the second inversion was ill-posed. These inverse modeling results identify that only pressure anomalies induced by leaks in the overlying formation are critical to estimate the leakage pathway location.

Another interesting finding is that the deviations at MW5 are relatively very large as shown in Fig. 4.8. The pressures at MW5 include significant pressure anomalies associated with capillary effects at the leakage well. The degree of deviation at MW5 can be an important factor to estimate the leakage location even if the idealized case has several errors in MW5. In the model domain, intervals of initial guesses of the leakage pathway are at most 100 m. To alleviate the errors at MW5, the model with finer intervals of initial guesses is required. This is because the drifts of times when CO_2 reaches the bottom of leakage pathway can result in errors. As mentioned earlier, weighting factors are needed to scale the magnitude of measurements and residuals. The storage formation in which CO_2 is injected has higher pressure than that of the overlying formation, so the objective function values, which are calculated by sum of residuals, can be much larger. Such contrast can lead to failure of the inverse model solution. In Fig. 4.7 and Fig. 4.8, the magnitude of residuals in the storage formation is approximately 100 times larger than that in the overlying formation. Therefore, 1 Pa and 100 Pa weighting factors were used to scale residuals of the overlying and storage formations, respectively. The weighting factors should be assigned based on such residual analysis to improve results.

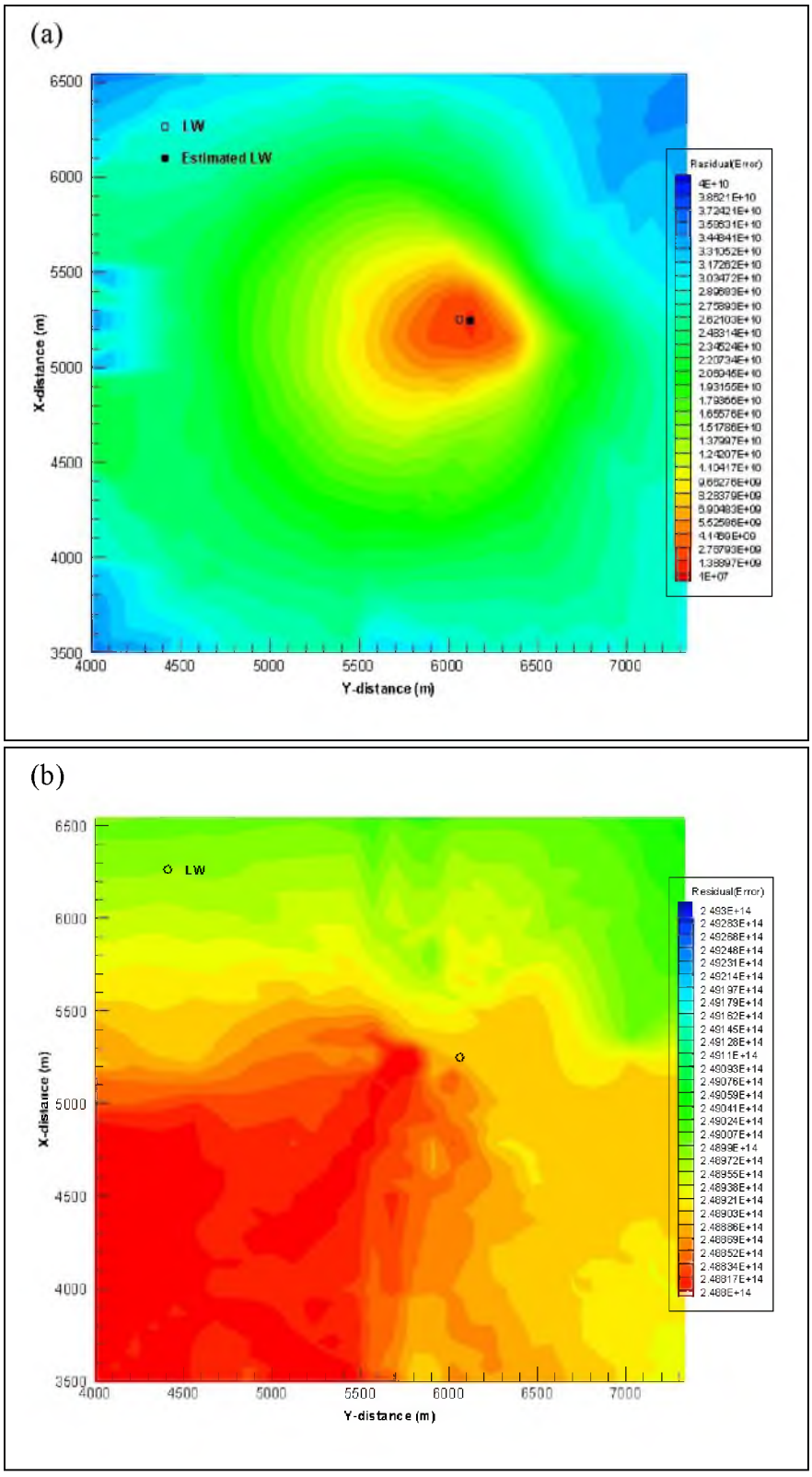


Fig. 4.9 Estimated leakage well location from objective function: (a) using measurements in the overlying formation and (b) using measurements in the storage formation.

4.1.2 Additional Inverse Modeling

As described in the previous section, it was identified that the accuracy of the leakage pathway estimation can be increased by calibrating the uncertain permeability. In this section, two different errors are introduced into the inverse modeling. Inverse modeling examines their effects on leakage pathway estimation. First, the inversion identifies the impact of uncertainty in the leakage pathway size on leakage pathway estimation. In the second calibration, random noises are included in the measurement data, and the effect of noises on leakage pathway estimation is examined.

4.1.2.1 Effect of Uncertain Leakage Pathway Size

Inverse modeling of the previous three scenarios estimated the leakage pathway by 48 initial guesses on the basis of an actual leakage pathway size (0.3 m × 0.3 m). Various leakage pathway sizes, like abandoned wells or faults, can exist in the field. The uncertainty in leakage pathway sizes can influence the accuracy of parameter estimations. Thus, the inversion examines the impact of uncertainty in the leakage pathway size. The leakage pathway sizes of 48 initial guesses are meshed using a unit area (1 m × 1 m) in the model domain. Inverse modeling calibrates the vertical permeability of each initial guess based on overestimated its sizes. The other inversion conditions are the same as the "idealized case." Fig. 4.10 presents pressure differences in the overlying formation induced by the overestimated size (1 m × 1 m) and the actual size (0.3 m × 0.3 m) of the leakage pathway after the simulation period (10 years). In Fig. 4.10, the overestimated leakage pathway size significantly magnifies the pressure anomalies in the overlying formation.

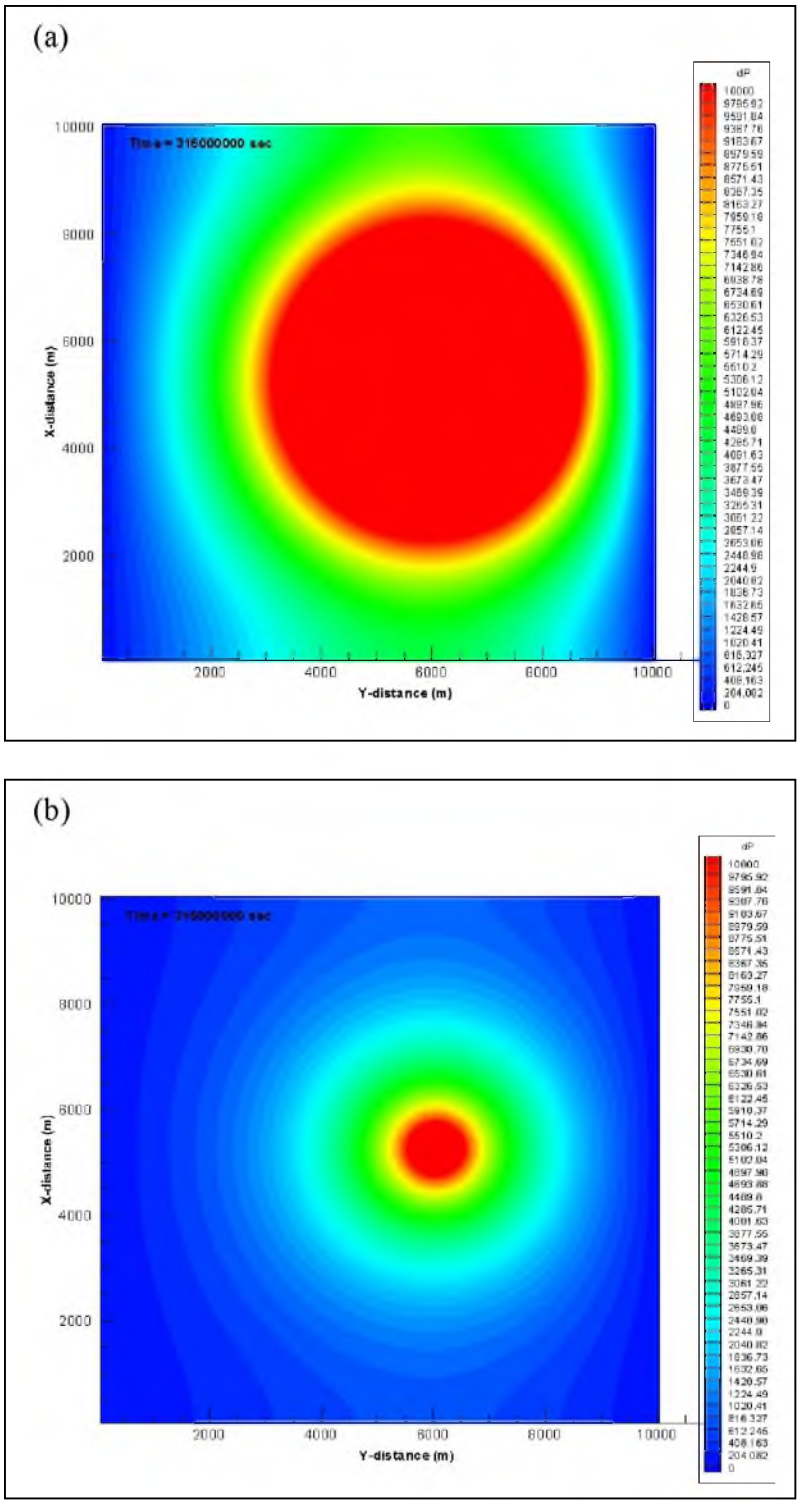


Fig. 4.10 *dP* in the overlying formation by two different leakage pathway sizes: (a) overestimated leakage pathway size: 1 m × 1 m (*dP* scale: 0 – 10,000 Pa) and (b) actual leakage pathway size: 0.3 m × 0.3 m (*dP* scale: 0 – 10,000 Pa).

Results leading to an increase in the pressure anomalies can influence the accuracy of the leakage pathway estimation. Fig. 4.11 illustrates a contour plot of the objective function from the inversion based on the initial guesses with the overestimated leakage pathway size. In the inversion, 1 Pa and 100 Pa were used for the weighting factors of the overlying and storage formations, respectively. In Fig. 4.11 the most possible leakage well was estimated as (5150 m, 6550 m). A deviation from the actual leakage well is approximately 510 m. Table 4.5 denotes the specific objective function values of initial guesses that were estimated from the inversion.

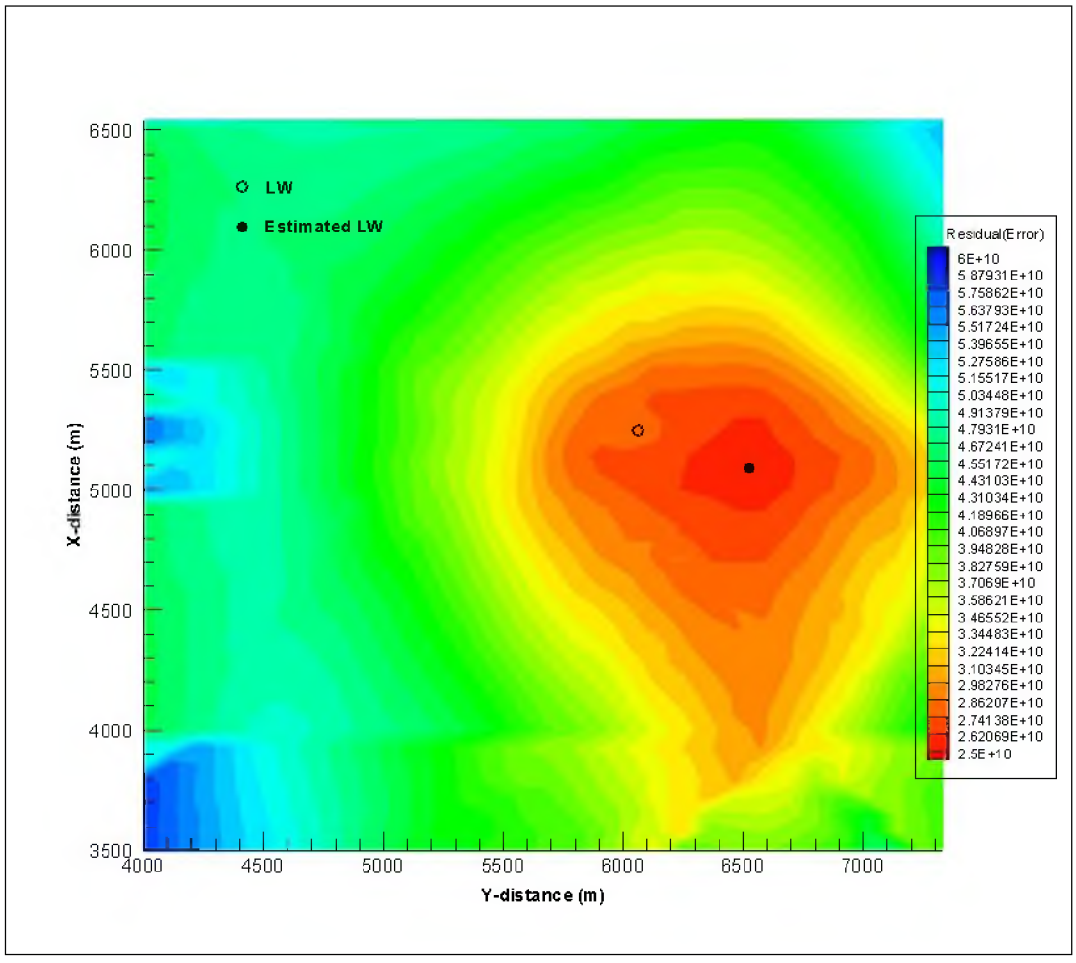


Fig. 4.11 Estimated leakage well location based on the objective function for the simulation with overestimated leakage pathway size 1 m × 1 m.

Table 4.5 Objective function values of the leakage pathway estimation.

Initial guess		**Objective function (Pa^2)**
Number	**Coordinate (m)**	
True	**(5250, 6050)**	-
6	(4950, 6550)	0.2591e+11
12	(5150, 6050)	0.2722e+11
14	(5150, 6350)	0.2582e+11
15 (best)	**(5150, 6550) (Deviation: 510 m from true)**	**0.2556e+11**
16	(5150, 7050)	0.2843e+11
23 (true)	**(5250, 6050)**	**0.2856e+11**
25	(5250, 6350)	0.2657e+11
26	(5250, 6550)	0.2583e+11
31	(5350, 5550)	0.3305e+11
37	(5350, 6550)	0.2632e+11

The incorrect leakage pathway size assigned to the initial guesses results in reducing accuracy in the leakage pathway estimation due to errors in the calculation of pressure anomalies. This indicates that the uncertainty in the leakage pathway sizes should be calibrated to improve the accuracy of the leakage pathway estimation. The conventional method for leakage simulation characterizes the geometry of the leakage pathway as meshes (Nordbotten et al., 2004). In the conventional method, the inversion should be iterated depending on each initial guess meshed to characterize various leakage pathway sizes. Those kinds of inversions can be inefficient because the number of inverse modeling will be increased. Therefore, the sizes have to be parameterized for more effective parameter estimation. The parameterization of leakage pathway sizes will be specifically described in Chapter 5.

4.1.2.2 Effect of Singular Noises in Measurements

The effect of noises in the measurements on parameter estimation was examined by one simulation case with noises. The noises are randomly added in measured pressure profiles at all of the monitoring wells by ± 0.1 % of the magnitude of each pressure data point. The noises at each measurement data point have a nonzero mean. The statistics of noises at nine measurement points are shown in Table 4.6.

The inversion is applied to case 3 in section 4.1. The inversion simultaneously estimates both the vertical permeability of the 48 initial guesses of leakage pathway and the permeability of the overlying formation based on measurements with random noises by 0.1 %. The inverse modeling calibrates the optimum combination of both parameters to minimize the objective function. The weighting factor of 10 Pa is used for measurements in the overlying formation to reduce residuals that are increased by noises. A weighting factor of 10,000 Pa is assigned to measurements in the storage formation. The weighting factor of 10,000 Pa can deactivate the measurements in the storage formation. Fig. 4.12 illustrates the fluctuations of measurements with random noises in the overlying formation. Fig. 4.13 represents the random fluctuations of measurements in the storage formation.

Table 4.6 Statistics of noises at all measurement points.

	Overlying formation					**Storage formation**			
	MW1	**MW2**	**MW3**	**MW4**	**MW5**	**MW1**	**MW2**	**MW4**	**MW5**
Mean	-0.19E3	0.21E3	-0.90E2	0.34E2	0.10E3	0.15E2	-0.47E2	0.86E2	-0.46E3
Std. dev.	0.60E4	0.61E4	0.61E4	0.59E4	0.59E4	0.85E4	0.86E4	0.84E4	0.85E4

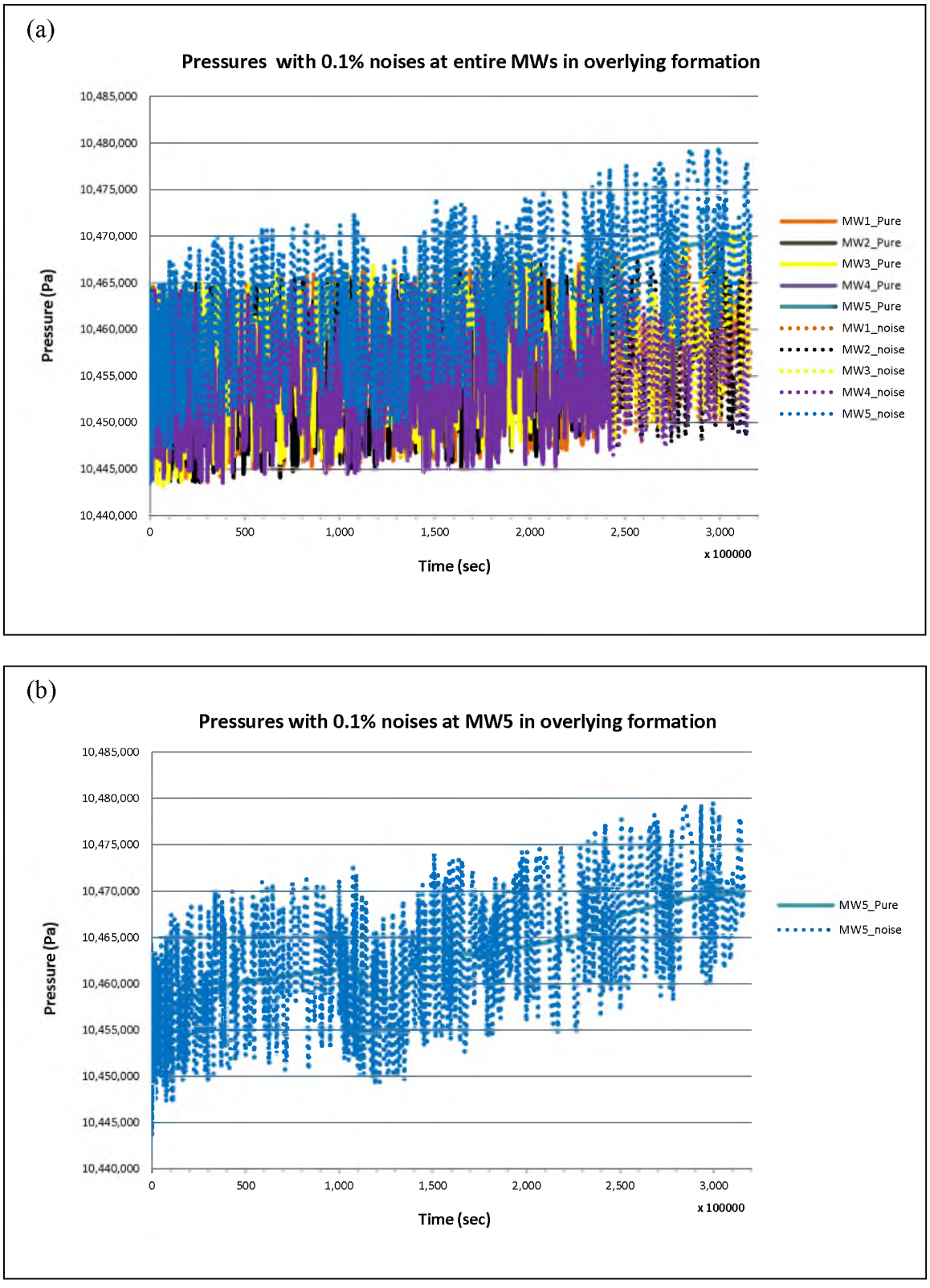
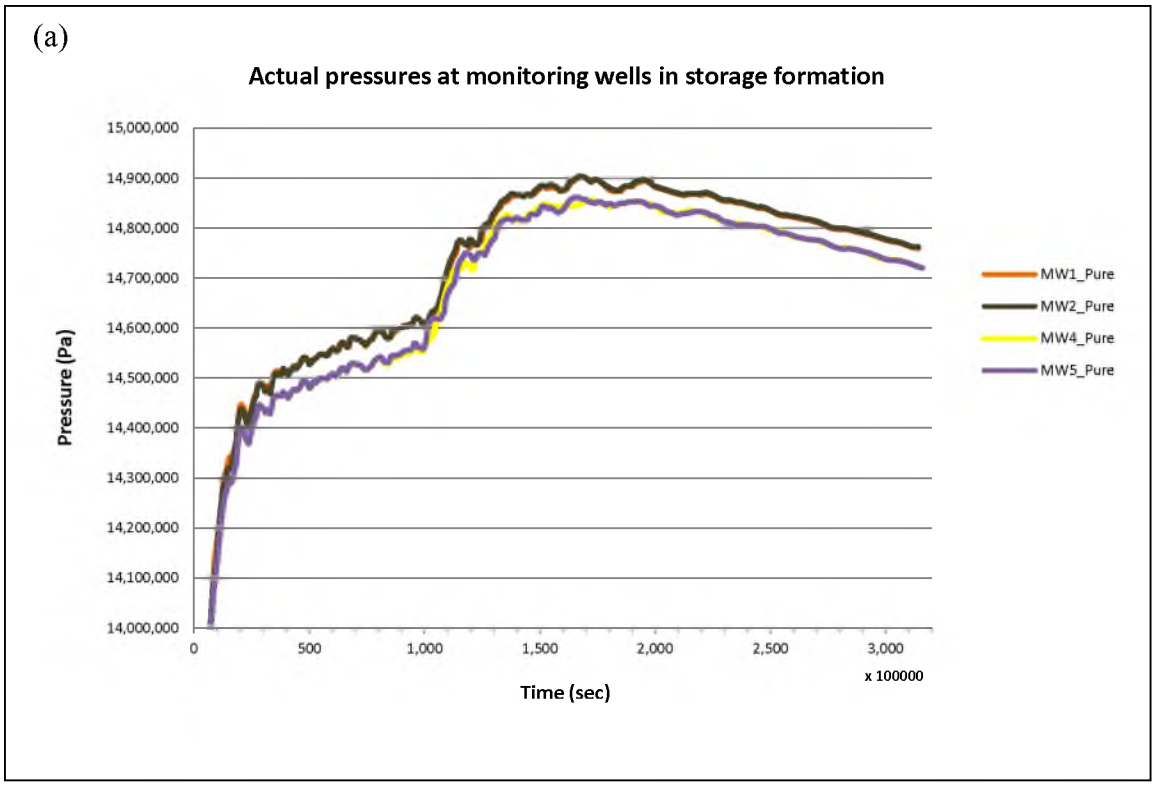


Fig. 4.12 Measurements with random noises by 0.1 % in the overlying formation at (a) whole MWs and (b) MW5.



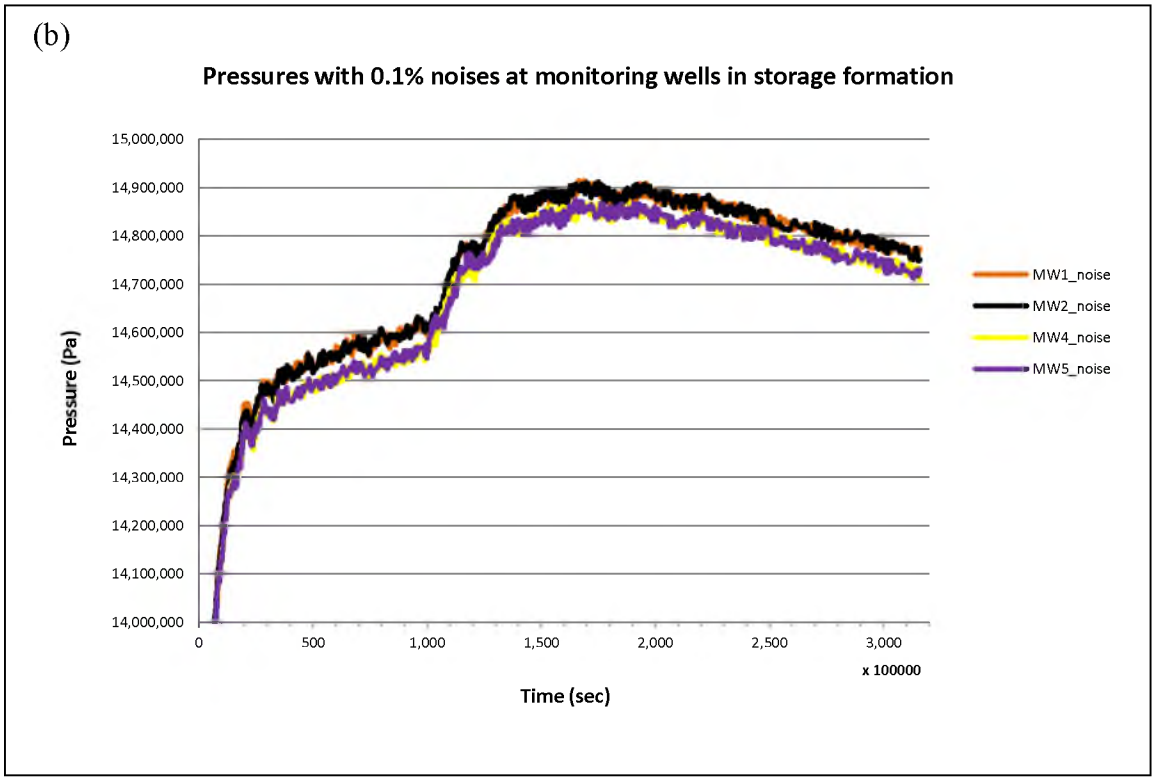


Fig. 4.13 Measurements at MWs in the storage formation: (a) actual measurements and (b) measurements with random noises by 0.1 %.

A contour plot of the objective function from measurements with random noises by 0.1 % is shown in Fig. 4.14. The parameter estimation determined that the most probable leakage well is located at (5250 m, 6150 m). The deviation from the actual leakage well is 100 m. In this inversion, the leakage pathway was reasonably estimated. Table 4.7 shows the specific objective function values of some of the initial guesses that were estimated. Table 4.8 describes the statistics of two estimated parameters with minimum objective function values.

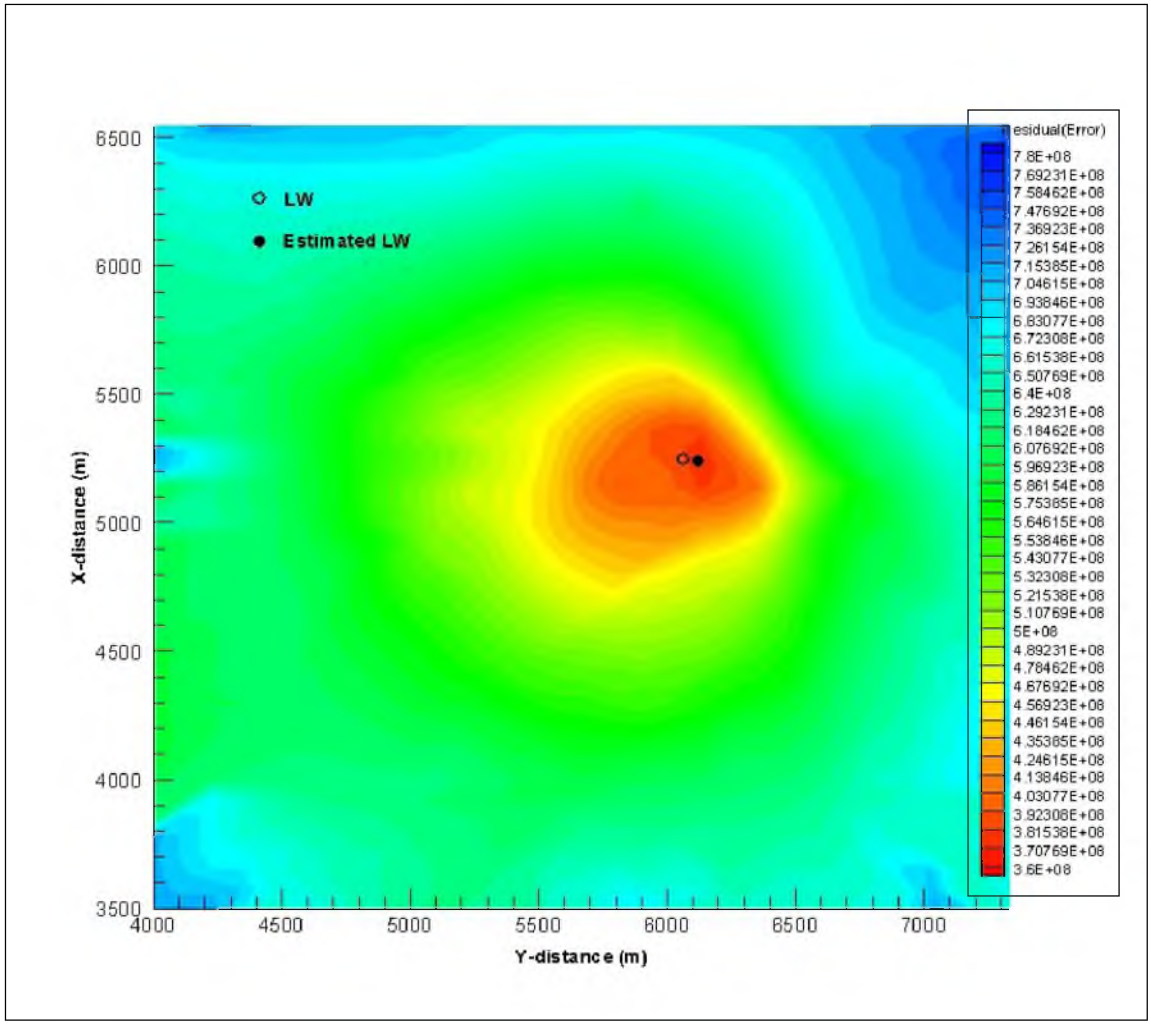


Fig. 4.14 Estimated leakage well location from the objective function for the measurements with random noises by 0.1%.

Table 4.7 Objective function values of estimated initial guesses.

Initial guess		**Objective function (Pa^2)**
Number	**Coordinate (m)**	
True	**(5250, 6050)**	-
6	(4950, 6550)	0.5487E+09
12	(5150, 6050)	0.3921E+09
14	(5150, 6350)	0.3969E+09
15	(5150, 6550)	0.5077E+09
16	(5150, 7050)	0.6358E+09
23 (true)	**(5250, 6050)**	**0.3903E+09**
24 (best)	**(5250, 6150)** **(Deviation: 100 m from true)**	**0.3707E+09**
25	(5250, 6350)	0.4312E+09
26	(5250, 6550)	0.5383E+09
31	(5350, 5550)	0.4531E+09
34	(5350, 6050)	0.3776E+09
35	(5350, 6150)	0.3824E+09

Table 4.8 Statistics of two estimated parameters.

	Estimated LW locations (m)			**Logarithm of estimated permeability of initial guesses (m^2)**		**Logarithm of estimated permeability of overlying formation (m^2)**	
	X	**Y**	**Deviation**	**Average**	**Std. dev.**	**Average**	**Std. dev.**
True	5250	6050	-	-10.0	-	-15.0	-
Estimation	5250	6150	100	-10.81	1.14	-15.1	0.297

4.2 Leakage Detection in Heterogeneous Domain

This section focuses inverse analysis to estimate a leakage pathway in a heterogeneous field, and the effect of weighting coefficients on inversion results. The inverse analysis is applied to the heterogeneous domain of section 3.4, the model characterized from the SACROC unit in Texas. This inverse modeling estimates the location of the actual leakage pathway from the initial guesses and renormalized permeability groups to reduce the impact of its errors.

Before discussing the inverse analysis of a heterogeneous domain, it needs to be noted that estimating significant heterogeneity by inverse modeling is a fundamentally difficult problem (Finsterle, 2004). It is not simple to estimate a large number of permeability values (including vertical permeability of initial guesses of a leakage pathway) in a heterogeneous system on the basis of limited pressure data measured from monitoring wells. As mentioned in section 4.1, the inverse modeling requires a priori information about the geologic and hydrologic properties, particularly fundamental permeability values and heterogeneity. If that information is not known or significantly erroneous, the number of parameters to be estimated can be enormous. The inversion may estimate many sets of "optimum" parameter values (nonuniqueness), or the estimated parameters may significantly deviate from true parameter values (Finsterle, 2004). The difficulties of estimation of heterogeneity are related to not only the number and uncertainties of parameters but also the sensitivity of the permeability values at different locations to measured pressure data as described in section 2.3.1. Therefore, because of these complicating factors, this study is limited to known heterogeneity in the model domain of Fig. 3.12. In this inverse analysis with heterogeneity, approximated average

permeability values are assigned to each discrete grid block of the model, and the grid blocks with similar permeability values can be also grouped to simplify parameterization of the model. This is called "zonation" of heterogeneity (Finsterle, 2004). In Fig. 3.12, 27 permeability "groups," each with a different average value, are assigned to the 116,699 grid blocks in the model domain. In practice, when such permeability "groups" are estimated by inversion, the accuracy of parameter estimation may decrease because of the high number of unknowns to be estimated. For the parameter estimation in a heterogeneous system, the scenario is idealized to realize a general modeling approach. In the idealized inversion, the model domain of 27 permeability groups in Fig. 3.12 is assumed as the actual field and the pressure profiles at nine measurement points generated from the model domain are used as observed measurement data in the inversion. The number of permeability groups and grid blocks in the actual system needs to be reduced both to simplify and to reduce computational expense. Therefore, an upscaling technique, generally called renormalization, is introduced. An algorithm developed by King (1989) is used. Fig. 4.15 presents a schematic diagram of renormalization by King's algorithm. King's upscaling equation is

$$k_{rnp} = \frac{4(k_1+k_3)(k_2+k_4)[k_1k_2(k_3+k_4)+k_3k_4(k_1+k_2)]}{[k_1k_2(k_3+k_4)+k_3k_4(k_1+k_2)][k_1+k_2+k_3+k_4]+3(k_1+k_2)(k_3+k_4)(k_1+k_3)(k_2+k_4)}. \quad (4.1)$$

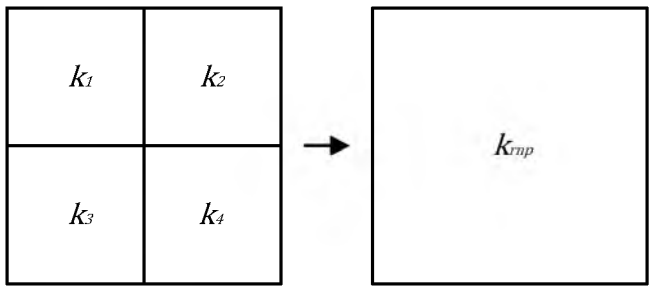


Fig. 4.15 Upscaled grid blocks by King's equation. Modified from Han et al. (2010).

The King algorithm was developed through an equivalent resistor network model. This equation generates approximately renormalized permeability in upscaled grid blocks. If the permeability values at four grid blocks have significant differences, this algorithm will underestimate the effective permeability at an upscaled grid block (Han et al., 2010). Thus this method will propagate errors to the results of forward modeling. Nevertheless, the renormalization is applied to the inversion for leakage pathway estimation.

The actual domain with 116,699 grid blocks specified by 27 permeability groups is sequentially renormalized, two times, so that the number of grid blocks is reduced to 7,436 and each grid block size is approximately 400 m × 400 m × 20 m in the overlying and storage formations. In addition, 27 permeability groups are simplified to eight permeability groups with similar permeability values, and the eight groups are assigned to the 7,436 grid blocks. Table 4.9 summarizes the statistics of permeability, number of grid blocks and the size of grid blocks following renormalization. The twice-permeability distributions in the overlying and storage formations renormalized for two times are shown in Fig. 4.16. The renormalized permeability distributions in the vicinity of the leakage well, the injection well and the four measurement wells are substantially underestimated, and become effectively smoothed.

Table 4.9 Statistics of permeability and grid blocks following renormalization.

	Number of Grid			**Total number of Grids**	**Grid size (m)**			**Permeability (m^2)**			
	X	**Y**	**Z**		**X**	**Y**	**Z**	**Arithmetic mean**		**Std. dev.**	
								X and Y	**Z**	**X and Y**	**Z**
1	103	103	11	116,699	100	100	20	1.06 E-14	5.85 E-15	3.84 E-14	7.17 E-13
2	52	52	11	29,744	200	200	20	8.75 E-15	5.83 E-16	3.38 E-14	2.26 E-15
3	26	26	11	7,436	400	400	20	5.58 E-15	3.72 E-16	2.26 E-14	1.50 E-15

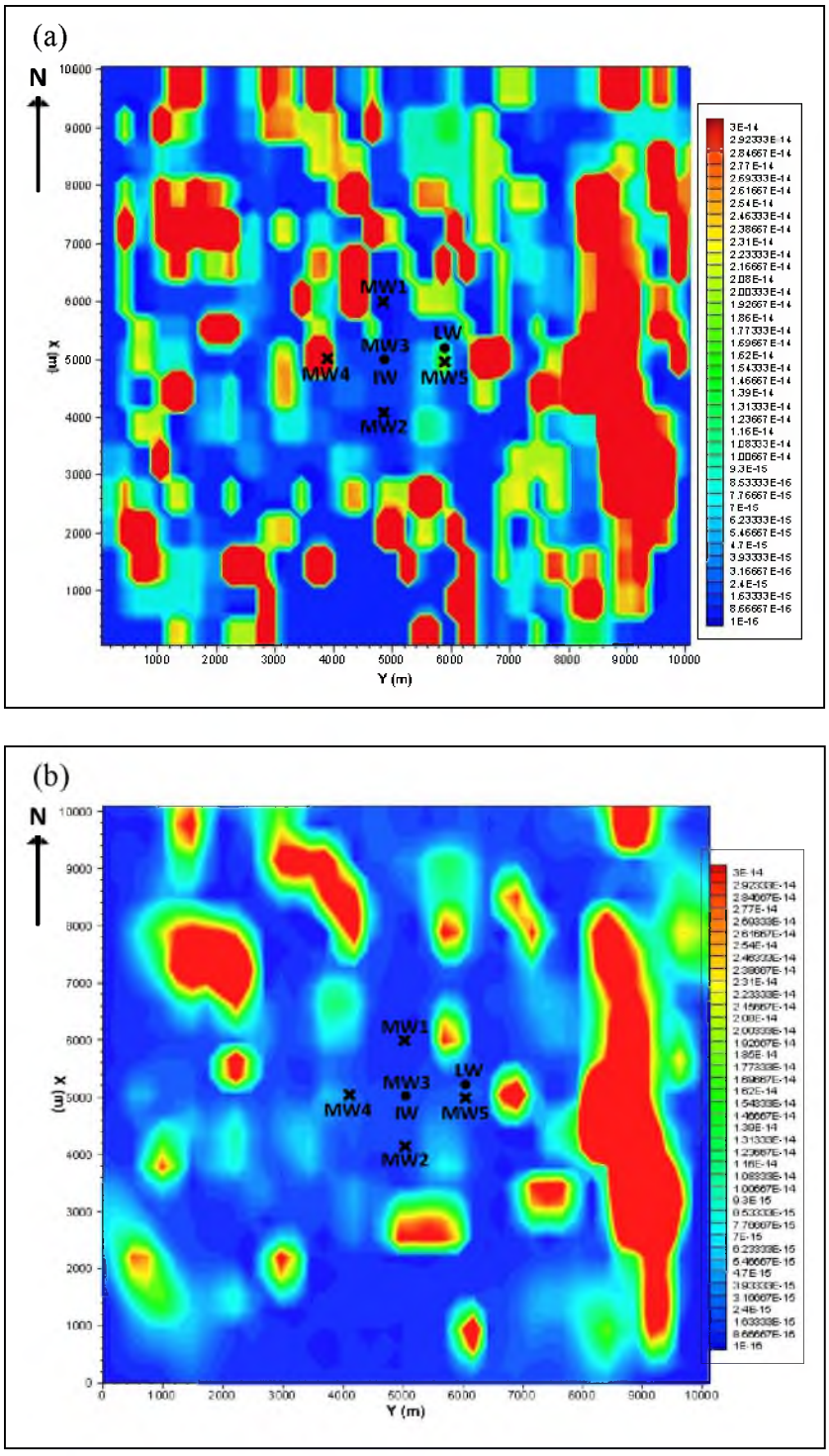


Fig. 4.16 Permeability distributions before and after renormalization in the overlying (a) and (b) and storage (c) and (d) formations, respectively: (a) before renormalization in the overlying formation, (b) after renormalization in the overlying formation, (c) before renormalization in the storage formation and (d) after renormalization in the storage formation.

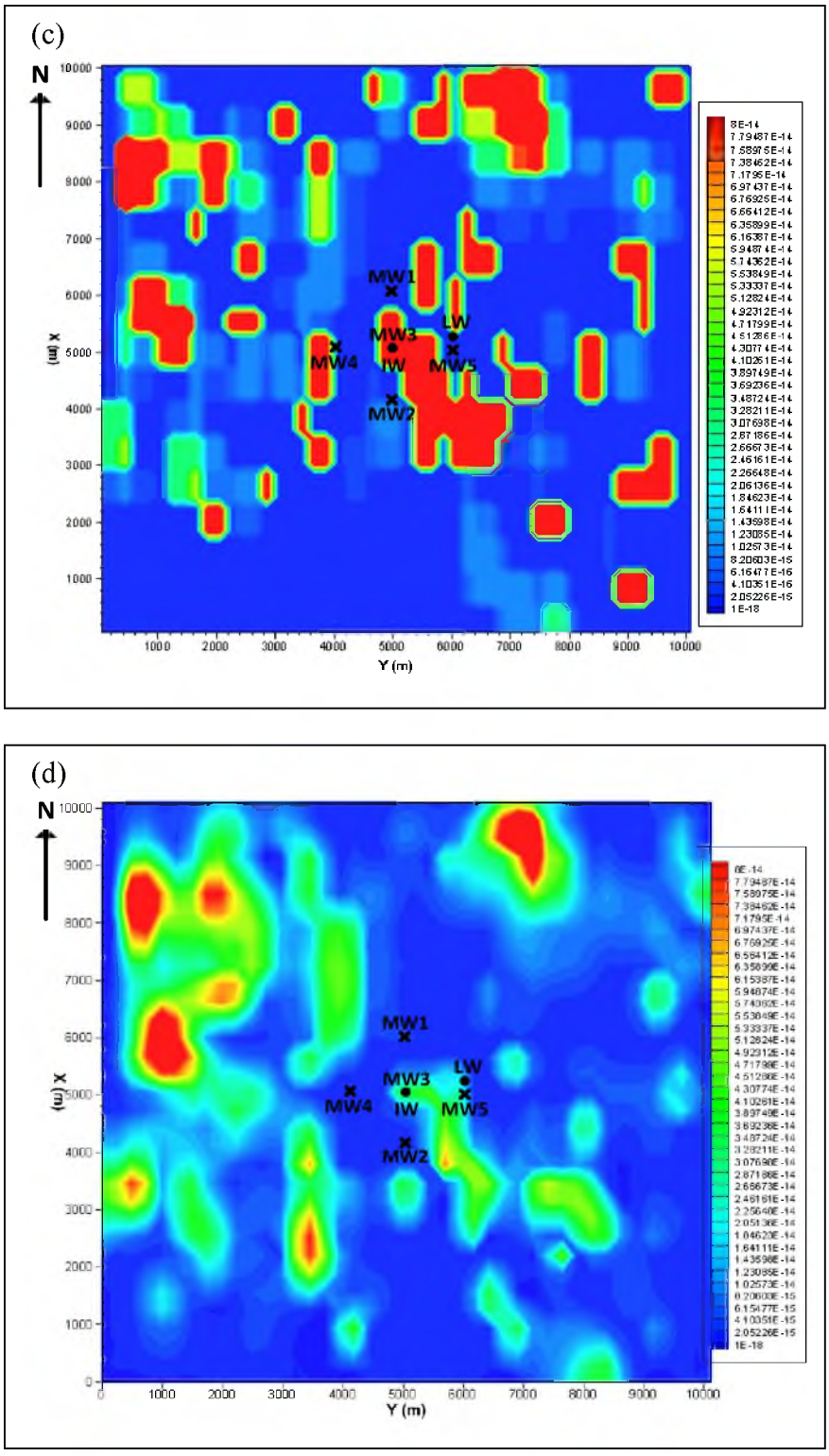


Fig. 4.16 Continued.

Fig. 4.17 presents resulting pressure perturbations after 10 years in the YZ-plane of the leakage pathway in the renormalized domain. When the pressures in the renormalized domain are compared to the pressures in the original domain of Fig. 3.16 (d), the pressures in the renormalized domain build more vertically around the injection well and lateral pressure propagation is reduced. Although the pressure gradient increases around the injection well, fluid migration in the storage formation decreases with underestimated permeability. Fig. 4.18 shows leakage rate deviation at the top of the leakage pathway penetrating caprock. Solid lines indicate the leakage rate from original heterogeneity and dashed lines denote the leakage rate from renormalized heterogeneity.

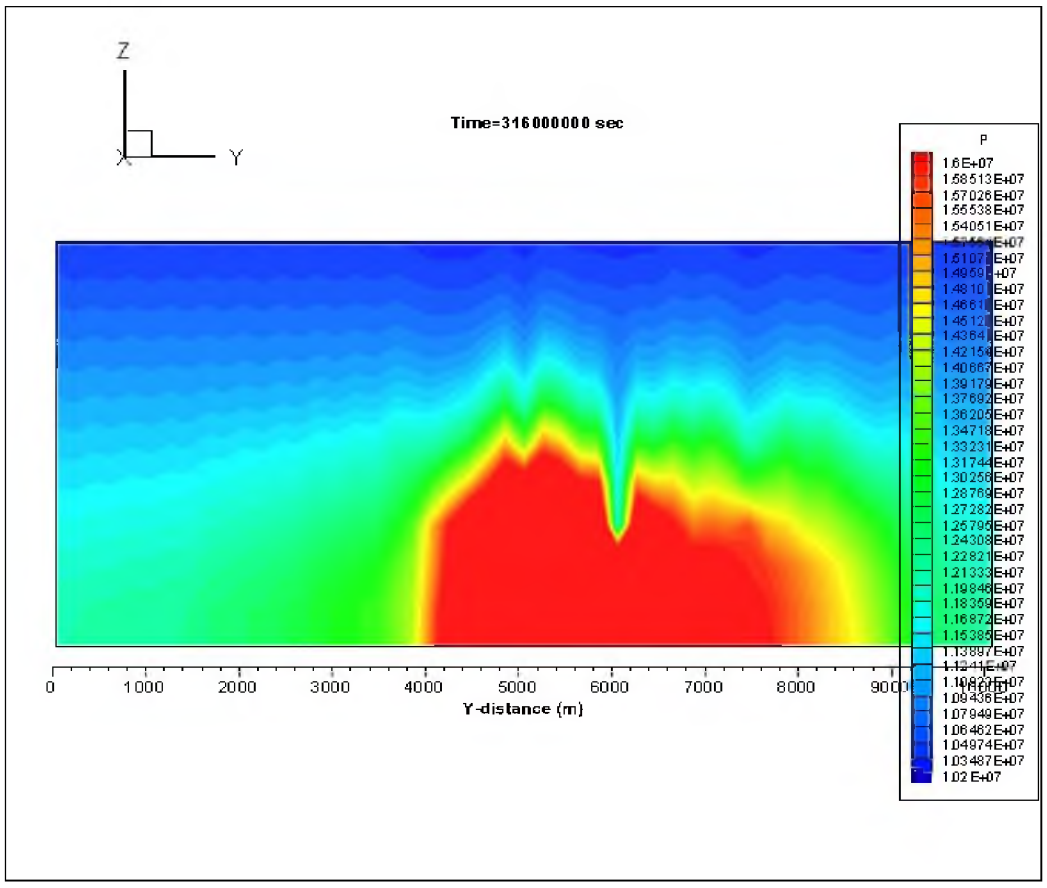


Fig. 4.17 Pressure propagation after 10 years in the YZ-plane of the leakage pathway in the renormalized domain.

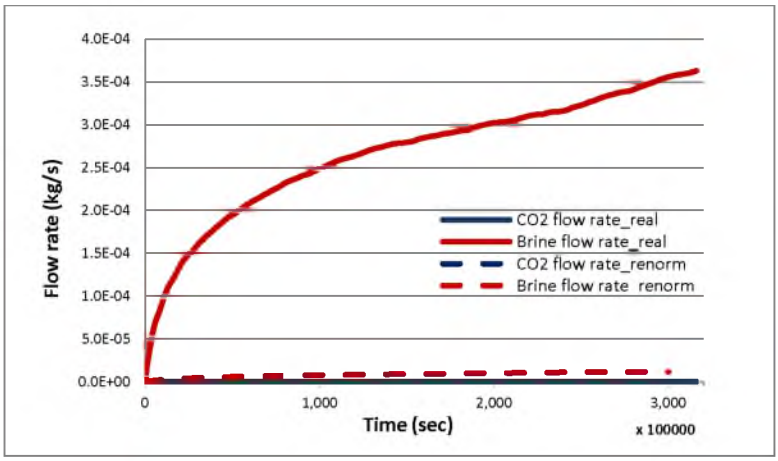


Fig. 4.18 Difference of leakage rates at the top of leakage pathway between the original and renormalized heterogeneous domains.

In Fig. 4.18 the flow rates through a leakage well are significantly diminished in the renormalized domain because fluid migration decreased for the underestimated permeability in the storage formation. Fig. 4.19 illustrates pressure drifts at five measurement points in the overlying formation of both the original (solid lines) and renormalized (dashed lines) domains.The reduced leakage rates substantially decrease the pressures at measurement points in the renormalized overlying formation.

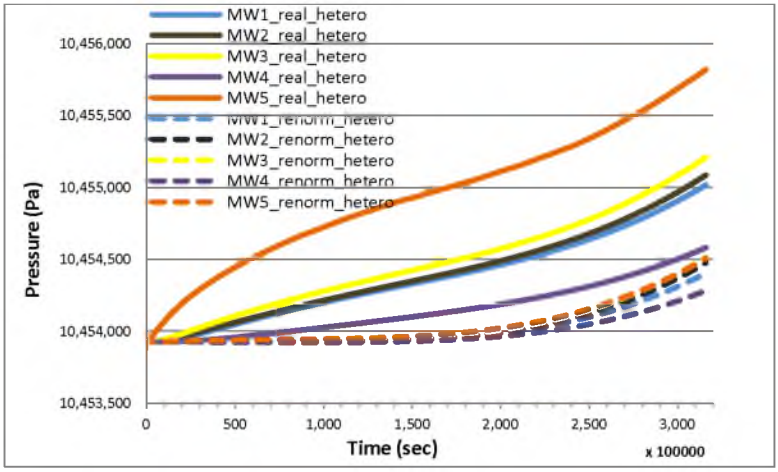


Fig. 4.19 Pressure drifts between the original and renormalized heterogeneous domains at measurement points in the overlying formation.

The systematic modeling error through renormalized permeability values may be related to poor estimation of the leakage pathway. This is because the underestimated heterogeneity by renormalization will result in errors of calculated pressures. The heterogeneity inversion includes two simulation scenarios. In the first, the inversion estimates only the actual (one) leakage location based on underestimated permeability by the renormalization. The leakage pathway location is estimated through calibrating vertical permeability of initial guesses. Second, this inversion estimates an optimum combination of both the vertical permeability of the initial guesses of the leakage pathway and the eight permeability zones grouped in both aquifers. This case identifies that estimating the renormalized permeability in both reservoirs can improve the accuracy of the leakage pathway estimation.

As mentioned in section 4.1.1, the weighting coefficients (σ_z) should be used to scale the magnitude of measurements. Here, two weighting coefficients, 1,000 Pa and 10,000 Pa, are assigned for measurements in the storage formation in each scenario, but only one weighting coefficient (10 Pa) is assigned for the overlying formation in both scenarios. Pressure profiles from nine monitoring wells in the overlying and storage formations of the original (nonrenormalized) domain are used as observed measurement data.

The first inverse analysis ("case 1") for leakage pathway estimation identifies the impact of systematic error of permeability distribution induced by renormalization. In this case, only initial guesses of vertical permeability values in the leakage pathway are calibrated. The methodology for inversion is the same as that of the first scenario for homogeneous condition of section 4.1. The inversion is applied to the two-dimensional

model domain of Fig. 4.16 (b) and (d).

In Fig. 4.19, the pressure perturbation induced by leaks in the overlying formation reach the 5th monitoring well first (orange colored solid line). The magnitude of the pressure perturbation is also largest at the 5th monitoring well, so the leakage well may be closest to the 5th monitoring well. Based on the observation in the overlying formation, initial guesses of the leakage pathway are assigned around the 5th monitoring well with more points.

The geometry of 30 initial guesses with 0.3 m × 0.3 m was meshed in the model domain. During the inversion, the logarithm of absolute vertical permeability of each initial guess was iteratively estimated.

Fig. 4.20 presents two contour plots of the objective function values for two inversions with different weighting coefficients (σ_z). Fig. 4.20 (a) is the objective function distribution when σ_z = 10 Pa and 1,000 Pa are weighted for measurements of the overlying and storage formations, respectively; Fig. 4.20 (b) is the resulting objective function distribution from σ_z = 10 Pa and 10,000 Pa for each formation. In Fig. 4.20 (a), the contour plot has one global minimum (red filled circle) and five local minima (black filled circles), and the inversion is well-posed (stable); an ill-posed inverse problem leads poorly to the minimum by displaying level plains, long narrow valleys, etc. (Finsterle, 2007a). However, the objective function of five local minima is close to that of the global minimum (less 3%) as seen in Table 4.10, so this inversion can be considered as nonunique. That is, it can be difficult to evaluate the most likely leakage well location from this inversion because the leakage well can exist anywhere approximately around six minima.

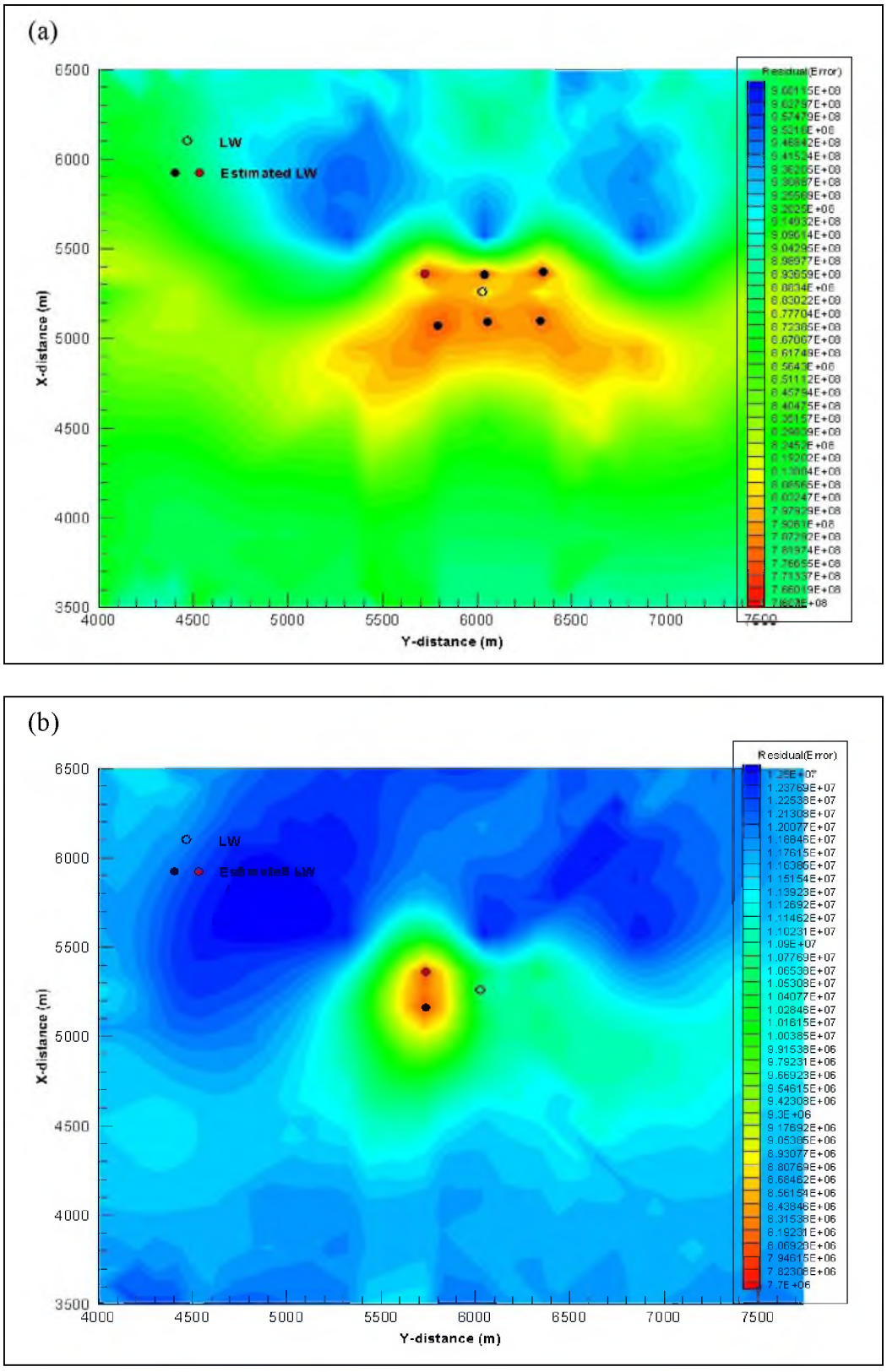


Fig. 4.20 Simulation results from objective function in case 1 for (a) σ_z = 10 Pa and 1,000 Pa for each formation and (b) σ_z = 10 Pa and 10,000 Pa for each formation.

Table 4.10 Estimated parameters at six minima of Fig. 4.20 (a).

	X (m)	Y (m)	Deviation (m)	Objective function values
True	**5250**	**6050**	-	-
1st minimum	5150	5750	316	0.7780e9
2nd minimum	5150	6050	100	0.7808e9
3rd minimum	5150	6350	316	0.7810e9
4th minimum *	**5350**	**5750**	**316**	**0.7601e9**
5th minimum	5350	6050	100	0.7629e9
6th minimum	5350	6350	316	0.7630e9

* value is a best estimation.

Fig. 4.20 (b) presents one global minimum and one local minimum, and is also stable and nonunique. However, the result of Fig. 4.20 (b) is relatively improved compared to that of Fig. 4.20 (a). The possible leakage pathway in Fig. 4.20 (b) can be limited to a minimized area of objective function around two minima. Table 4.10 and Table 4.11 denote the objective functions and deviations of estimated leakage locations at six minima and two minima, corresponding to Fig. 4.20 (a) and Fig. 4.20 (b), respectively.

Table 4.11 Estimated parameters at two minima of Fig. 4.20 (b).

	X (m)	Y (m)	Deviation (m)	Objective function values
True	**5250**	**6050**	-	-
1st minimum	5150	5750	316	0.7850e7
2nd minimum *	**5350**	**5750**	**316**	**0.7746e7**

* value is a best estimation.

The possible leaky area is significantly smaller than that of Fig. 4.20 (a) although the most likely point (global minimum) has a deviation of 316 m from the actual location. This result suggests that the values of weighting coefficients can substantially affect estimations of the leakage pathway. Even if the first scenario was performed based on the systematic error associated with renormalized permeability, the two estimated leakage wells in the first scenario (b) are calibrated within 316 m from the actual leakage well.

The second scenario ("case 2") consists of evaluating the leakage pathway location and the renormalized permeability in both formations. The methodology of inverse modeling is the same as case 1. Moreover, coupled weighting coefficients $\sigma_z =$ 10 Pa and 1,000 Pa are assigned to measurements of each overlying and storage formation in case 2 (a), as well as $\sigma_z = 10$ Pa and 10,000 Pa for measured data of each formation in case 2 (b). Resulting contour plots of the objective function of case 2 (a) and (b) are shown in Fig. 4.21.

In Fig. 4.21 (a), the inversion using $\sigma_z =$ 10 Pa and 1,000 Pa does not yield a global minimum, i.e., the result is an ill-posed inversion. On the other hand, in Fig. 4.21 (b), the inverse results for $\sigma_z = 10$ Pa and 10,000 Pa yielded one stable global minimum for the leakage pathway. Similar to the results in case 1, case 2 indicates that the inversion is improved when $\sigma_z = 10$ Pa and 10,000 Pa are used for the overlying and storage formations, respectively. The large amount of injected CO_2 further increases pressures in the storage formation, so the residuals between measured and calculated pressures in the storage formation can be much larger than those in the overlying formation. The residuals must be scaled accordingly. A detailed discussion of the ill-posed inversion of case 2 (a) is described in the next residual analysis section.

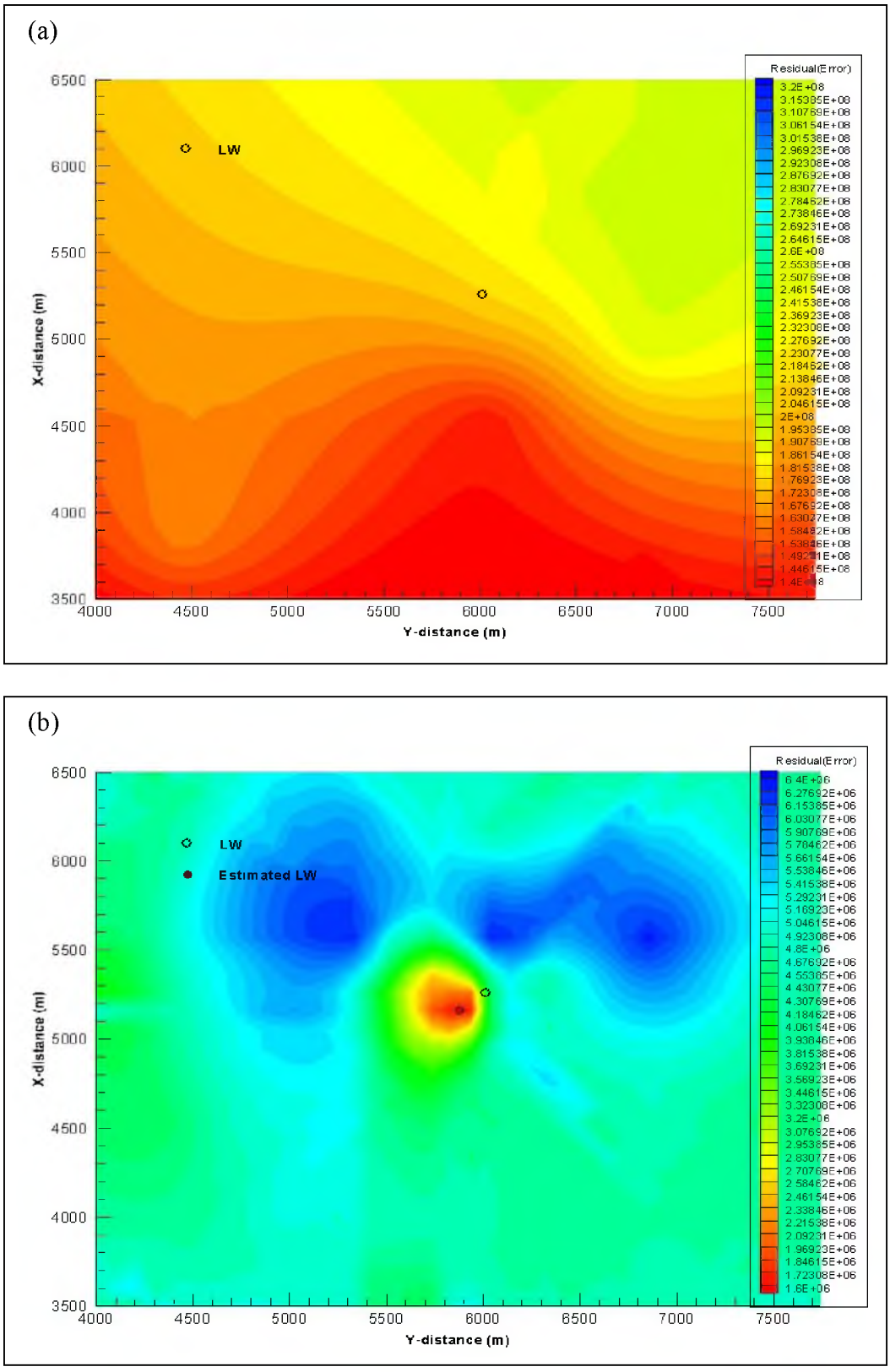


Fig. 4.21 Simulation results from objective function in case 2 for (a) $\sigma_z = 10$ Pa and 1,000 Pa for each formation and (b) $\sigma_z = 10$ Pa and 10,000 Pa for each formation.

The possible leaky area (minimized objective function area) from case 2 (b) of Fig. 4.21 is smaller than that for case 1 (b). Furthermore, the objective function of case 2 (b) is improved through calibrating renormalized permeability values (of the eight groups) in comparison to that of case 1 (b). This study suggests that calibration of renormalized permeability values in the inversion reduces systematic modeling errors. The statistics of estimated parameters in case 2 (b) are shown in Table 4.12.

Table 4.12 Statistics of estimated parameters for simulation case 2 (b).

	Estimated LW locations (m)			**Logarithm of estimated permeability of initial guesses (m^2)**		**Logarithm of estimated permeability of 8 groups (m^2)**	
	X	**Y**	**Deviation**	**Average**	**Std. dev.**	**Average**	**Std. dev.**
True	5250	6050	-	-10.0	-	-12.60	-
						-13.12	-
						-13.60	-
						-14.12	-
						-14.60	-
						-15.12	-
						-15.60	-
						-16.12	-
Case 2 (b)	5150	5850	223	-10.52	1.26	-12.51	0.144
						-12.61	0.220
						-13.11	0.156
						-14.50	0.000
						-14.51	0.046
						-15.50	0.000
						-15.05	0.071
						-15.57	0.098

4.2.1 Residual Analysis

Fig. 4.22 summarizes residuals (vector $\mathbf{r} = \mathbf{z}^* - \mathbf{z(p)}$) of the best estimates for measured pressures (vector $\mathbf{z}^*$) vs. calculated pressures (vector $\mathbf{z(p)}$) in the storage formation for both case 1 (b) and case 2 (b). Fig. 4.23 presents the residuals for the overlying formation for both case 1 (b) and case 2 (b). In Fig. 4.22, the residuals of case 2 (b) are reduced compared to those for case 1 (b), because case 2 (b) estimates permeability values (of the eight renormalized groups) to obtain a more minimized objective function.

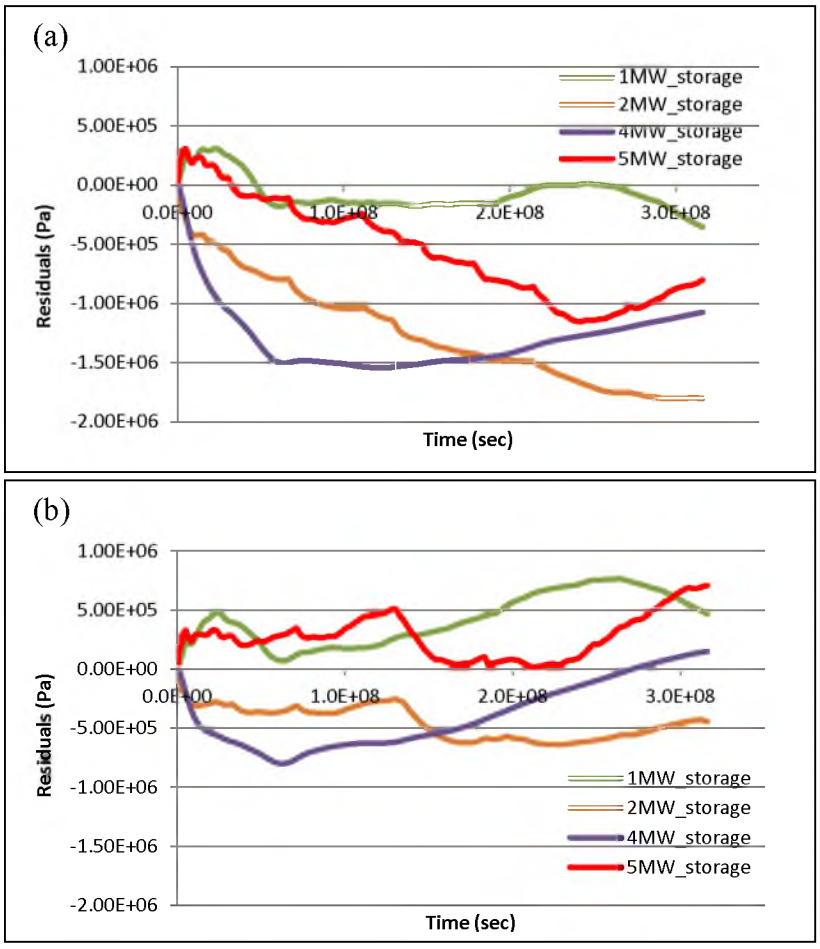


Fig. 4.22 Residuals between measured and calculated pressures in the storage formation for (a) case 1 (b), and (b) case 2 (b).

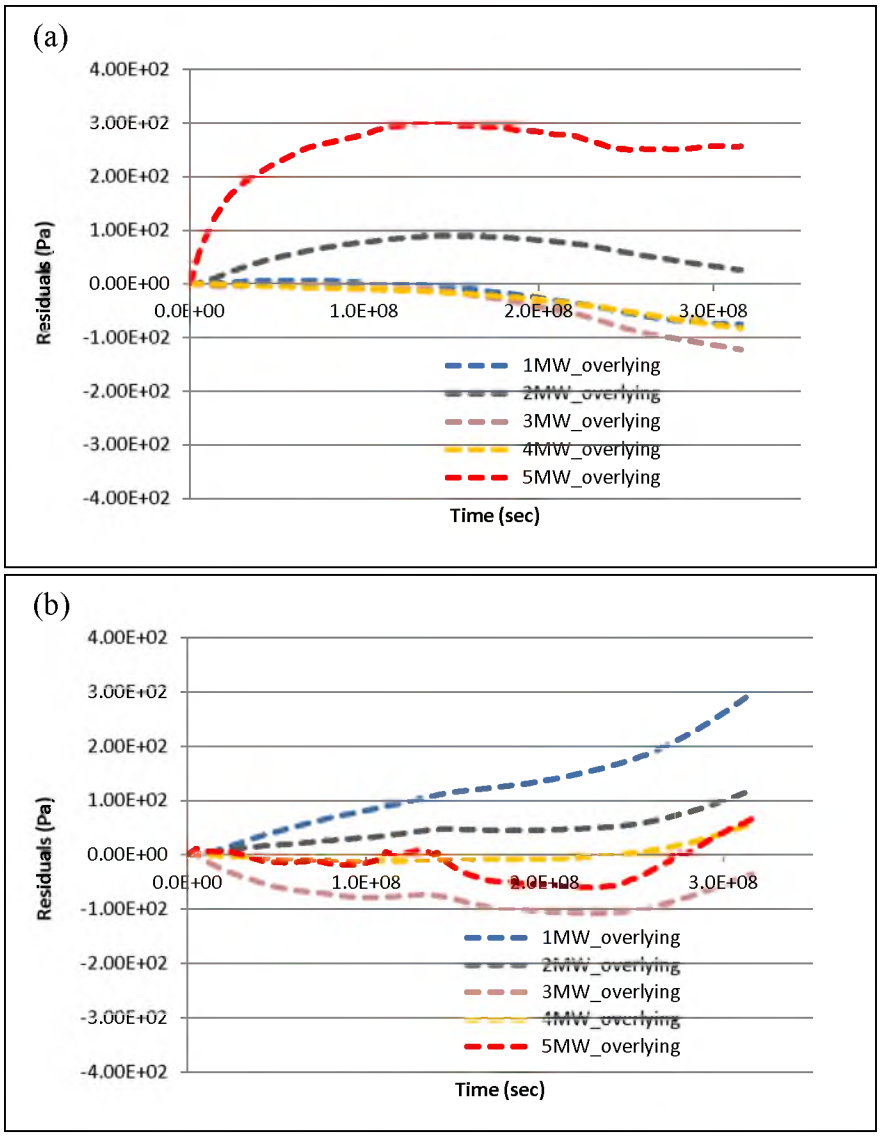


Fig. 4.23 Residuals in the overlying formation for (a) case 1 (b) and (b) case 2 (b).

In fact, the inversions of the homogeneous model of section 4.1 do not estimate the uncertainty of the permeability values in the storage formation (correct permeability is assigned to the storage formation), and as such the residuals in the storage formation are not improved. On the other hand, the heterogeneous model residuals in the storage formation are reduced because the erroneous permeability values of eight groups are estimated. In Fig. 4.23, the residuals in the overlying formation of case 2 (b) are less than

those of case 1 (b), especially at MW5. That is, calibration of permeability values (to reduce systematic error from renormalization) improves the accuracy of estimation of leakage pathway location.

We infer why case 2 (a) becomes an ill-posed inversion by residual analysis. From Fig. 4.22 (b) and Fig. 4.23 (b), the degrees of residuals in case 2 are approximately ~100,000s Pa and ~100s Pa in the storage and the overlying formations, respectively. The magnitude of residuals in the storage formation is approximately 1,000 times larger than that in the overlying formation. Thus, σ_Z in the storage formation should be 1,000 times larger compared to that in the overlying formation to scale both residuals. Since the weighting coefficients in case 2 (a) are assigned as σ_Z = 1,000 Pa and 10 Pa for the storage and overlying formation, respectively, the magnitude of the objective function values ($S=\sum_{i=1}^{m}\frac{r_i^2}{\sigma_{Z_i}^2}$) in the storage formation are 100 times larger than that of the overlying formation. Thus, the inversion will minimize the objective function values in terms of the storage formation (i.e., the inverse modeling minimizes the objective function to estimate an optimum parameter value set of the eight permeability groups in the storage formation). As mentioned in section 4.1.1, the critical information required to estimate the leakage pathway are pressure anomalies in the overlying formation induced by leaks (see Fig. 4.9). Therefore, the inversion of case 2 (a) becomes ill-posed. Higher magnitude residuals for the storage formation must be scaled to be approximately equivalent to the degree of residuals in the overlying formation for successful model calibration.

4.2.2 Future Tasks for the Heterogeneous Field

So far the heterogeneous field inversion improves parameter estimation accuracy by reducing systematic error from renormalized permeability, if employing the assumption that the permeability distributions are known. On the other hand, some other inversion methods to characterize or estimate subsurface heterogeneities have been studied.

The geostatistical approach and pilot point method are popularly applied to generate realizations of heterogeneity in geological fields (Finsterle, 2004). The pilot point method incorporated with inverse modeling and geostatistics was applied to estimate heterogeneity by Kowalsky et al. (2004). The pilot point method generates mapping of heterogeneity on pilot points, and grid blocks in the vicinity of the pilot points, by interpolation (or the geostatistical approach) using permeability values measured from several boreholes. In inverse analysis, permeability values at the pilot points are parameters to be estimated, and permeability values in the vicinity of the pilot points are automatically generated by the interpolation method (or the correlation length from geostatistics). The inverse modeling modifies the parameter values at pilot points until the fit is improved between the measured and calculated data.

Methods can be incorporated with a leakage well location estimation to improve the match between calculated output and measured data. However, discussing applicability of those methods to leakage estimation is beyond the scope of this dissertation. Therefore, it remains as a future task to improve the solution of leak detection in the heterogeneous field.

4.3 Summary and Conclusion

Parameter estimation by inversion (using the iTOUGH2 code) was applied to detect locations of leakage pathways by calibrating the absolute permeability of initial guesses of the leakage pathways in the homogeneous and heterogeneous conceptual domains in Chapter 3.

For homogeneous models, the idealized scenario demonstrated the detectability of inverse analysis for the leakage well. The second scenario showed the impact of uncertainty on the inverse solution. The third scenario analysis found that parameterization of uncertain overlying formation permeability could improve location estimation accuracy. Residual analysis illustrated that pressure anomalies in the overlying formation induced by leaks are critical information for this type of inverse analysis. Weighting factors, if appropriate, should be assigned to improve inversion results. In the first additional inversion, uncertain leakage pathway size could reduce leakage pathway estimation accuracy. The second additional inversion illustrated leakage pathway location can be detected from measurements with random noises by 0.1 %. However, more case studies will be conducted by measurements including various magnitudes of random noises. Moreover, if needed, the noise filtering method will be considered to reduce or stabilize random errors.

The inverse analysis was applied by the SACROC unit in Texas, for sake of a heterogeneous case study. The general approach of inverse modeling for leakage pathway estimation is based on the assumption of known heterogeneity. The upscaling technique (renormalization) was introduced for the general modeling approach. The systematic error of renormalized permeability values can cause an incorrect estimation of leakage

pathway location. Thus, the eight groups of renormalized heterogeneity of overlying and storage formations were parameterized in the inversion to reduce the impact of systematic error from renormalization. It was identified that the calibration of renormalized permeability values can reduce systematic modeling errors from renormalization and should improve the leakage pathway location estimation accuracy. On the other hand, the inversion with $\sigma_z = 10$ Pa and 1,000 Pa weighting coefficients for the overlying and storage formation, respectively, leads to an ill-posed inversion. The inversion was improved when weighting coefficients of 10 Pa and 10,000 Pa were used for the overlying and storage formations. The higher magnitude of residuals in the storage formation must be scaled to be less than the degree of residuals in the overlying formation for successful leakage pathway estimation.

4.4 References

Altevogt, A. S., and Celia, M. A. (2004). "Numerical modeling of carbon dioxide in unsaturated soils due to deep subsurface leakage." *Water Resources Research*, 40(3).

Altunin, V. (1975). *Thermophysical properties of carbon dioxide*, Publishing House of Standards, Moscow, Russia.

Anderson, M. P., and Woessner, W. W. (1992). *Applied groundwater modeling: simulation of flow and advective transport*, Elsevier Science, Oxford, UK.

Avci, C. B. (1994). "Evaluation of flow leakage through abandoned wells and boreholes." *Water Resources Research*, 30(9), 2565-2578.

Babbar, M., and Minsker, B. (2006). "Groundwater remediation design using multiscale genetic algorithms." *Journal of Water Resources Planning and Management*, 132(5), 341-350.

Beckford, O., Hilton, A. C., and Liu, X. "Development of an enhanced multi-objective robust genetic algorithm for groundwater remediation design under uncertainty." *Proc., Proc. Am. Soc. Civ. Eng*, 98.

Benjamin, J. R. (1970). *Probability, statistics, and decision for civil engineers*, McGraw-Hill, New York, NY.

Bennett, G. D. (1976). *Introduction to groundwater hydraulics*, U.S. Geological Survey, Denver, CO.

Bevington, P., and Robinson, D. (1969). *Data reduction and data analysis for the physical sciences*, McGraw-Hill, New York, NY.

Carroll, S., Hao, Y., and Aines, R. (2009). "Transport and detection of carbon dioxide in dilute aquifers." *Energy Procedia*, 1(1), 2111-2118.

Chan-Hilton, A. B., Iyer, S. K., Magar, V., and Kelley, M. "Optimization of natural attenuation with active remediation under uncertainty." *Proc., Seventh International In Situ and On-Site Bioremediation Symposium, Orlando, Florida, USA, 2-5 June 2003. Part H. Natural Attenuation, Long-Term Monitoring, and Site Closure*, Battelle Press, Columbus, OH.

Chen, C. S. (1989). "Solutions approximating solute transport in a leaky aquifer receiving wastewater injection." *Water Resources Research*, 25(1), 61-72.

Chen, Z., Huan, G., and Ma, Y. (2006). *Computational methods for multiphase flows in porous media*, Siam, Philadelphia, PA.

Christensen, S., and Cooley, R. (1999). "Evaluation of confidence intervals for a steady-state leaky aquifer model." *Advances in Water Resources*, 22(8), 807-817.

Cihan, A., Zhou, Q., and Birkholzer, J. T. (2011). "Analytical solutions for pressure perturbation and fluid leakage through aquitards and wells in multilayered-aquifer systems." *Water Resources Research*, 47(10), W10504.

Cobb, P., McElwee, C., and Butt, M. (1982). "Analysis of leaky aquifer pumping test data: An automated numerical solution using sensitivity analysis." *Groundwater*, 20(3), 325-333.

Committee, I. F. (1967). "A formulation of the thermodynamic properties of ordinary water substance." *IFC Secretariat, Düsseldorf, Germany*, 26.

Corey, A. T. (1954). "The interrelation between gas and oil relative permeabilities." *Producers Monthly*, 19(1), 38-41.

Deming, D. (1994). "Factors necessary to define a pressure seal." *AAPG bulletin*, 78(6).

Doughty, C., and Pruess, K. (2004). "Modeling supercritical carbon dioxide injection in heterogeneous porous media." *Vadose Zone Journal*, 3(3), 837-847.

Espinoza, F. P., Minsker, B. S., and Goldberg, D. E. (2005). "Adaptive hybrid genetic algorithm for groundwater remediation design." *Journal of Water Resources Planning and Management*, 131(1), 14-24.

Finsterle, S. (2004). "Multiphase inverse modeling." *Vadose Zone Journal*, 3(3), 747-762.

Finsterle, S. (2007a). "iTOUGH2 user's guide." *Report LBNL-40040*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007b). "iTOUGH2 command reference." *Report LBNL-40041*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007c). "iTOUGH2 sample problems." *Report LBNL-40042*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2010). "iTOUGH2 V3. 2, verification and validation report." *Report LBNL-42002*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S., Moridis, G., and Pruess, K. (1994). "A tough2 equation-of-state module for the simulation of two-phase flow of air, water, and a miscible gelling liquid." *Report LBL-36086*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Garcia, J. E. (2001). "Density of aqueous solutions of CO2." *Report LBNL-49023*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Gasda, S. E., Bachu, S., and Celia, M. A. (2004). "Spatial characterization of the location of potentially leaky wells penetrating a deep saline aquifer in a mature sedimentary basin." *Environmental Geology*, 46(6-7), 707-720.

Gasda, S. E., Wang, J. Z., and Celia, M. A. (2011). "Analysis of in-situ wellbore integrity data for existing wells with long-term exposure to CO2." *Energy Procedia*, 4, 5406-5413.

Han, W. S., McPherson, B. J., Lichtner, P. C., and Wang, F. P. (2010). "Evaluation of trapping mechanisms in geologic CO2 sequestration: Case study of SACROC northern platform, a 35-year CO2 injection site." *American Journal of Science*, 310(4), 282-324.

Hou, Z., Murray, J. C., and Rockhold, L. M. (2012). "CO2 migration in intact caprock and leakage risk in three-dimensional heterogeneous formations." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittssburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012a). "Early detection of brine or CO2 leakage through high-permeability pathways using pressure-based monitoring data." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittsburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012b). "Impact of data uncertainty on identifying leakage pathways in CO2 geologic storage systems and estimating their hydrogeological properties by inverse modeling." *TOUGH Symposium 2012,* Lawrence Berkeley National Laboratory, Berkeley, CA.

King, P. (1989). "The use of renormalization for calculating effective permeability." *Transport in Porous Media*, 4(1), 37-58.

Ko, N.-Y., and Lee, K.-K. (2008). "Reliability and remediation cost of optimal remediation design considering uncertainty in aquifer parameters." *Journal of Water Resources Planning and Management*, 134(5), 413-421.

Kowalsky, M. B., Finsterle, S., and Rubin, Y. (2004). "Estimating flow parameter distributions using ground-penetrating radar and hydrological measurements during transient flow in the vadose zone." *Advances in Water Resources*, 27(6), 583-599.

Krevor, S., Perrin, J.-C., Esposito, A., Rella, C., and Benson, S. (2010). "Rapid detection and characterization of surface CO2 leakage through the real-time measurement of δ13 C signatures in CO2 flux from the ground." *International Journal of Greenhouse Gas Control*, 4(5), 811-815.

Liggett, J. A., and Chen, L.-C. (1994). "Inverse transient analysis in pipe networks." *Journal of Hydraulic Engineering*, 120(8), 934-955.

Metz, B., Davidson, O., de Coninck, H., Loos, M., and Meyer, L. (2005). *IPCC special report on carbon dioxide capture and storage*, Cambridge University Press, New York, NY.

Nogues, J. P., Nordbotten, J. M., and Celia, M. A. (2011). "Detecting leakage of brine or CO2 through abandoned wells in a geological sequestration operation using pressure monitoring wells." *Energy Procedia*, 4, 3620-3627.

Nordbotten, J. M., Celia, M. A., and Bachu, S. (2004). "Analytical solutions for leakage rates through abandoned wells." *Water Resources Research*, 40(4), W04204.

Nordbotten, J. M., Kavetski, D., Celia, M. A., and Bachu, S. (2008). "Model for CO2 leakage including multiple geological layers and multiple leaky wells." *Environmental Science & Technology*, 43(3), 743-749.

Onuma, T., and Ohkawa, S. (2009). "Detection of surface deformation related with CO2 injection by DInSAR at In Salah, Algeria." *Energy Procedia*, 1(1), 2177-2184.

Phillips, S. L., Igbene, A., Fair, J., Ozbek, H., and Tavana, M. (1981). *A technical databook for geothermal energy utilization*, Lawrence Berkeley Laboratory, University of California, Berkeley, CA.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in Fortran 77: The art of scientific computing second edition*, Cambridge University Press, New York, NY.

Pruess, K. (1987). "TOUGH user's guide, nuclear regulatory commission report NUREG/CR-4645." *Report LBL-20700*, Lawrence Berkeley Laboratory Berkeley, CA.

Pruess, K. (1991). "TOUGH2: A general-purpose numerical simulator for multiphase fluid and heat flow." *Report LBL-29400*, Lawrence Berkeley Laboratory, Berkeley, CA.

Pruess, K. (2004). "Numerical simulation of CO2 leakage from a geologic disposal reservoir, including transitions from super-to subcritical conditions, and boiling of liquid CO2." *Spe Journal*, 9(2), 237-248.

Pruess, K. (2005). "ECO2N: A TOUGH2 fluid property module for mixtures of water, NaCl, and CO2." *Report LBNL-57952*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Pruess, K., and García, J. (2002). "Multiphase flow dynamics during CO2 disposal into saline aquifers." *Environmental Geology*, 42(2-3), 282-295.

Pruess, K., Moridis, G., and Oldenburg, C. (1999). "TOUGH2 user's guide, version 2.0." *Report LBNL-43134*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Rao, S. S. (2009). *Engineering optimization: Theory and practice*, John Wiley & Sons, Hoboken, New Jersey.

Singh, S. K. (2009). "Simple method for quick estimation of leaky-aquifer parameters." *Journal of Irrigation and Drainage Engineering*, 136(2), 149-153.

Spycher, N., and Pruess, K. (2005). "CO2-H2O mixtures in the geological sequestration of CO2. II. Partitioning in chloride brines at 12–100° C and up to 600 bar." *Geochimica et Cosmochimica Acta*, 69(13), 3309-3320.

Sun, A. Y., Zeidouni, M., Nicot, J.-P., Lu, Z., and Zhang, D. (2013). "Assessing leakage detectability at geologic CO2 sequestration sites using the probabilistic collocation method." *Advances in Water Resources*, 56, 49-60.

Van Genuchten, M. T. (1980). "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils." *Soil Science Society of America Journal*, 44(5), 892-898.

Zheng, C., Bennett, G. D., Melton, J., and Simon, A. R. (2002). "Applied contaminant transport modeling." *Industrial and Commercial Training*, 34(7), 256-262.

Zhou, Q., Birkholzer, J. T., and Tsang, C.-F. (2009). "A semi-analytical solution for large-scale injection-induced pressure perturbation and leakage in a laterally bounded aquifer–aquitard system." *Transport in Porous Media*, 78(1), 127-148.

CHAPTER 5

SINGLE-PHASE FLOW SIMULATION

A simultaneous solution model, based on the Finite Difference Method (FDM), is developed for three-dimensional forward analysis of transient flows in arbitrary (general) groundwater aquifers with leakage. The forward method is combined with a genetic algorithm (GA) of optimization to search unknown model parameters and locations of leakage zones.

The forward model simultaneously calculates leakage rates based on hydraulic gradients between coupled leakage points in two leaky aquifers, and evaluates propagation of hydraulic head in multiple aquifers resulting from that leakage. A leakage term was added to the groundwater flow governing equation, specifically to realize associated hydraulic head anomalies. For this model, it is assumed that leaks flow vertically in confining beds along specific leakage pathways.

In the inverse model, the important consideration is that the cross-sectional area and vertical hydraulic conductivity of a leakage pathway are integrated as a single parameter inherent to a leakage term. This indicates a parameterization of the leakage pathway properties. The inverse model evaluates possible locations of leakage pathways by estimating the integrated parameters between coupled leakage points of initial guesses. Estimation of integrated parameters can provide three advantages for effective inverse

modeling: (1) reducing the number of required grid blocks, (2) decreasing the impact of uncertainty of geometry of the leakage pathways and (3) reducing the number of parameters to be estimated.

Furthermore, two kinds of leakage pathways are specified in terms of the generated time, including: (1) pre-existing leakage pathways and (2) abruptly-induced leakage pathways at specific times. The governing equation with its leakage term is composed of three finite difference equations, and its form depends on whether the cells reflect coupled leakage points at the time of interest.

5.1 Conceptual Framework and Model

The developed model for forward simulation in the single-phase system is a simultaneous solution model for three-dimensional analysis of transient flow induced by leakage in an arbitrary groundwater aquifer(s) with anisotropic, heterogeneous and isothermal conditions. The developed forward model uses the FDM and is designed to realize the propagation of groundwater through many possible leakage zones in porous media, such as fractures and abandoned wells. The developed inverse model, which consists of the forward model and the GA, is designed specifically to estimate possible leakage zones in the single-phase system. In fact, this model was developed for this research as an initial, single-phase analysis in a broader research program focused on storage of CO_2 in geological formations. An ultimate goal is to develop a full, multiphase method to simulate CO_2 storage with leakage pathways. However, in this chapter, the developed forward and inverse models are applied to multiphase fluid system of mobile brine into or out of the confined aquifer through leakage pathways.

As mentioned in Chapter 3, when CO_2 injection starts in a reservoir that is saturated with brine, the increased pressure gradient by injection continuously invokes brine discharge through the leakage pathway. Before CO_2 leaks into an overlying formation, the overlying formation is a single-phase reservoir, at least in most situations. This study evaluates applicability of leakage pathway detection using a single-phase model. If the single-phase inverse model can be applied to a multiphase formation with leakage, it should provide at least three advantages. Firstly, the inverse modeling would exclude errors in capillary pressure and relative permeability functions. Secondly, computational expense is relatively reduced, since the forward model for single-phase flow is much simpler. Lastly, estimates based on only brine leaks may provide an early warning before CO_2 leakage.

In terms of forward analysis, confined aquifers with leakage can be simulated using a leakage term added to the three-dimensional flow equation (Nordbotten et al., 2004; Nordbotten et al., 2008; Zhou et al., 2009; Cihan et al., 2011). In this dissertation study, a leakage term based on Darcy's law was included in the groundwater flow governing equation to realize hydraulic head anomalies induced by leaks in multiple aquifers. Consistent with Darcy's law, leakage rates depend on hydraulic conductivity of the leakage pathway, the cross-sectional area of leakage pathway and the hydraulic gradient between the two aquifers overlying and underlying the confining bed. The hydraulic gradient between the leakage aquifers is calculated from coupled leakage points at each end of the leakage pathway. Hydraulic head at one leakage point is calculated from one of the explicitly derived two finite difference equations based on hydraulic heads at six discrete adjacent nodes of FDM and one corresponding leakage point. So the

inflow and outflow rates along the leakage pathway through the confining bed are calculated, and the hydraulic head changes (due to leakage) within the two aquifers can be modeled simultaneously. This methodology offers additional advantages with respect to computational expense. In the conventional methodology described in Chapter 3, the geometry of the leakage pathway should be meshed explicitly in the domain, but this method with a leakage term does not require that leakage pathways be specifically designated by nodes in the mesh (Nordbotten et al., 2004). In addition, when "abrupt" leakage pathways arise within confining beds (e.g., by external forces like increased pressure from injection), resulting propagation of hydraulic head changes between the two aquifers (reservoirs) can be simulated.

The inverse model evaluates possible locations of leakage pathways by estimating a parameter that incorporates both hydraulic conductivity and cross sectional area of initial guesses of the leakage pathways. This method of analyzing possible leakage pathways is distinct from the inverse methodology in Chapter 4. In Chapter 4, the inversion considered only vertical permeability values of initial guesses specified by grid blocks to identify possible leakage pathways. This kind of inversion is likely to reduce accuracy of leakage pathway estimation because of the uncertainty of size of leakage pathway. (This problem will be specifically discussed in section 5.5) On the other hand, the parameterized cross sectional area with vertical hydraulic conductivity of the leakage pathway can be further effective in the inversion. The geometry of leakage pathway does not need to be meshed in domain as well as it is simultaneously calibrated with vertical hydraulic conductivity. Therefore, this method implicitly reduces impact of leakage pathway size uncertainty.

The next section discusses the mathematical background for the developed forward and inverse simulator.

5.2 Governing Equation

The governing equation is based on the mass conservation equation for the three-dimensional movement of groundwater through porous media. Fluid density is assumed uniform and constant. The governing equation may be described by a partial-differential equation as follows:

$$\frac{\partial}{\partial x}(Kx\frac{\partial h}{\partial x})+\frac{\partial}{\partial y}(Ky\frac{\partial h}{\partial y})+\frac{\partial}{\partial z}(Kz\frac{\partial h}{\partial z})+W-L=Ss\frac{\partial h}{\partial t}, \tag{5.1}$$

where Kx, Ky, and Kz are values of hydraulic conductivity (L/T); h is the hydraulic head (L); W is a volumetric flux per unit volume representing sources and/or sinks of water (T^{-1}); L is the leakage term, the leakage rate per unit volume (T^{-1}); Ss is the specific storage coefficient of the porous media (L^{-1}); and t is time (T).

Equation (5.1) describes groundwater flow with a leakage term for a heterogeneous and anisotropic medium. The relationship between velocities and flow rates of groundwater is given by Darcy's law:

$$Qx=Kx\left(\frac{\Delta h}{\Delta x}\right)(\Delta y\Delta z),\ Qy=Ky\left(\frac{\Delta h}{\Delta y}\right)(\Delta x\Delta z)\ \text{and}\ Qz=Kz\left(\frac{\Delta h}{\Delta z}\right)(\Delta x\Delta y), \tag{5.2}$$

where Qx is x-directional discharge in the cell, Qy is y-directional discharge and Qz is z-directional discharge; $(\Delta h/\Delta x)$, $(\Delta h/\Delta y)$, and $(\Delta h/\Delta z)$ are the hydraulic gradients for each flow direction of the cell; $(\Delta y\Delta z)$, $(\Delta x\Delta z)$, and $(\Delta x\Delta y)$ indicate the area of the cell faces perpendicular to each flow direction.

5.2.1 Leakage Term

As mentioned in Chapter 3 and Chapter 4, leakage induces anomalies of hydraulic heads in the aquifers. To realize those hydraulic head anomalies, the leakage term has been added in the governing equation. In the same way, multiple aquifers should be considered to simulate a confined system with possible leakage pathways (see Fig. 3.1). As shown in Fig. 3.1, the leakage rate depends on the hydraulic gradient between the overlying and storage aquifers, and both the hydraulic conductivity and the cross sectional area of leakage pathway.

The leakage rate can change with time (because leakage also directly affects the hydraulic heads in the overlying and storage aquifers). So, the leakage rates should be simultaneously solved based on the hydraulic heads in the overlying and storage aquifers as well as the transient flow in the two aquifers due to the fact that leakage has to be calculated.

To simplify simulations, leakage flows in the pathways are only one-dimension (vertical). There is no recharge of water from the confining bed into the leakage pathway or discharge of water from the leakage pathway into the confining bed. In addition, the leakage pathway is not deformable and the liquid is incompressible; therefore, the continuity equation is satisfied in the leakage pathway and the instantaneous leakage rate is the same in all sections of the leakage pathway.

As introduced earlier, Darcy's law is used to model the leakage term (Anderson and Woessner, 1992):

$$Qleak_{i,j,k} = Kzbleak_{i,j,k} \frac{Aleak_{i,j,k}}{Dzbleak(I)} \left(h_{i,j,zls(I)} - h_{i,j,zlsu(I)}\right), \qquad (5.3)$$

where I is the leakage column (or pathway) number, $Aleak_{i,j,k}$ is the leakage area, $Dzbleak(I)$ is the length of I-th leakage pathway, $Kzbleak_{i,j,k}$ is the z-directional hydraulic conductivity of the leakage pathway, $zls(I)$ is the z-coordinate at a storage aquifer of I-th leakage and $zlsu(I)$ is the z-coordinate at an overlying aquifer of I-th leakage. The leakage rate (L) in Equation (5.1) is expressed as $Qleak_{i,j,k}$ per unit volume.

Fig. 5.1 depicts how the leakage pathways and parameters are related. Node (i, j, zlsu(1)) is coupled with node (i, j, zls(1)) to specify the 1st leakage pathway. Thus, the leakage rate at the 1st leakage pathway is calculated by the difference of hydraulic heads (h) between coupled leakage points (i, j, zls(1)) and (i, j, zlsu(1)). In the same way, the leakage rate at the 2nd leakage pathway is calculated using the heads at the coupled leakage points (i, j+2, zls(2)) and (i, j+2, zlsu(2)).

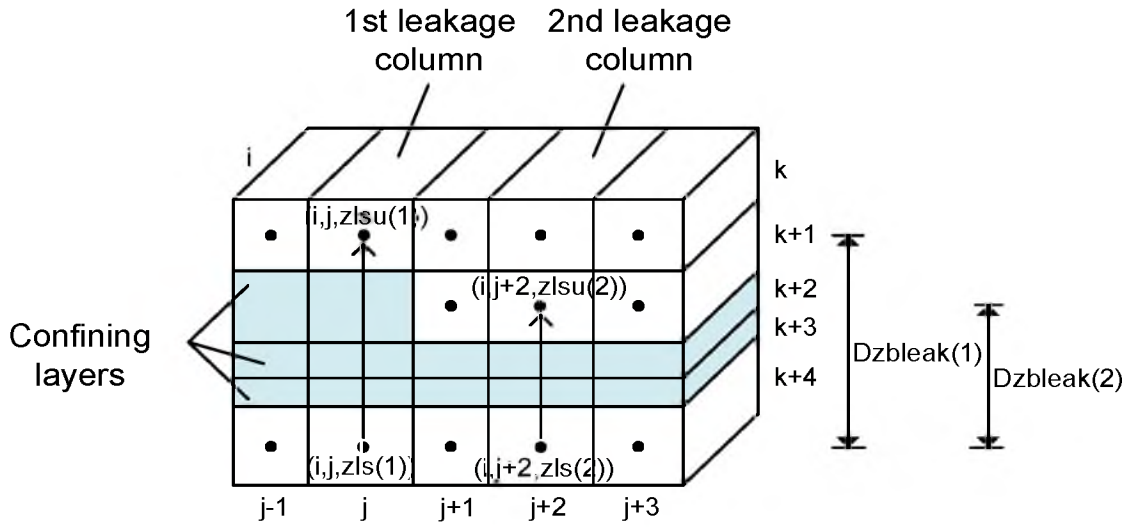


Fig. 5.1 Schematic leakage pathways and associated parameters.

5.2.2 Leakage Features

One of the considerations for leakage simulation is variability of leakage rate with time and leakage starting time. As mentioned above, the leakage rate and hydraulic head propagation are simulated explicitly.

However, timing of leakage through confining beds may be uncertain. If the confined aquifers exhibit pre-existing leakage pathways (completely saturated), leakage will coincide with water from the storage reservoir to the reservoir on the opposite side of the confining bed. Increased pressure from water injection can cause fractures or cracks in confining beds, and leakage pathways can be abruptly induced. Thus leakage can be induced at any specific time. This study does not consider specific mechanisms for how leakage pathways are generated, but rather only on migration of leakage.

Two kinds of leakage pathways are specified, including (1) pre-existing leakage pathways and (2) induced leakage pathways at certain time. These two pathways are characterized as follows:

(1) Pre-existing leakage pathways: this situation corresponds to a pre-existing leakage pathway that is completely saturated. The instantaneous inflow rate into the leakage pathway is assumed equivalent to the outflow rate from the leakage pathway, corresponding to the fundamental continuity equation.

(2) Induced leakage pathways: this case corresponds to a leakage pathway that is generated at an arbitrary-time injection-induced increased pressure, or other mechanism(s). The model releases leakage from the pathway into an adjacent confined aquifer at the time when the pathway is generated. By assumption (see section 5.2.1), the inflow rate into the leakage pathway is simultaneous with, and

the same, as the outflow rate from the leakage pathway.

5.3 Discretization for Leakage Simulations

In a finite-difference flow simulation, hydraulic heads are calculated at discrete points in space, which in this study are nodes in a mesh. Generally, the FDM consists of either node-centered (mesh-centered) or block-centered schemes. In this study, we use a block-centered scheme so that assigning hydraulic conductivities and specific storage coefficients in the cells is straightforward.

The leakage simulations are carried out by solving for the hydraulic heads at the coupled leakage points in the leakage pathway. The methodology uses three finite difference equations, including two difference equations for a couple of leakage points and one difference equation for no leakage points. This methodology can provide a computational advantage because it directly uses a couple of leakage points on the leakage pathway for leakage simulations. Hydraulic head at one leakage point is solved by hydraulic heads at seven discrete nodes which consist of six discrete adjacent nodes of FDM and one corresponding leakage point. Hydraulic head at the corresponding leakage point is also solved in the same way. This method can simultaneously calculate hydraulic heads at a couple of leakage points and the leakage rates, solved by computing the difference of hydraulic heads between the coupled leakage points. Thus, the methodology does not need meshes to specify leakage pathways, which is more computationally efficient. The mass conservation with a leakage term has been explicitly discretized to three finite difference equations as described in the next section.

5.3.1 Difference Equation of First-Order Derivative

Fig. 5.2 depicts six aquifer cells adjacent to cell (i, j, k), i.e., (i-1, j, k); (i+1, j, k); (i, j-1, k); (i, j+1, k); (i, j, k-1); and (i, j, k+1). Fig. 5.3 depicts a y-dimensional flow of FDM cells for a block of aquifer extending from node (i, j-1, k) to (i, j, k) and from node (i, j, k) to (i, j+1, k) with a cross-sectional area $\Delta x_i \Delta z_k$. Each cell (i, j-1, k), (i, j, k) and (i, j+1, k) is assigned hydraulic conductivity, $Kx_{i,j-1,k}$, $Ky_{i,j-1,k}$, $Kz_{i,j-1,k}$, $Kx_{i,j,k}$, $Ky_{i,j,k}$, $Kz_{i,j,k}$, $Kx_{i,j+1,k}$, $Ky_{i,j+1,k}$, and $Kz_{i,j+1,k}$, respectively.

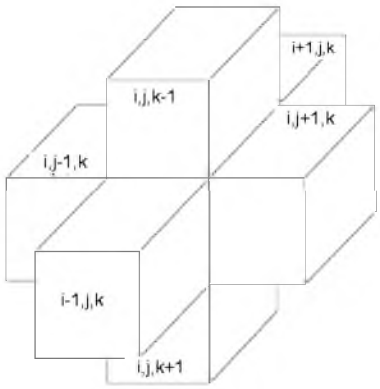


Fig. 5.2 The six adjacent cells surrounding cell (i, j, k) (hidden). Modified from Harbaugh (2005).

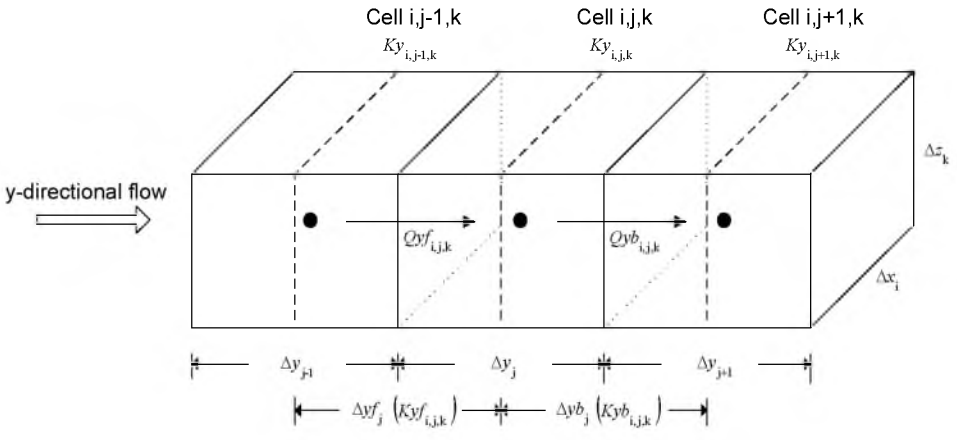


Fig. 5.3 Flow into cell (i, j, k) from cell (i, j-1, k). Modified from Harbaugh (2005).

The effective y-directional hydraulic conductivity of the material between nodes (i, j, k) and (i, j-1, k) is described as $Kyf_{i,j,k}$, and the effective y-directional hydraulic conductivity between nodes (i, j, k) and (i, j+1, k) is denoted as $Kyb_{i,j,k}$. In terms of node (i, j, k), the notation f indicates the region into which water flows from upstream node, and the notation b indicates the region from which water flows to downstream node.

In the same manner, the each effective x-directional and z-directional hydraulic conductivity values are described as $Kxf_{i,j,k}$, $Kxb_{i,j,k}$, $Kzf_{i,j,k}$, and $Kzb_{i,j,k}$. The effective hydraulic conductivity is calculated as a weighted harmonic mean as described by Collins (1961). For example,

$$Kxf_{i,j,k} = \frac{\left(\Delta x_{i-1} + \Delta x_{i}\right)}{\left(\frac{\Delta x_{i-1}}{kx_{i-1,j,k}} + \frac{\Delta x_{i}}{kx_{i,j,k}}\right)} \text{ and } Kxb_{i,j,k} = \frac{\left(\Delta x_{i} + \Delta x_{i+1}\right)}{\left(\frac{\Delta x_{i}}{kx_{i,j,k}} + \frac{\Delta x_{i+1}}{kx_{i+1,j,k}}\right)}, \tag{5.4}$$

and $Kyf_{i,j,k}$, $Kyb_{i,j,k}$, $Kzf_{i,j,k}$, and $Kzb_{i,j,k}$ can be calculated in the same way. In Fig. 5.3, $Qyf_{i,j,k}$ represents the volumetric flux through the face between cells (i, j, k) and (i, j-1, k), and $Qyb_{i,j,k}$ represents the volumetric flux through the face between cells (i, j, k) and (i, j+1, k). The term Δyf_{j} is the distance between nodes (i, j, k) and (i, j-1, k) and Δyb_{j} is the distance between nodes (i, j, k) and (i, j+1, k).

Fig. 5.4 illustrates x-directional cross sections through three cells and the numerical approximation of derivatives of hydraulic head under anisotropic and heterogeneous conditions (Bennett 1976).

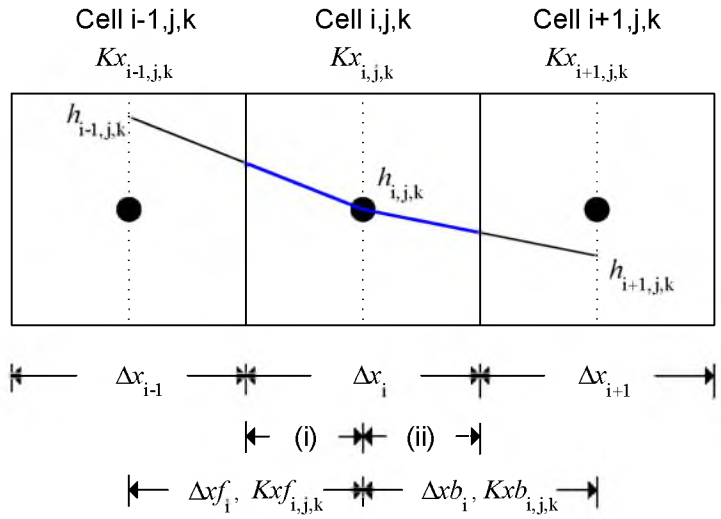


Fig. 5.4 Schematic x-directional cross section. Modified from Bennett (1976).

The difference equations of sections (i) and (ii) in Fig. 5.4 can be approximated by

$$\left(Kx\frac{\Delta h}{\Delta x}\right)_{(i)} \approx Kxf_{i,j,k}\frac{(h_{i-1,j,k}-h_{i,j,k})}{\Delta xf_i}, \text{and} \left(Kx\frac{\Delta h}{\Delta x}\right)_{(ii)} \approx Kxb_{i,j,k}\frac{(h_{i,j,k}-h_{i+1,j,k})}{\Delta xb_i}. \quad (5.5)$$

Thus, the x-directional first-order partial derivative can be approximated by the arithmetic mean of the fluxes of section (i) and (ii),

$$\left(Kx\frac{\partial h}{\partial x}\right)_{i,j,k} \approx \frac{1}{2}\left(Kxf_{i,j,k}\frac{(h_{i-1,j,k}-h_{i,j,k})}{\Delta xf_i} + Kxb_{i,j,k}\frac{(h_{i,j,k}-h_{i+1,j,k})}{\Delta xb_i}\right). \quad (5.6)$$

In the same way, we can obtain the difference equations of $\left(Ky\frac{\partial h}{\partial y}\right)_{i,j,k}$ and $\left(Kz\frac{\partial h}{\partial z}\right)_{i,j,k}$.

These difference formulas of first-order partial derivatives have a local truncation error of order $O(\Delta x^2)$, $O(\Delta y^2)$, and $O(\Delta z^2)$.

The time derivative of the heads can be approximated with the backward difference method:

$$\left(\frac{\partial h}{\partial t}\right)_{t^n} \approx \left(\frac{(h^n_{i,j,k} - h^{n-1}_{i,j,k})}{t^n - t^{n-1}}\right), \tag{5.7}$$

where n is the time step index. This difference formula has a local truncation of order O(Δt). If Equation (5.7) is zero, the heads are or have reached steady-state.

5.3.2 Difference Equation of Second-Order Partial Derivative

Similarly, a second-order accurate approximation for the x-directional second order partial derivative of head at cell i, j, k can be given as

$$\left(\frac{\partial}{\partial x}(Kx\frac{\partial h}{\partial x})\right)_{i,j,k} \approx \frac{\left(Kx\frac{\Delta h}{\Delta x}\right)_{(i)} - \left(Kx\frac{\Delta h}{\Delta x}\right)_{(ii)}}{\Delta x_i} \approx$$

$$\frac{1}{\Delta x_i}\left(Kxf_{i,j,k}\frac{(h_{i-1,j,k} - h_{i,j,k})}{\Delta xf_i} + Kxb_{i,j,k}\frac{(h_{i+1,j,k} - h_{i,j,k})}{\Delta xb_i}\right). \tag{5.8}$$

In the same way, the difference equations of $\left(\frac{\partial}{\partial y}(Ky\frac{\partial h}{\partial y})\right)_{i,j,k}$ and $\left(\frac{\partial}{\partial z}(Kz\frac{\partial h}{\partial z})\right)_{i,j,k}$ can be obtained. These difference equations have a local truncation error of order O(Δx^2), O(Δy^2), and O(Δz^2).

5.3.3 Difference Equations with a Leakage Term

The governing equation with the leakage term consists of one of three finite difference equations, the form of which depends on whether the cells reflect leakage pathways at the time step in question.

The first equation is applied to leakage node (i, j, k) in the storage aquifer of the I-

th leakage pathway. If the leakage node (i, j, k) is assigned as an induced leakage pathway, the following equation is applied to node (i, j, k) after the time when the leakage is induced at node (i, j, k):

$$h^n_{i,j,k} = \frac{1}{\mu 2^n_{i,j,k}}\left[\left(Cxf_{i,j,k}h^n_{i-1,j,k} + Cxb_{i,j,k}h^n_{i+1,j,k}\right) + \left(Cyf_{i,j,k}h^n_{i,j-1,k} + Cyb_{i,j,k}h^n_{i,j+1,k}\right) + \left(Czf_{i,j,k}h^n_{i,j,k-1} + Czb_{i,j,k}h^n_{i,j,k+1}\right) + W^n_{i,j,k} + \frac{Ss_{i,j,k}h^{(n-1)}_{i,j,k}}{t^n - t^{n-1}} + Czbleak_{i,j,k}h^n_{i,j,zlsu(I)}\right], \quad (5.9)$$

where $\mu 2^n_{i,j,k} = Cxf_{i,j,k} + Cxb_{i,j,k} + Cyf_{i,j,k} + Cyb_{i,j,k} + Czf_{i,j,k} + Czb_{i,j,k} + \frac{Ss_{i,j,k}}{t^n - t^{n-1}} + Czbleak_{i,j,k}$,

$$Czbleak_{i,j,k} = Kzbleak_{i,j,k}\frac{Aleak_{i,j,k}}{Dzbleak(I)\cdot(\Delta x_i \cdot \Delta y_j \cdot \Delta z_k)}, \quad Cxf_{i,j,k} = Kxf_{i,j,k}\frac{1}{\Delta x_i \Delta xf_i},$$

$$Cxb_{i,j,k} = Kxb_{i,j,k}\frac{1}{\Delta x_i \Delta xb_i}, \quad Cyf_{i,j,k} = Kyf_{i,j,k}\frac{1}{\Delta y_j \Delta yf_j}, \quad Cyb_{i,j,k} = Kyb_{i,j,k}\frac{1}{\Delta y_j \Delta yb_j},$$

$$Czf_{i,j,k} = Kzf_{i,j,k}\frac{1}{\Delta z_k \Delta zf_k}, \quad Czb_{i,j,k} = Kzb_{i,j,k}\frac{1}{\Delta z_k \Delta zb_k}.$$

The second equation is applied to the leakage node (i, j, k) in the overlying aquifer of the I-th leakage pathway. If the node (i, j, k) becomes a point on the induced leakage pathway at the same specific time during the simulation, this equation is applied to that node after the time that marks the start of leakage:

$$h^n_{i,j,k} = \frac{1}{\mu 2^n_{i,j,k}}\left[\left(Cxf_{i,j,k}h^n_{i-1,j,k} + Cxb_{i,j,k}h^n_{i+1,j,k}\right) + \left(Cyf_{i,j,k}h^n_{i,j-1,k} + Cyb_{i,j,k}h^n_{i,j+1,k}\right) + \left(Czf_{i,j,k}h^n_{i,j,k-1} + Czb_{i,j,k}h^n_{i,j,k+1}\right) + W^n_{i,j,k} + \frac{Ss_{i,j,k}h^{(n-1)}_{i,j,k}}{t^n - t^{n-1}} + Czbleak_{i,j,k}h^n_{i,j,zls(I)}\right]. \quad (5.10)$$

In the third equation, if the node (i, j, k) is a normal cell without leakage at the

leakage aquifers, or a leakage cell has or is in no leakage condition before leakage is induced at the certain time, the following difference equation is applied to node (i, j, k):

$$h^n_{i,j,k} = \frac{1}{\mu l^n_{i,j,k}}\left[\left(Cxf_{i,j,k}h^n_{i-1,j,k} + Cxb_{i,j,k}h^n_{i+1,j,k}\right) + \left(Cyf_{i,j,k}h^n_{i,j-1,k} + Cyb_{i,j,k}h^n_{i,j+1,k}\right) + \left(Czf_{i,j,k}h^n_{i,j,k-1} + Czb_{i,j,k}h^n_{i,j,k+1}\right) + W^n_{i,j,k} + \frac{Ss_{i,j,k}h^{(n-1)}_{i,j,k}}{t^n - t^{n-1}}\right], \quad (5.11)$$

where $\mu l^n_{i,j,k} = Cxf_{i,j,k} + Cxb_{i,j,k} + Cyf_{i,j,k} + Cyb_{i,j,k} + Czf_{i,j,k} + Czb_{i,j,k} + \frac{Ss_{i,j,k}}{t^n - t^{n-1}}$.

5.3.4 Initial and Boundary Conditions

The hydraulic head distribution of transient flow is calculated at node (i, j, k) by solving the difference equations, which requires initial and boundary conditions. For these models, a boundary condition on time step zero was used to evaluate steady-state flow, and the steady-state condition was then assigned as the initial condition for subsequent transient flow. The boundary conditions generally consist of two categories, either constant-head (Dirichlet boundary) or no-flow boundary (Neumann boundary). For the Dirichlet boundary condition, the developed model can specify time-varying heads for specific time periods along a boundary. In transient flow simulations, the time-dependent boundary heads must correspond in terms of simulation time.

Fig. 5.5 denotes procedure of calculation of transient flow. The simulator developed for this study utilized a Gauss-Seidel iterative scheme for solving linear systems. Convergence criterion of the iterative method required that the difference, at all nodes, between the new approximation and previous approximation is less than a specified tolerance.

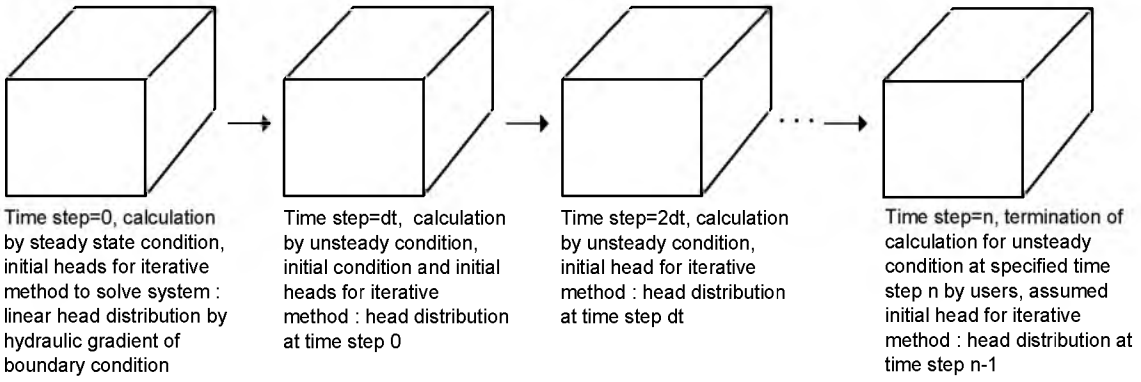


Fig. 5.5 Schematic description of iterative calculation of a head distribution.

Once a new approximation within that tolerance is determined, then the calculation for the next time step has begun. One of the two boundary conditions should be used for every cell in the outside of six faces of the grid.

5.4 Forward Analysis

This section discusses validation of the developed forward model, and provides a simulation example of an aquifer with leakage.

5.4.1 Validation of Forward Model

The TOUGH2 program coupled with a general equation of state (EOS1) provides a simulation of pure water in its liquid, vapor and/or two-phase states (Pruess et al., 1999), and this code was used to validate a developed forward model for single-phase fluid.

The EOS1 module calculates all water properties (density, viscosity, specific enthalpy, etc.) from the steam table equations in nonisothermal and isothermal conditions. Validation of the developed forward model was investigated with a two-dimensional example with and without water leaks (isothermal only).

5.4.1.1 Example without Leakage

A FDM model of this conceptual model was developed. Simulations were conducted using the new simulator developed for this study, and using TOUGH2. Results of the two simulations were compared. Fig. 5.6 shows a schematic of multiple aquifers without a leakage pathway. In Fig. 5.6, the 1st and 2nd layers represent an overlying aquifer, the 3rd to 5th layers represent a cap rock, and the 6th to 13th layers represent a storage aquifer. The two- dimensional domain is assigned a no-flow boundary at both the top and bottom layers, and a 20 m constant head is assigned to the left boundary and an 18 m constant head along the right boundary of the domain. Table 5.1 summarizes the specifications of the numerical model.

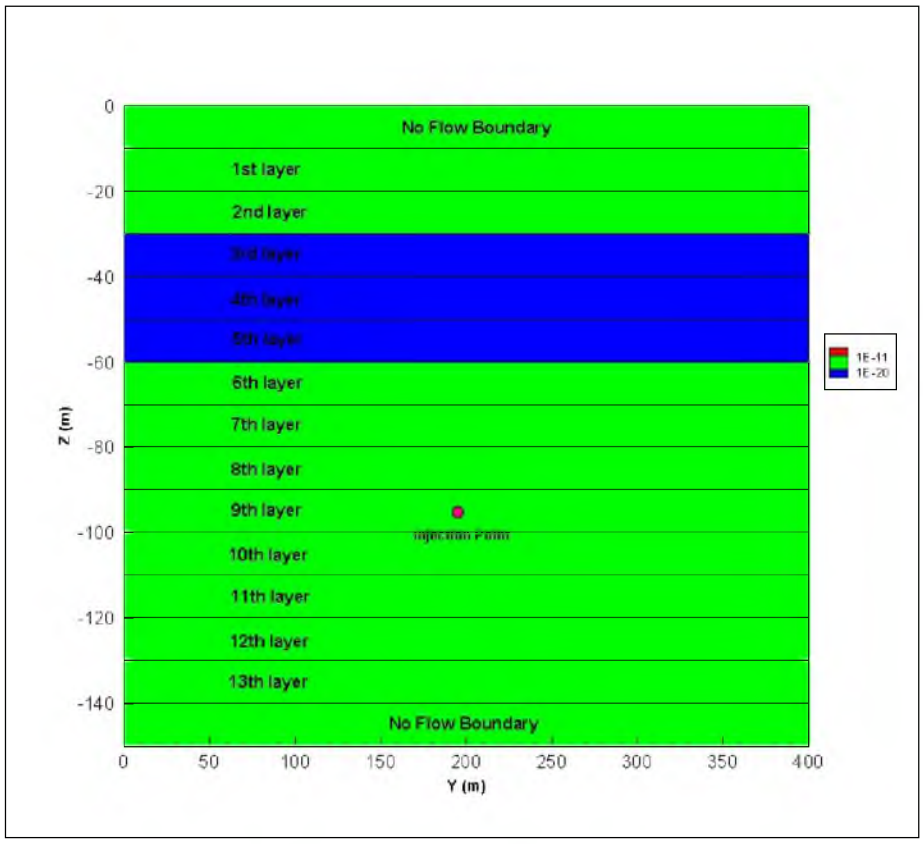


Fig. 5.6 Schematic of a two-dimensional domain without a leakage pathway.

Table 5.1 Specifications of the conceptual model.

Domain size (m)	400×150	**Time step size (sec)**		100,000
Each cell size (m)	10×10	**Tolerance**		1e-9
Number of cells	40×15 (600 total)	**Permeability (m^2)**	**Overlying and storage aquifers**	$k_y = k_z$ = 1.0e-11
Simulation time (sec)	0 – 20,000,000 (231.48 days)		**Cap rock**	$k_y = k_z$ = 1.0e-20

The FDM model uses a constant hydraulic conductivity ($K = k\rho g/\mu$ where K: hydraulic conductivity, k: permeability, ρ: density, g: gravity acceleration and μ: water dynamic viscosity) for each discrete cell, whereas, the TOUGH2 coupled with EOS1 module uses a constant permeability. Water density and viscosity are calculated from the steam table equations for each cell. Therefore, the average density and viscosity in the model domain calculated from the EOS1 are assigned to the developed single-phase model to calculate the hydraulic conductivity. In addition, the FDM model uses volumetric injection rates but TOUGH2 uses mass injection rates. The mass injection rates are converted to equal volume injection rates. Specific storage of the aquifers is calculated using:

$$S_s = \emptyset\rho g(\beta_w + \beta_p), \tag{5.12}$$

where, ∅: porosity, ρ water density, g: gravity acceleration, β_w: water compressibility, β_p: aquifer pore compressibility.

Table 5.2 denotes the average water density and viscosity, the assigned hydraulic conductivity values and other properties in the model domain. Table 5.3 summarizes the water mass and volumetric injection rates. Fig. 5.7 presents simulation results in the form of pressure distributions at nodes in the 9^{th} layer from both the TOUGH2 and the new

Table 5.2 Water properties and domain properties of the FDM model.

Density (kg/m^3)	1000	**Hydraulic conductivity (m/s)**	**Overlying and storage aquifers**	$K_y = K_z =$ 7.544e-05
Viscosity (kg/m·s)	0.0013		**Cap rock**	$K_y = K_z =$ 7.544e-14
Water compressibility (Pa^{-1})	3.5×10^{-10}	**Porosity**	**Both aquifers and Cap rock**	0.03
Both aquifers and cap rock pore compressibility (Pa^{-1})	4.5×10^{-10}	**Specific storage (m^{-1})**	**Both aquifers and Cap rock**	2.3×10^{-7}
Gravity acceleration (m/s^2)	9.80665			

Table 5.3 FDM model injection rates.

Cell # (Location from origin)	**Water Injection Rates**		
	Injection time (sec)	**Mass rate (kg/s)**	**Volumetric rate (m^3/s)**
(20, 10) (195 m, -95 m)	0	0.0	0.0
	20,000,000	1.0	0.001

FDM model. In Fig. 5.7, solid lines indicate pressures from the TOUGH2 and dashed lines represent pressures from the new FDM model. Fig. 5.8 describes relative errors in the 9th layer between both models. The maximum error is approximately 0.00006 (0.006%) at the injection node (195 m, -95 m from the origin). Fig. 5.9 illustrates pressure distributions at nodes in the 2nd layer from both models. Fig. 5.10 shows relative errors in the 2nd layer. The maximum error is approximately 0.0000085 (0.00085%).

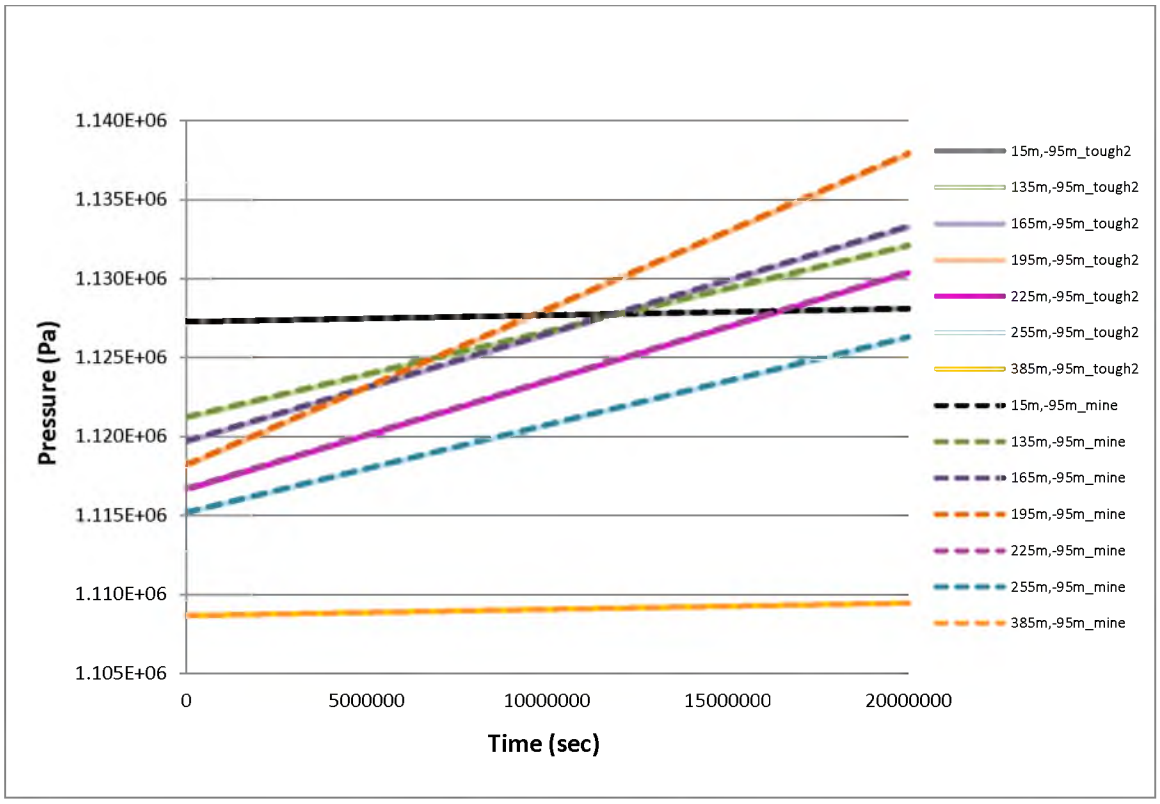


Fig. 5.7 Simulated pressure distributions in the 9th layer from both models.

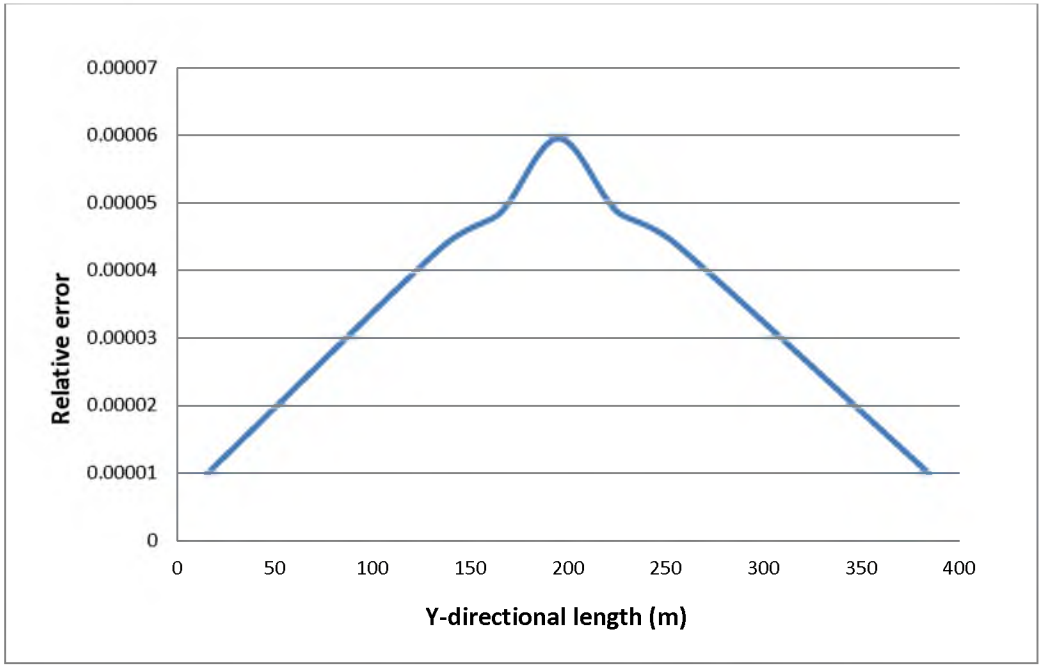


Fig. 5.8 Relative errors in the 9th layer between both models.

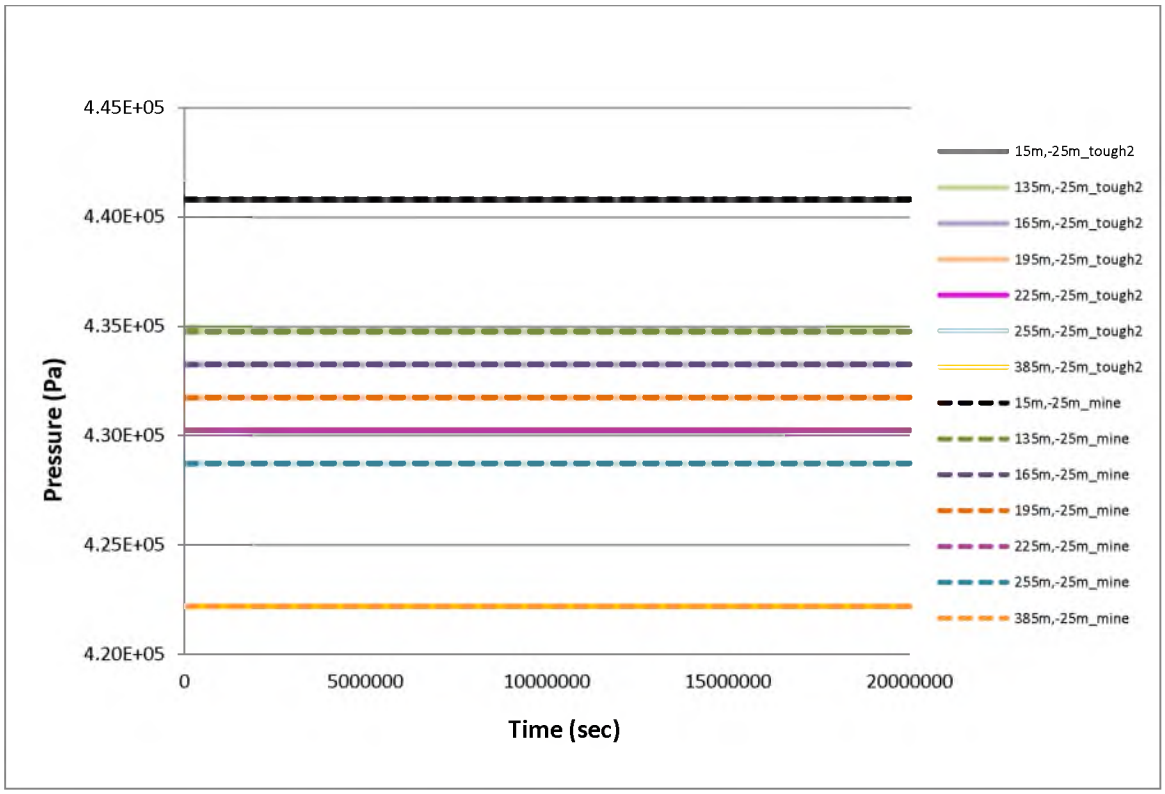


Fig. 5.9 Simulated pressure distributions in the 2nd layer from both models.

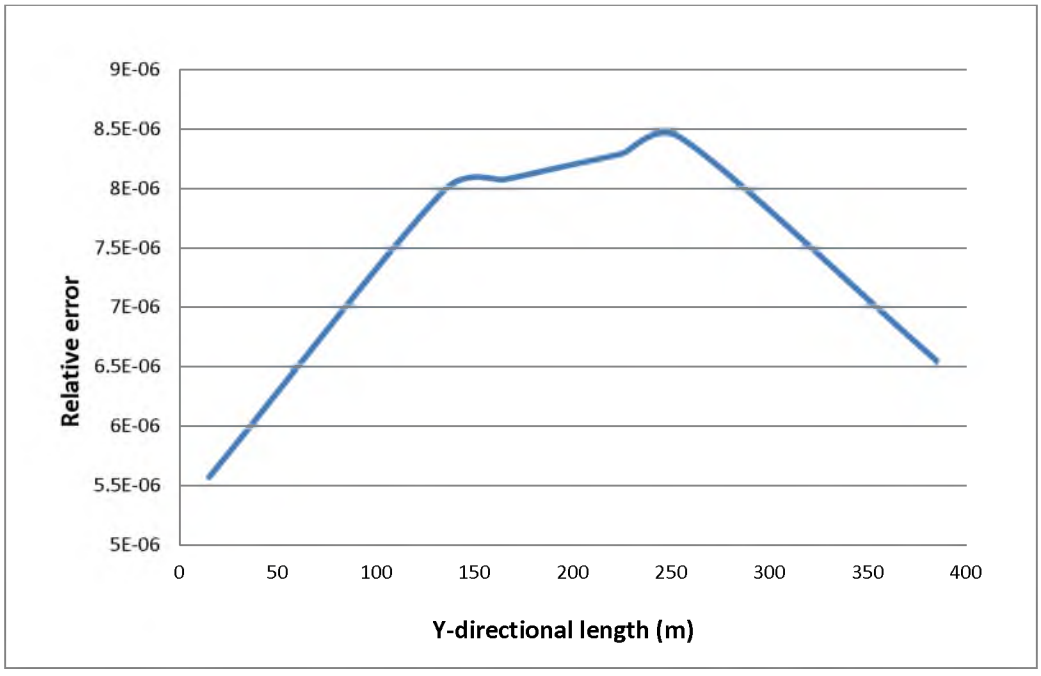


Fig. 5.10 Relative errors in the 2nd layer between both models.

5.4.1.2 Example with Leakage

For leakage simulations, the geometry of a leakage pathway was meshed in the seal layers of the original model domain (see Fig. 5.6 and Fig. 5.11). The leakage simulations using TOUGH2 are conducted by calculating pressures at the inner nodes of an explicitly-meshed leakage pathway. Nordbotten et al. (2004) called this kind of leakage simulation "a conventional method." As mentioned before, however, the new FDM model uses coupled leak points in both aquifers, without use of explicit (tailored/special) mesh leakage pathways. The leakage simulation conditions are the same as those used for no-leakage scenario simulations, except for properties of the leakage pathway.

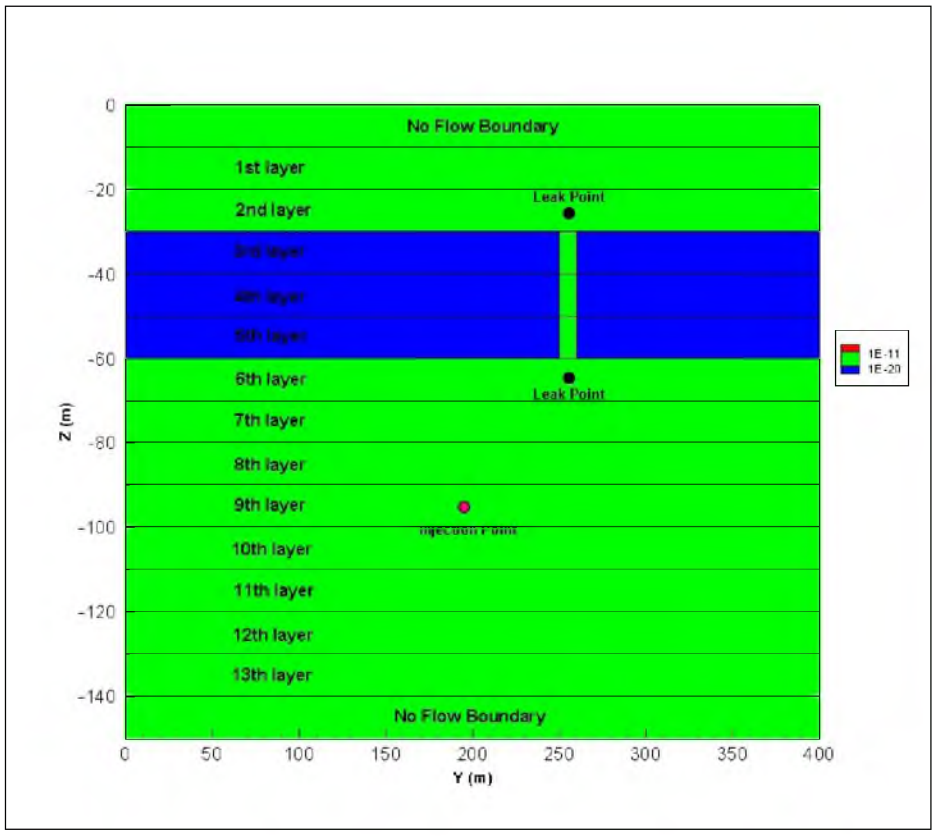


Fig. 5.11 Schematic of a two-dimensional domain with a leakage pathway.

Table 5.4 summarizes the assigned properties of the leakage pathway in the model domain. Fig. 5.12 illustrates simulated pressure distributions in the 9th layer for both the TOUGH2 model and the new FDM model. Fig. 5.13 describes relative errors in the 9th layer between both models. The maximum error is approximately 0.00005 (0.005%) at the injection node. Fig. 5.14 represents simulated pressure distributions in the 2nd layer from both models.

Table 5.4 Assigned leakage pathway properties.

Leakage pathway			
Cell # and Location from origin	**Size (m)**	**Permeability (m^2)**	**Hydraulic conductivity (m/s)**
(26,4) ~ (26,6) (255 m, -30 m) ~ (255 m, -60 m)	10 × 30	k_z =1.0e-11	K_z =7.544e-05

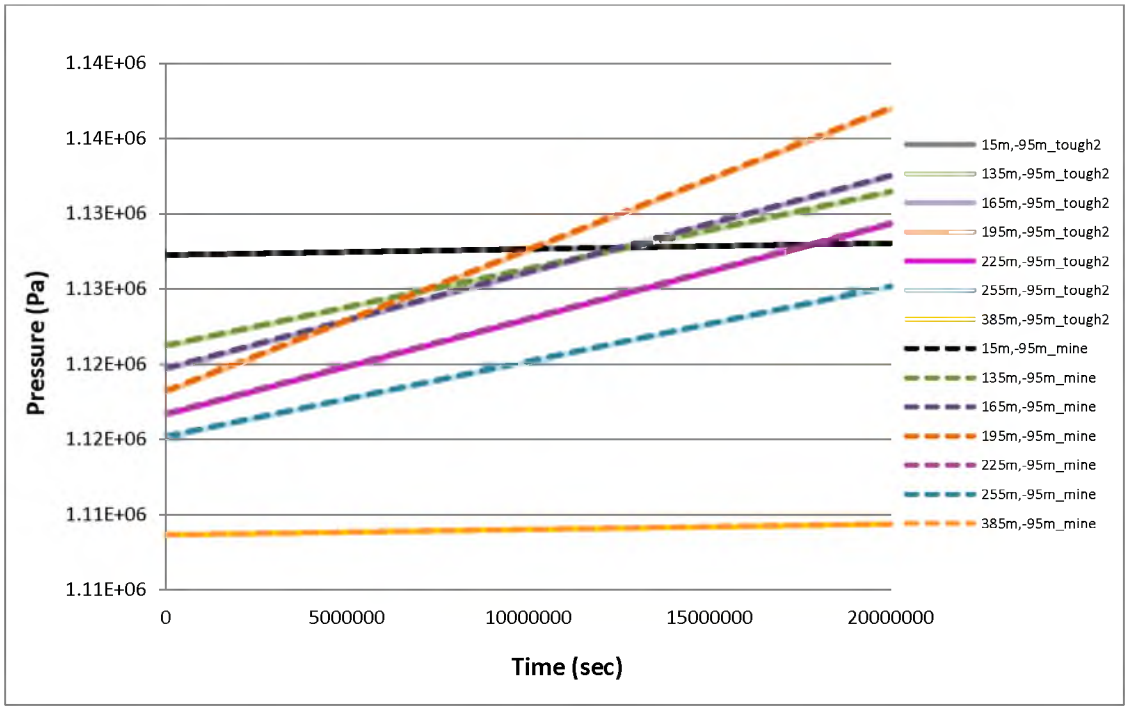


Fig. 5.12 Simulated pressure distributions in the 9th layer from both models. The legend indicates which trends correspond to TOUGH2 simulations and which trends correspond to the new FDM model simulation ("mine").

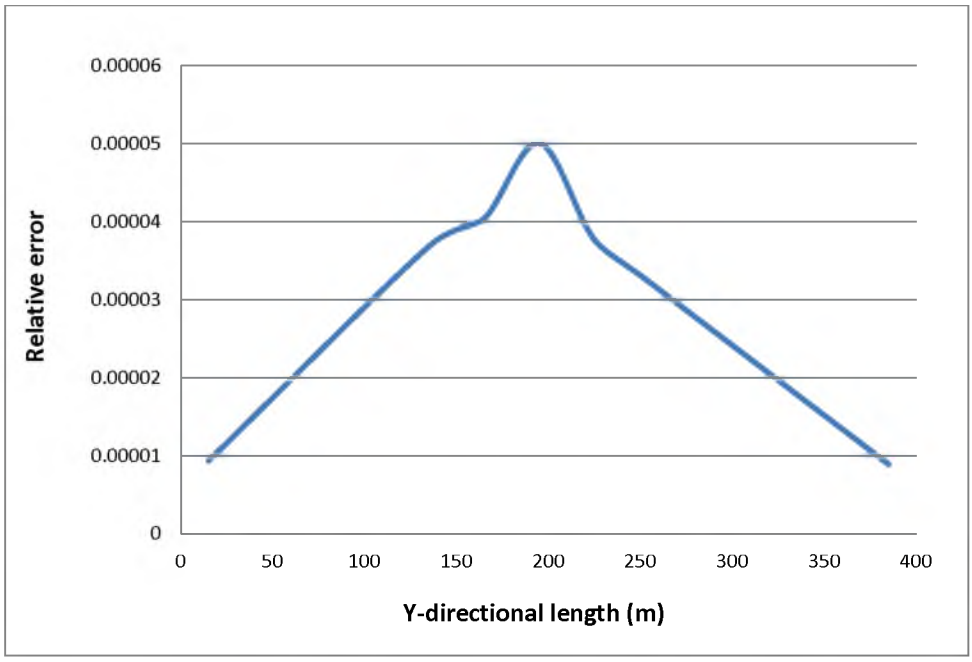


Fig. 5.13 Relative errors in the 9th layer between both models (TOUGH2 vs. new model).

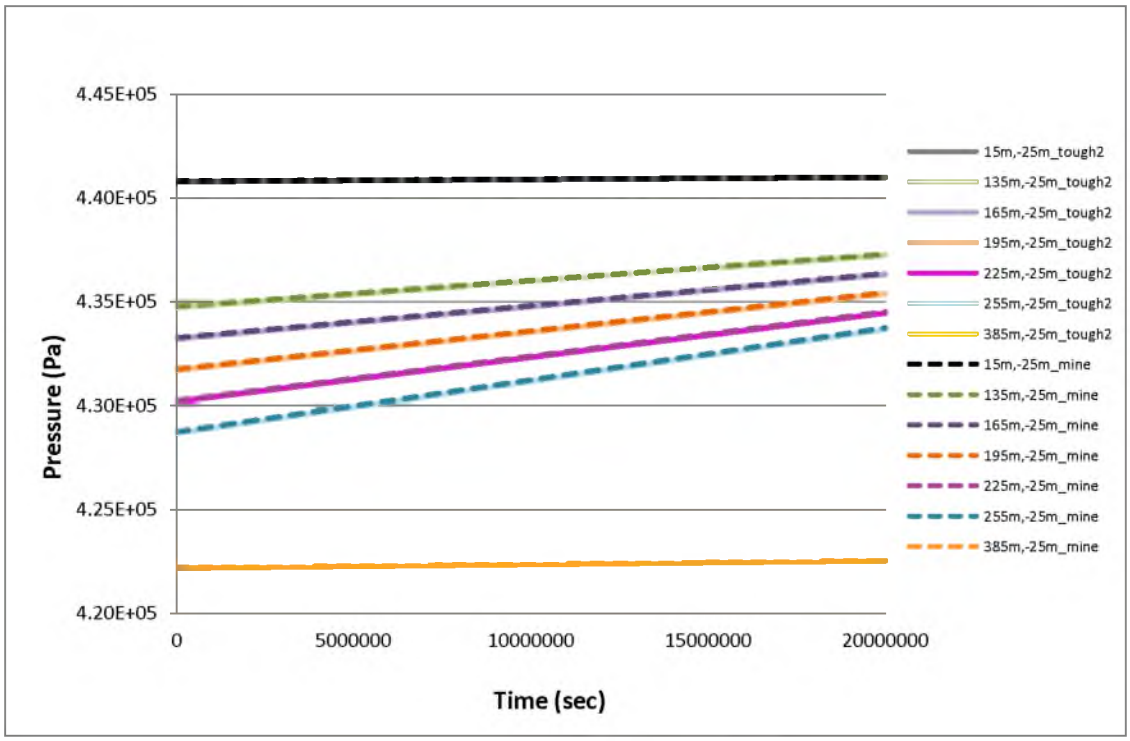


Fig. 5.14 Simulated pressure distributions in the 2nd layer from both models. The legend indicates which trends correspond to TOUGH2 simulations and which trends correspond to the new FDM model simulation ("mine").

Fig. 5.15 illustrates relative errors in the 2nd layer. The maximum error is approximately 0.000135 (0.0135%) at the leak point in the overlying formation. Fig. 5.16 shows simulated pressure distributions at two leak points and the injection node from both models. In Fig. 5.16 (a), a relative error at the leak point in the storage formation is approximately 0.000108 (0.0108%).

The new FDM model was tested by comparing to corresponding results from the TOUGH2 simulations. Recall that the new FDM model uses constant hydraulic conductivities characterized by the average density and viscosity calculated from TOUGH2 as shown in Table 5.2. Incorrect density and viscosity might cause under- or overestimated hydraulic conductivity values, inducing errors in simulation outputs of the new FDM model.

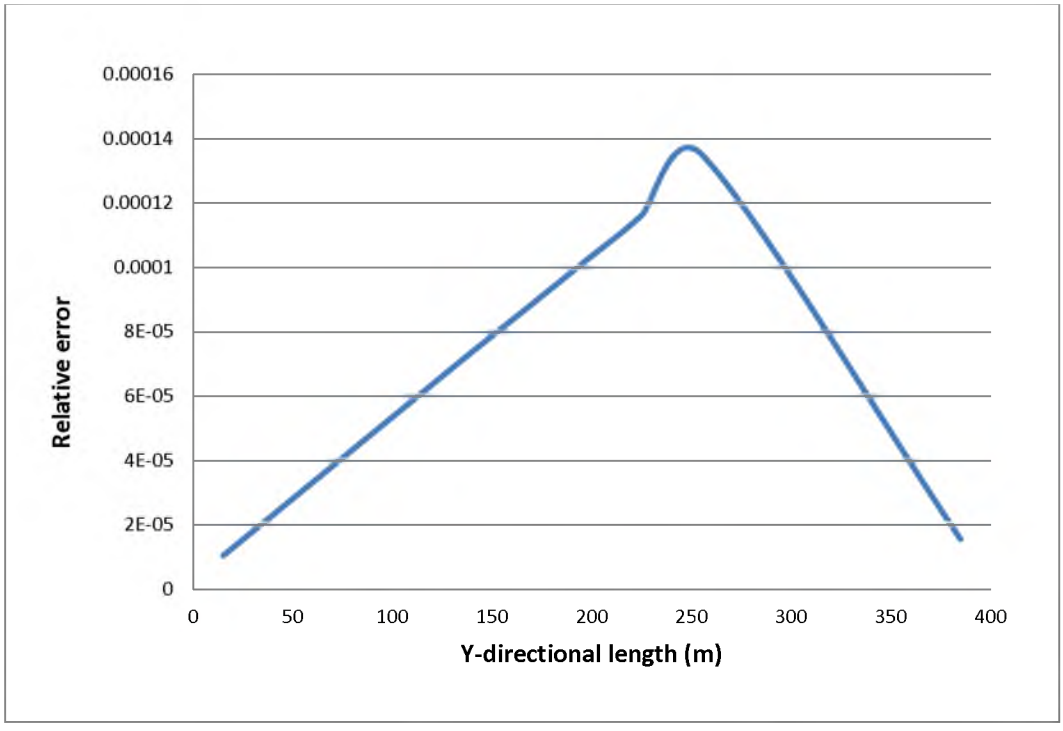


Fig. 5.15 Relative errors in the 2nd layer between both models (TOUGH2 vs. new model).

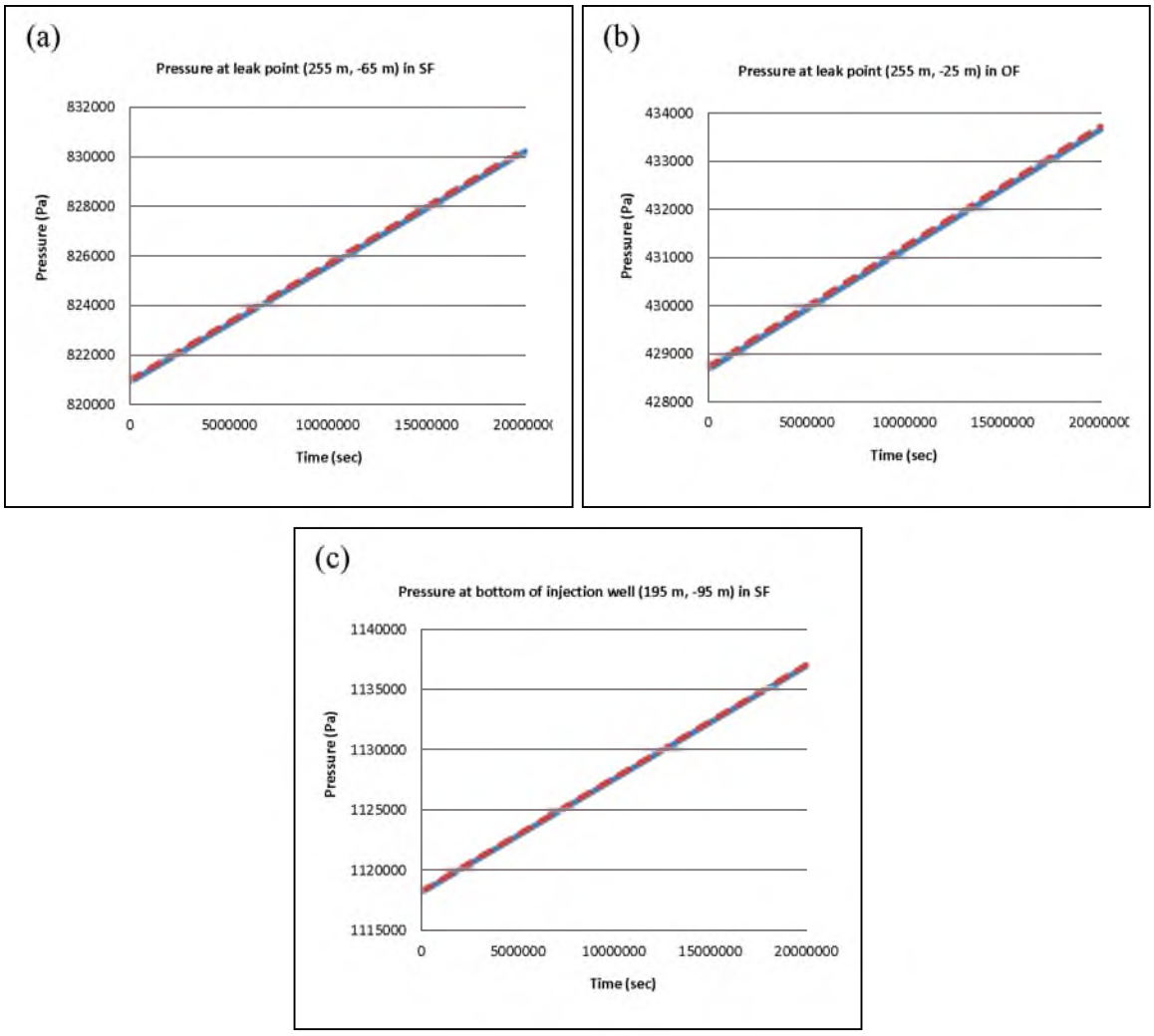


Fig. 5.16 Simulated pressure distributions at two leak points and the injection well (Solid blue line: TOUGH2 and dashed red line: new FDM model): (a) at the leak point in the storage formation (b) at the leak point in the overlying formation and (c) at the injection node.

Fig. 5.17 illustrates increased pressure deviations at two leak points and the injection well for both simulators; note that the used viscosity value in the new FDM model is underestimated to 0.0010 kg/m·s, compared to the TOUGH2-calculated value of 0.0013 kg/m·s, or 30 % relative error. The hydraulic conductivity values in each cell are correspondingly increased from 7.544e-5 m/s to 7.708e-5 m/s in the overlying and storage aquifers, respectively.

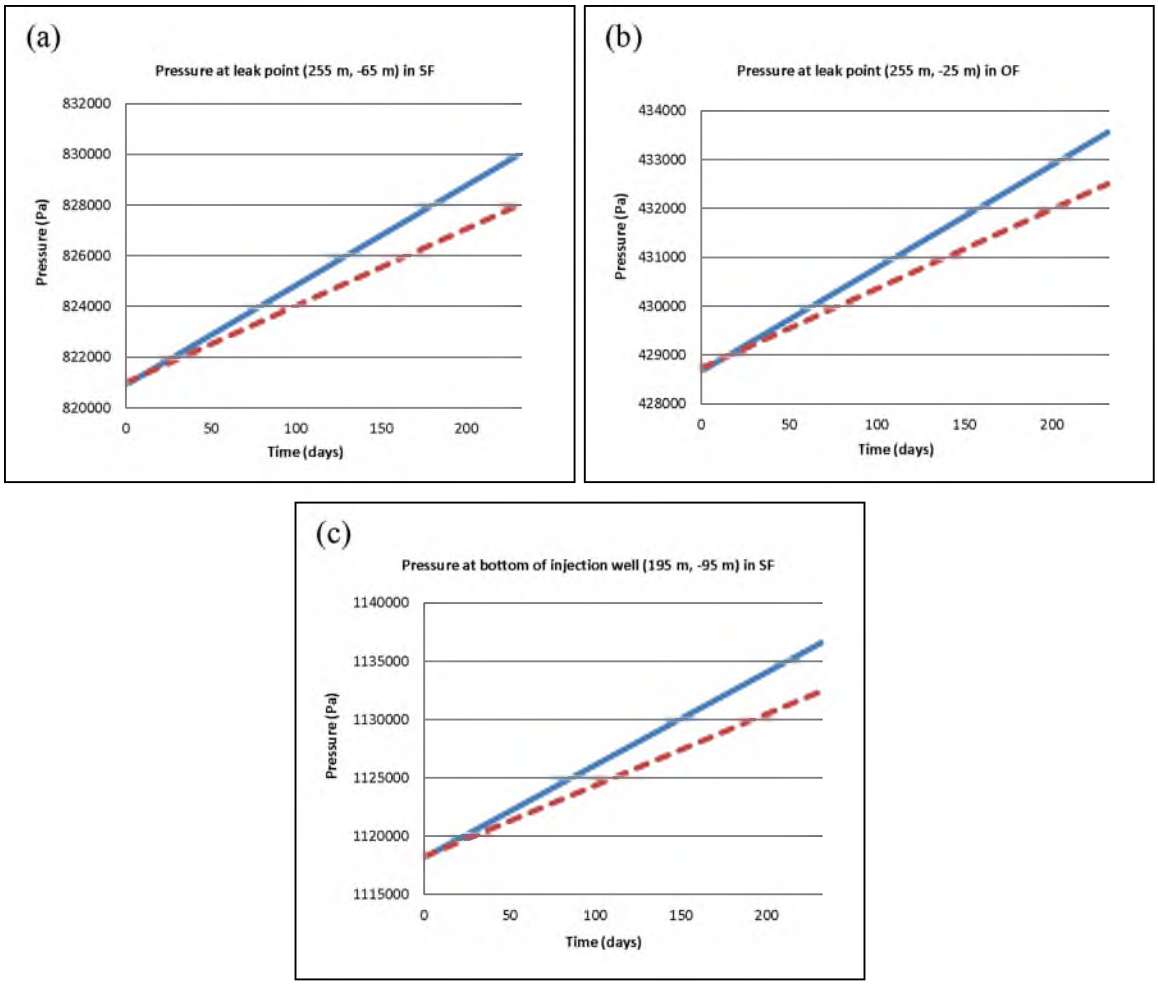


Fig. 5.17 Pressure deviations from 0.0010 kg/m·s viscosity (Solid blue line: TOUGH2 and dashed red line: new FDM model): (a) at the leak point in the storage formation (b) at the leak point in the overlying formation and (c) at the injection node.

The relative errors in pressure illustrated by Fig. 5.17 (a), (b) and (c) are 0.0024 (0.24 %), 0.0025 (0.25 %), and 0.0037 (0.37 %) at the last simulation time step, respectively. As mentioned earlier, an ultimate objective of this study is to identify applicability of the new FDM model to the overlying aquifer of brine/CO_2 systems for leakage pathway estimation. The results of Fig. 5.17 indicate that hydraulic conductivity of the overlying formation, as assigned in the new FDM model, should probably be estimated to reduce errors in calculated results.

5.4.2 Leakage Forward Simulation

The purpose of this section is to demonstrate how the developed forward model simulates hydraulic head anomalies from two kinds of leakage pathways in terms of the leakage generated (start) time; (1) pre-existing leakage pathways and (2) induced leakage pathways at specified starting-times in the aquifer with model domain. Fig. 5.18 is a schematic diagram of multiple aquifers with leakage. As expressed by Fig. 5.18, single-phase water injection is assigned to realize the transient release of leakage. The z-directional-10th layer, between both aquifers, is a confining layer. It is assumed that the leakage occurs at two pathways: one pathway between node (51, 55, 9) and node (51, 55, 11) and another pathway between node (56, 60, 9) and node (56, 60, 11) with a time interval.

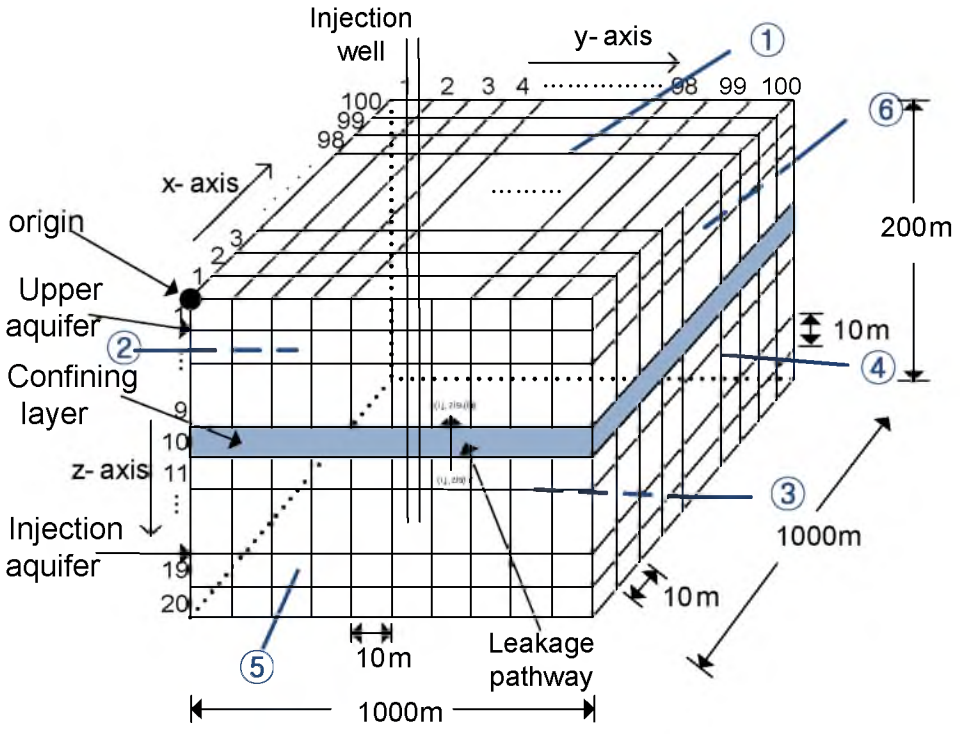


Fig. 5.18 Multiple aquifers with leakage along a leakage pathway (the circled numbers indicate facies).

The first pathway has leakage at time zero (this is a pre-existing pathway) and the second pathway begins leakage at time 2,000,000 sec. The second pathway is representative of leakage induced at an arbitrary time (in this case, 2×10^6 seconds after the simulation starts) by an external effect or mechanism, such as a microseismic events due to overpressure from water injection, etc. Therefore, three cases of simulations were performed: (1) no leakage, (2) one leakage at the first pathway, and (3) two leakages at the first and second pathways. Table 5.5 summarizes the general specifications of the conceptual model. Specific storage was calculated using Equation (5.12). Table 5.6 describes the assigned water injection. Table 5.7 summarizes the specific simulation conditions for leakage. Table 5.8 details the boundary condition.

Table 5.5 General specifications of the model.

Cubic size (m)	1,000×1,000×200	**Tolerance**		1.0×10^{-5}
Each cell size (m)	10×10×10	**Specific storage (m^{-1})**		2.3×10^{-7}
Simulation time (sec)	0 – 10,000,000	**Hydraulic conductivity (m/s)**	**Both aquifers**	$Kx_{i,j,k} = Ky_{i,j,k} = Kz_{i,j,k} =$ 0.0001
Time step size (sec)	100,000		**Confining layer**	$Kx_{i,j,k} = Ky_{i,j,k} = Kz_{i,j,k} =$ 0.0
Water compressibility (Pa^{-1})	3.5×10^{-10}	**Porosity**	**Both aquifers**	0.03
Pore compressibility (Pa^{-1})	4.5×10^{-10}	**Gravity acceleration (m/s^2)**		9.80665

Table 5.6 Assigned injection time and rate.

Cell #	**Injection Conditions**	
	Injection time (sec)	**Injected water** (m^3/s)
(50, 50, 14)	0	0
	100,000	0.02
	10,000,000	0.02

Table 5.7 Leakage specification of the model.

First pathway (between node (51, 55, 9) and (51, 55, 11))			Second pathway (between node (56, 60, 9) and (56, 60, 11))		
Leakage starting time (sec)	**Hydraulic conductivity (m/s)**	**Leakage area (m^2)**	**Leakage starting time (sec)**	**Hydraulic conductivity (m/s)**	**Leakage area (m^2)**
0	0.1	1.0	2,000,000	0.1	1.0

Table 5.8 Boundary conditions for leakage simulations.

Face	**Constant head boundary**	
	Simulation time (sec)	**Boundary head (m)**
Face 2 of upper aquifer	0 - 10,000,000	20
Face 2 of injection aquifer	0 - 10,000,000	25
Face 4 of upper aquifer	0 - 10,000,000	15
Face 4 of injection aquifer	0 - 10,000,000	20
No flow boundary		
Face	1, 3, 5, 6	

5.4.2.1 Simulation Results

Fig. 5.19 illustrates the simulated hydraulic head distribution at the designated leakage pathways in the model domain. In Fig. 5.19, "no_leak" indicates the first case with no leakage pathway, "one_leak" represents the second case with one leakage at the first pathway, and "two_leak" indicates the third leakage condition with two pathways. In the second and third simulations, the first leakage node (51, 55, 9) and (51, 55, 11) do not reflect rapid changes of hydraulic heads because this leakage pathway is pre-existing so it already has an opened pathway. However, in the third simulation, the second leakage node (56, 60, 9) and (56, 60, 11) shows a sharp rise and fall of hydraulic heads immediately after the leakage pathway is generated, and then the hydraulic heads from the second leakage becomes stable.

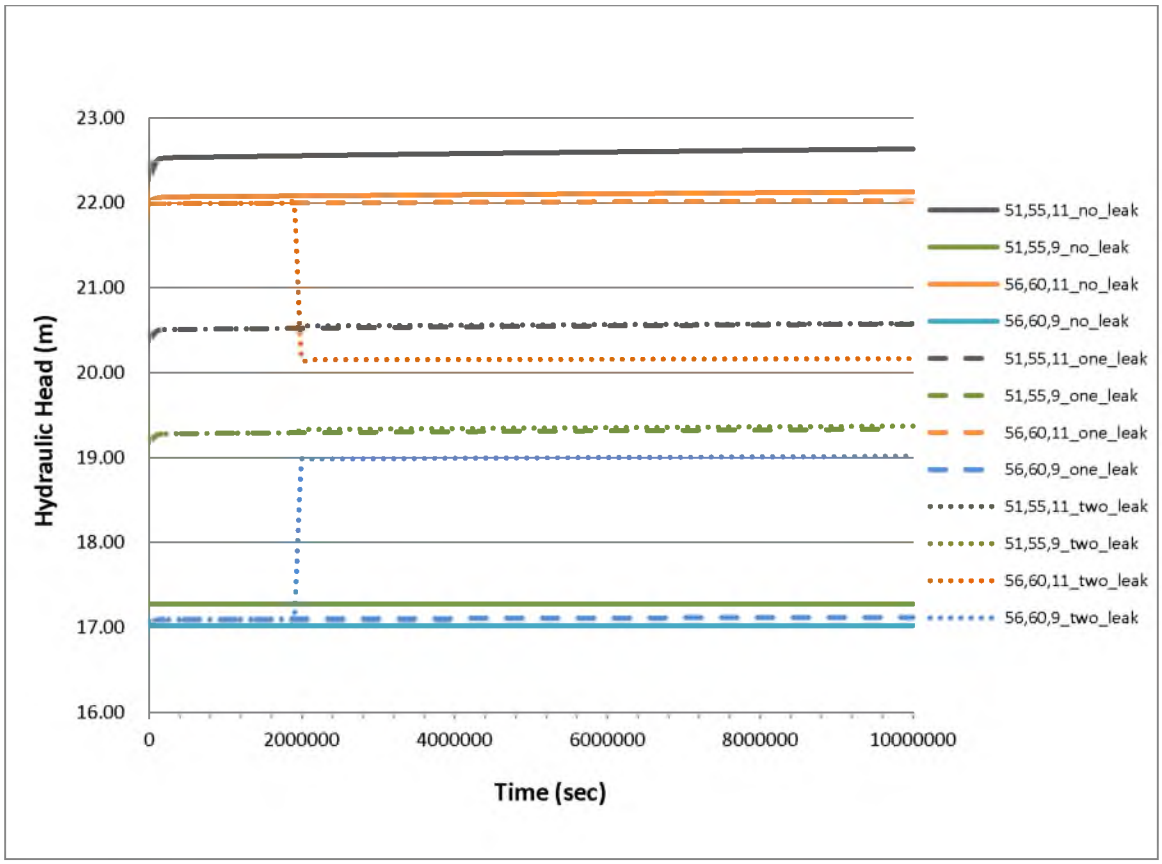


Fig. 5.19 Simulated hydraulic head distribution at leakage pathways of each simulation.

Fig. 5.20 illustrates the hydraulic head distribution at node (50, 60, 15), located in the storage aquifer below the leakage pathway. In the first simulation, the hydraulic head at node (50, 60, 15) rapidly increases as a direct result of water injected into node (50, 50, 14) (a solid line). In the second simulation, hydraulic head due to the pre-existing leakage at node (51, 55, 11) does not significantly increase. After 3,000,000 seconds the pressure does increase due to the elevated pressure from water injection (dashed line, Fig. 5.20). The result from the third simulation is similar to the second case, but hydraulic head does not significantly increase after 3,000,000 seconds because of the effect of induced leakage at node (56, 60, 11) at 2,000,000 seconds (dotted line, Fig. 5.20).

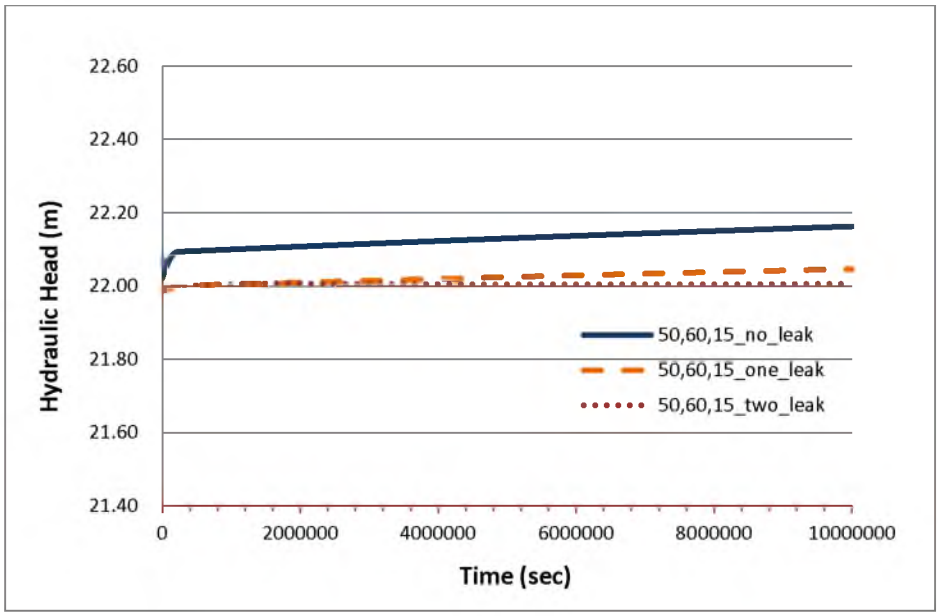


Fig. 5.20 The change of hydraulic head at node (50, 60, 15) due to each leakage.

Fig. 5.21 represents the hydraulic head propagation from two-dimensional slices on the second leakage node (56, 60, 9) and (56, 60, 11) during the third simulation. Fig. 5.21 (a) shows the hydraulic head distribution immediately preceding water injection. Fig. 5.21 (b) illustrates the hydraulic head distribution at 1,000,000 seconds after water injection and 100,000 seconds before the second leakage is generated. The increasing amount of head from water injection is meager because of the outflow from the first leakage pathway. Fig. 5.21 (c) presents the hydraulic head distribution at the time when the second leakage rate is induced. The aquifers have a rapid transient flow. Fig. 5.21 (d) and (e) illustrate simulated hydraulic head distributions at 7,000,000 and 10,000,000 seconds, respectively; the head distributions for two time step do not differ. This simulation is assigned a constant injection rate throughout the simulation time after 100,000 seconds, and constant boundary conditions at both left and right sides. Thus, it is assumed that the simulation reached equilibrium after the second leakage.

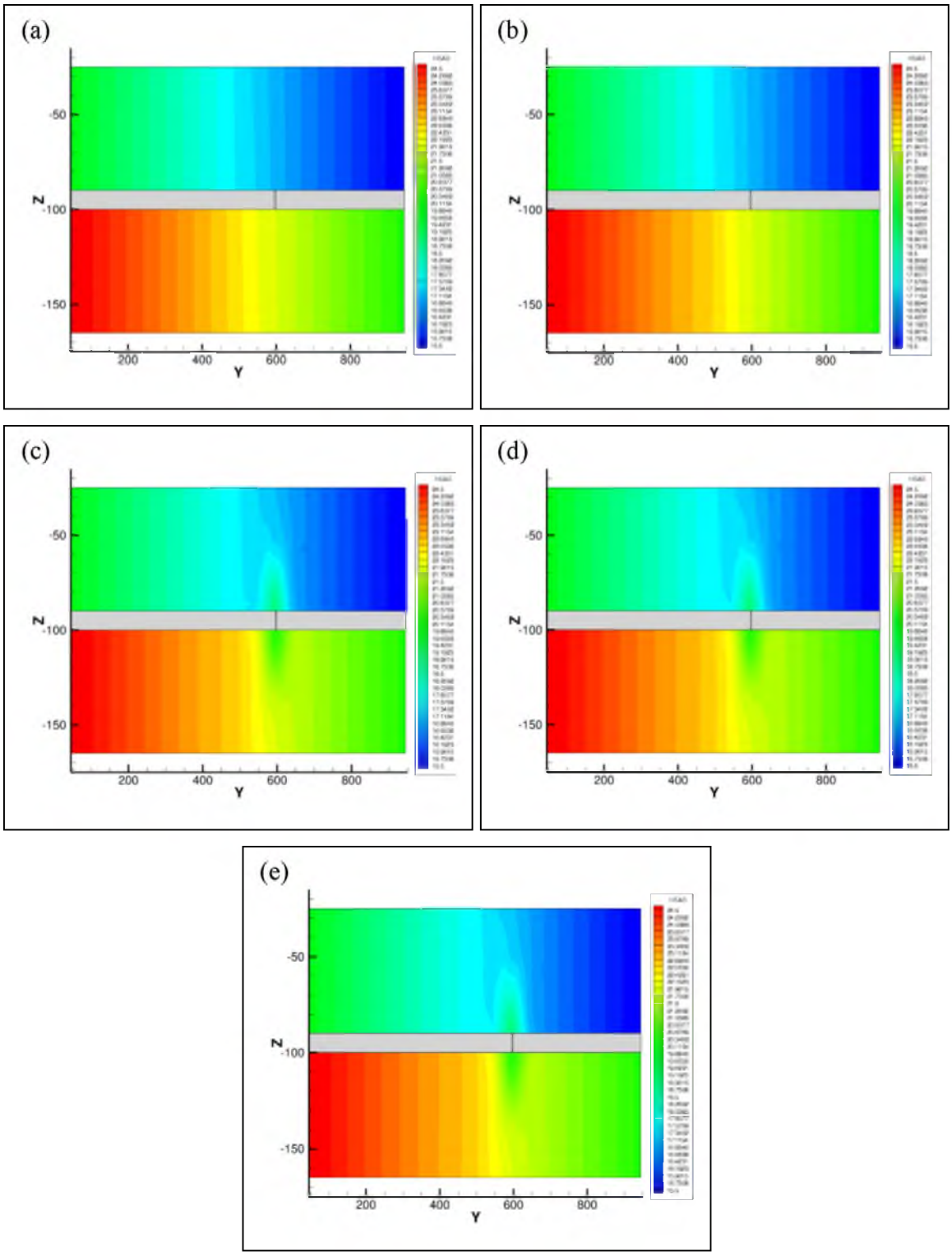


Fig. 5.21 Hydraulic head distribution around the second leakage pathway, (a) Simulation time: 0 seconds, (b) Simulation time: 1,000,000 seconds, (c) Simulation time: 2,000,000 seconds, (d) Simulation time: 7,000,000 seconds and (e) Simulation time: 10,000,000 seconds (Hydraulic head scale: 15.5 m- 24.5 m).

A log-log plot is used to illustrate truncation error distribution (Fig. 5.22). In Fig. 5.22, the logarithm of square of increase in cell sizes (Δx^2) of the model domain (indicated by the abscissa or x-axis) is plotted against the logarithm of errors in simulated hydraulic heads (indicated by the ordinate, or y-axis); this distribution corresponds to the end of simulation time.

To exhibit truncation error depending on cell sizes, the model domain was assigned to five different uniform grid block sizes: (dx, dy, dz) = (9.09 m, 9.09 m, 9.09 m), (11.00 m, 11.00 m, 11.00 m), (14.29 m, 14.29 m, 14.29 m), (20.00 m, 20.00 m, 20.00 m) and (33.33 m, 33.33 m, 33.33 m). Thus, the model domain was discretized to (110 × 110 × 22), (90 × 90 × 18), (70 × 70 × 14), (50 × 50 × 10) and (30 × 30 × 6) in the number of grid blocks, respectively.

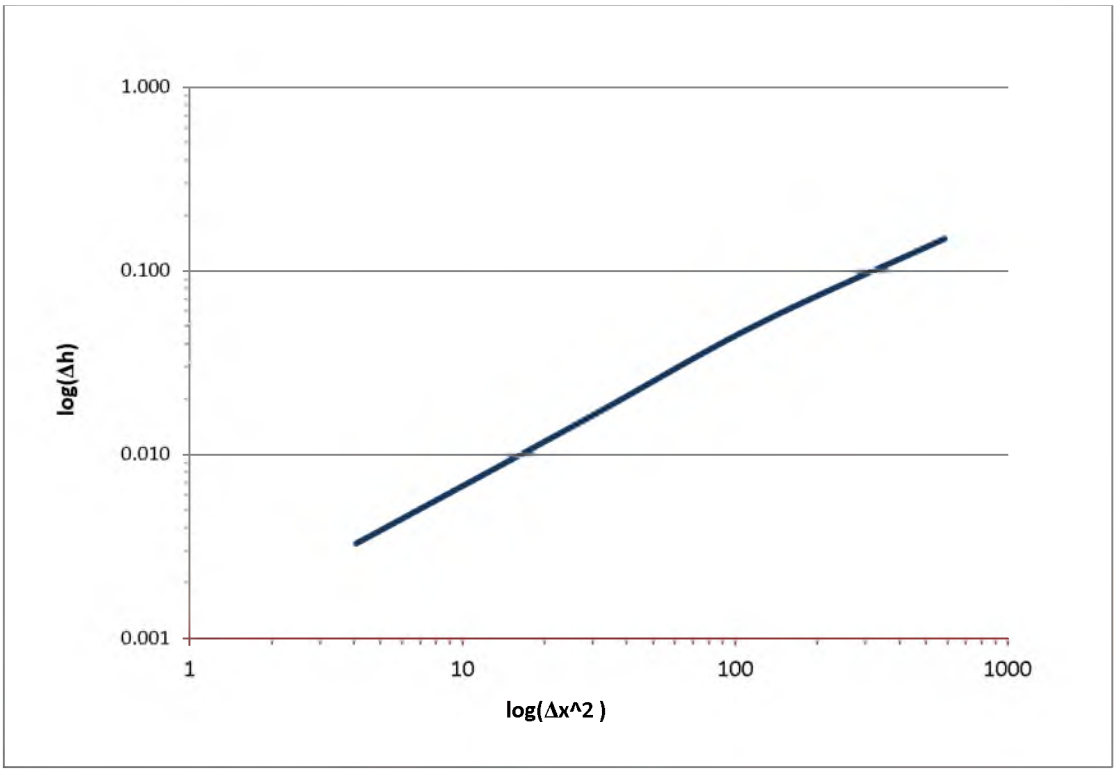


Fig. 5.22 Truncation error distribution.

Five simulations calculated the hydraulic heads in each model domain. The hydraulic heads at the same coordinate (x, y) = (760 m, 760 m, 170 m) in the five simulations were compared. The errors of the hydraulic heads were computed with respect to the finest grid blocks (110×110×22 cells). This plot (Fig. 5.22) for the new FDM model shows the truncation error as $O(\Delta x^2)$.

5.5 Inverse Modeling

In section 4.1.2, it was identified that uncertainty of the leakage pathway size results in reducing accuracy of leakage pathway estimation due to errors in calculation of pressure anomalies induced by leaks. Characterizing the geometry of initial guesses as meshes will increase not only the number of grid blocks in model domains but also the number of inverse modeling. The inversion may be repeated with various leakage pathway sizes (of initial guesses) to reduce the impact of unknown leakage pathway sizes. For instance, if the model domain includes one leakage pathway of size 0.3 m × 0.3 m, the inverse analysis should be iteratively conducted with a first model domain with the initial guesses meshed at 0.1 m × 0.1 m, a second model domain meshed at 0.2 m × 0.2 m for the initial guesses, the third domain meshed at 0.3 m × 0.3 m, etc. Objective function values calculated from many model domains designated to characterize each leakage pathway size should be compared to find its minimum value. A result from the model domain with the smallest objective function value will indicate the most possible leakage pathway location and size. If the model domain has multiple leakage pathways in various sizes, the number of inverse models may be increased to reduce uncertainty of leakage pathway sizes. Therefore, the sizes of leakage pathways have to be parameterized to more

effectively (and directly) apply the inverse modeling.

To estimate leakage pathways, the developed inverse model calibrates one parameter that integrates both average vertical hydraulic conductivities ($kzbleak_{i,j,k}$ in Equation (5.3)) and cross-sectional leakage areas ($Aleak_{i,j,k}$ in Equation (5.3)) between coupled leakage points, which make up a leakage term in the flow equation. This indicates the parameterization of leakage pathway geometry. In addition, estimating the integrated parameters ($kzbleak_{i,j,k} \cdot Aleak_{i,j,k}$) of initial guesses can increase efficacy of the inversion. If the two parameters ($kzbleak_{i,j,k}$ and $Aleak_{i,j,k}$) are separated in the inversion, the number of iterations of the inverse model will necessarily be increased because of increases in the number of parameters to be estimated. The number of solutions satisfying convergence criteria (nonuniqueness) may also increase. Calibration of integrated parameters including parameterization of geometry of leakage pathways can provide three advantages for effective inverse modeling: (1) reducing the number of grid blocks, (2) decreasing the impact of uncertainty in the geometry of leakage pathways and (3) diminishing the number of parameters to be estimated.

Each incorporated parameter value ($kzbleak_{i,j,k} \cdot Aleak_{i,j,k}$) of initial guesses of leakage pathways is explicitly part of the $Czbleak_{i,j,k}$ terms in the difference equations (Equation (5.9) and Equation (5.10)) and the developed forward model calculates anomalies of hydraulic heads due to the migration of leakage to another aquifer. If other parameters are needed to reduce uncertainty, the parameters must be part of the difference equations (the parameters associated with uncertainty in this study are discussed in section 5.5.2). A combination of parameter values is estimated by minimizing discrepancy between calculated and measured hydraulic heads.

An objective function of "least squares" type is used for calculating the discrepancy, and the minimizing discrepancy proceeds with a genetic algorithm (GA). The objective function of least squares is

$$S = \sum_{i=1}^{m} \frac{\mathbf{r}_i^2}{\sigma_{Z_i}^2}. \tag{5.13}$$

Here, σ_{Z_i} is the weighting coefficient for each observation and m is the number of calibration points. Fig. 5.23 summarizes the inverse modeling procedure.

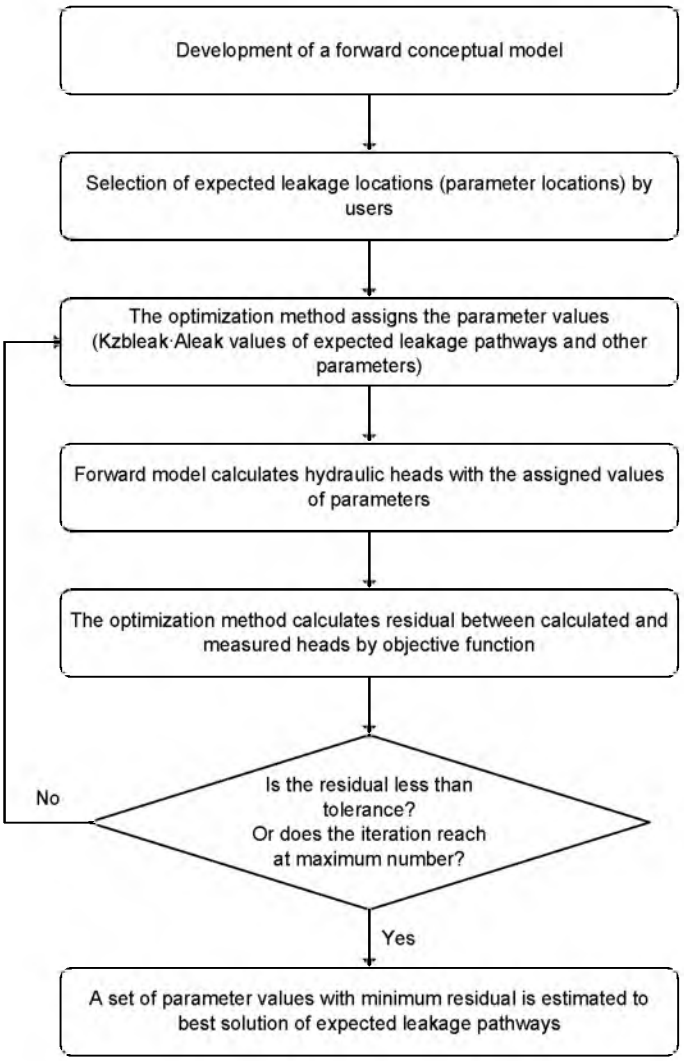


Fig. 5.23 Generalized protocol of inverse modeling.

This single-phase model was developed for preliminary storage of CO_2 in geological formations and associated potential CO_2 leakage before developing a full, multiphase method to simulate CO_2 storage with leakage pathways. However, as suggested previously, the developed inverse model is applied to leakage pathway estimation in brine/CO_2 systems using pressure anomalies induced by mobile brine into the overlying aquifer through leakage pathways.

5.5.1 Genetic Algorithm for Optimization Method

A genetic algorithm (GA) of a direct method type belongs to an evolutionary algorithm (EA) that mimics the process of natural evolution to generate solutions to optimize outcomes. The GA is generally utilized in decision analysis as an optimization method in hydrodynamics. Each set of random parameters is chosen within the given range for variables without a statistical function. The GA initially determines fitness about a randomly chosen parameter set within the given range, and improves parameters through repetitive application of reproduction, crossover, and mutation to generate optimum fitness (Rao, 2009).

In the GA, the population of a string which consists of individuals indicates the value of a parameter set, and the string is called a chromosome. Each individual indicates the value of the randomly selected parameters. The population is a candidate solution and evolves better solutions. The chromosomes are represented by strings which are made up of binary 0s and 1s. In each generation (a set of populations), the fitness of each population of randomly generated individuals is evaluated by an objective function, multiple populations are selected based on their fitness, and they evolve into a new

population by reproduction, crossover, and mutation. The new population is used in the next generation to get an increasingly improved population. The algorithm iterates until it reaches a maximum number of generations or the best fitness is calculated. The reproduction, crossover and mutation improve populations. This is explained below in detail.

(1) Reproduction: Reproduction is the selection operator. The reproduction operator is the first operation applied to the population to select good strings. In addition, the reproduction operator is used to pick above-average strings from the current population and insert their multiple copies in the mating pool.

Ex) String: 10001111110101000100100000110100011001001000 1

(2) Crossover: After reproduction, the crossover operator is implemented. The purpose of crossover is to create new strings by exchanging information among strings in the mating pool. In most crossover operators, two individual strings are picked at random from the mating pool generated by the reproduction operator and then some portions of the strings are exchanged between the strings. The two strings selected for participation in the crossover operators are known as parent strings and the strings generated by the crossover operator are known as child strings. The crossover site is usually chosen randomly.

Ex) Parent string: **1000111111010100010010000**0011010001100100100 01

Parent string: 10100111001110011001010**10110011101101010110 10**

By crossover, generation of child string (new string)

Child string: **1000111111010100010010010110011101101010110 10**

(3) Mutation: The crossover is the main operator by which new strings with better

fitness values are created for new generations. The mutation operator is applied to the new strings with a specific small mutation probability. The mutation operator changes the binary digit 1 to 0 and vice versa. The purpose of mutation is (a) to generate a string in the neighborhood of the current string, thereby accomplishing a local search around the current solution, (b) to safeguard against a premature loss of important genetic material at a particular position, and (c) to maintain diversity in the population.

In the inverse simulator developed as part of this dissertation, the GA developed by Carroll (2001) was implemented.

5.5.2 Applying Inverse Model of Single-phase to Multiphase Field

As mentioned in Chapter 3, in the multiple domains fully-saturated by brine, CO_2 injection continuously induces brine discharge through the leakage pathway before CO_2 flow reaches the bottom of the leakage pathway. The overlying formation (with only brine leakage) is a single-phase reservoir, at least in most situations, and for all simulations in this dissertation. With this context in mind, the inverse model for a single-phase fluid (developed as part of this dissertation) is applied to the overlying formation to estimate a leakage pathway for multiphase systems of brine and CO_2. Specifically, all simulations are conducted for the condition prior to CO_2 breakthrough into the overlying reservoir.

The homogeneous domain (see Fig. 3.1) under isothermal conditions (Chapter 3) was calibrated to estimate a leakage pathway using iTOUGH2, much like the simulation approach of Chapter 4. The simulation conditions are the same as the hydrogeological

properties of the model domain, boundary conditions, location of one injection well, CO_2 injection conditions, four monitoring wells and one leakage well (Fig. 3.1).

The initial condition of inversion is identical to the one in Chapter 4, i.e., the inversion is applied to the model domain as shown in Fig. 4.1 and 48 initial guesses of the leakage pathway are chosen. However, as mentioned earlier, the parameter values integrating both the vertical hydraulic conductivity and the cross sectional area of each initial guess of the leakage pathway are estimated as a means of leakage pathway detection.

As described in section 4.1.1, pressure anomalies in the overlying formation that are induced by leaks are critical to estimate possible leakage pathways. Therefore, pressure profiles in the overlying formation of the brine/CO_2 system at five observation points are used for leakage pathway estimation. In Fig. 3.11 pressure profiles from each monitoring well in the overlying formation exhibit sudden changes in pressure gradient because of capillary effects within the leakage pathway, induced by CO_2 leaks. Thus, the inverse modeling focuses on short-term brine leaks from 0 seconds to 100,000,000 seconds (approximately 3.17 years) before sudden changes in pressure gradient (Fig. 5.24).

Fig. 5.25 presents the YZ-plane of the leakage pathway located at coordinate (x, y) = (5250 m, 6050 m) in the model domain. To apply the single-phase inverse model to a multiphase domain and to obtain an accurate estimation of the leakage pathway location, the pressure distributions at the bottom of the leakage pathway (i.e., leak point in the 9^{th} layer in Fig. 5.25) between the single-phase and the multiphase simulations should be qualitatively consistent. This is examined in the next section.

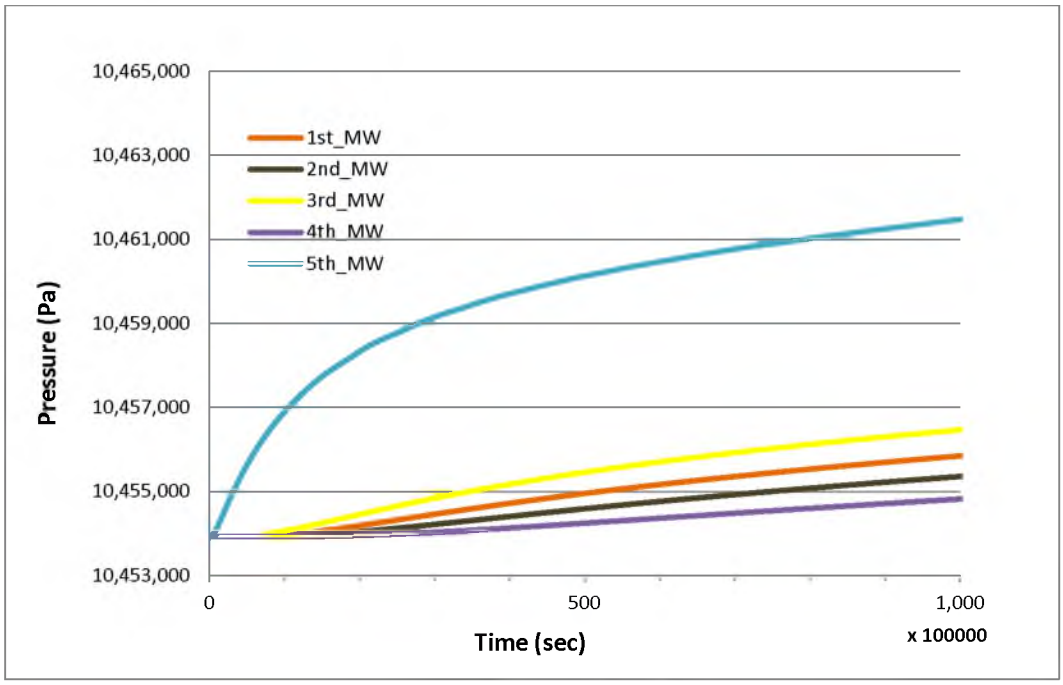


Fig. 5.24 Simulated pressure "measurements" from 0 to 3.17 years in the overlying formation.

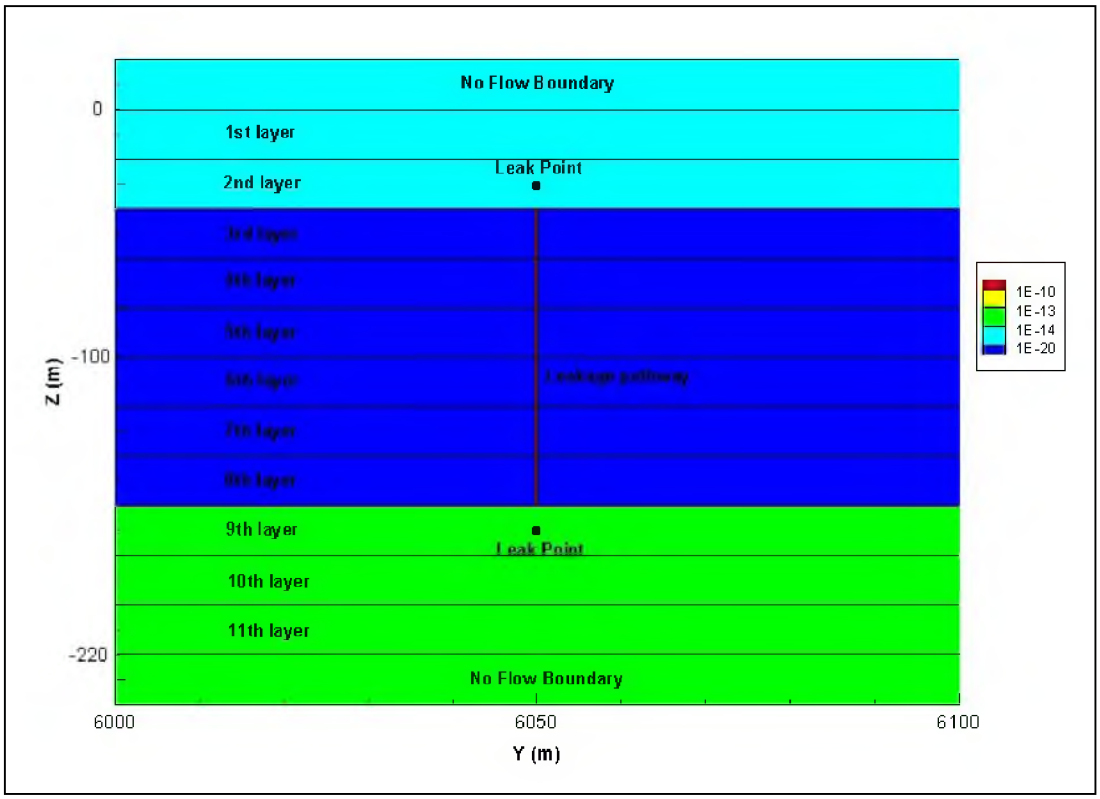


Fig. 5.25 YZ-plane of the leakage pathway in model domain (extended scale).

5.5.2.1 Comparing Pressures between Single-phase and Multiphase Flows

Per Darcy's law, the rate of leakage into the overlying aquifer depends on the vertical hydraulic conductivity and the cross-sectional area of the leakage pathway, and the hydraulic gradient between the overlying and the storage aquifers. The developed inverse model estimates the integrated parameter values of both the vertical hydraulic conductivity and cross-sectional area of each initial guess of the leakage pathway(s). When applying the single-phase inverse model to a multiphase system, the most important consideration is how the hydraulic gradient can be reasonably approximated by the single-phase simulator. That is, hydraulic head at the leak point in the 9^{th} layer (before CO_2 reaches bottom of the leakage pathway) of Fig. 5.25 must be approximated, by the single-phase model, as exactly as possible. Because this study focuses on only brine leaks before CO_2 leaks, the hydraulic head at the bottom of the leakage pathway can be approximated by the single-phase model. The hydraulic head at the leak point in the 2^{nd} layer can be quantitatively calculated as the single phase domain.

Of particular importance is to examine the hydraulic head distributions before CO_2 leaks into the leak point in the 9^{th} layer. As mentioned in section 5.4.1.1, the FDM model uses volumetric injection rates, so CO_2 mass injection rates are converted to volumetric injection rates. The volumetric injection rate of CO_2 is assigned to the volumetric injection rate of water in the FDM model. The pressures from the FDM model are compared to the actual pressure data of multiphase domain at the bottom of the leakage well (i.e., at the leak point in the 9^{th} layer). This comparison is used to investigate if the single-phase model can approximately realize pressure distributions in multiphase formation before CO_2 leaks. As such, CO_2 density at the bottom of the injection well

should be evaluated to calculate the appropriate volumetric injection rates. Fig. 5.26 illustrates simulated pressure and CO_2 density distributions at the bottom of the injection well in the actual brine/CO_2 system during the period of leakage (3.17 years). As shown in Fig. 5.26, as soon as CO_2 is initially injected into the storage aquifer of the model domain, pressure at the bottom of the injection well substantially increases because CO_2 injection pressure has to exceed the capillary pressure at the bottom of the injection well. After CO_2 is injected, the pressure abruptly drops because the bottom of the injection well no longer includes capillary effects. On the other hand, the sudden drop of significant pressure induces fluctuations of pressure. CO_2 density at the bottom of the injection well is increased by the pressure buildup during the initial CO_2 injection period, but the CO_2 density is decreased due to the drop of pressure over time.

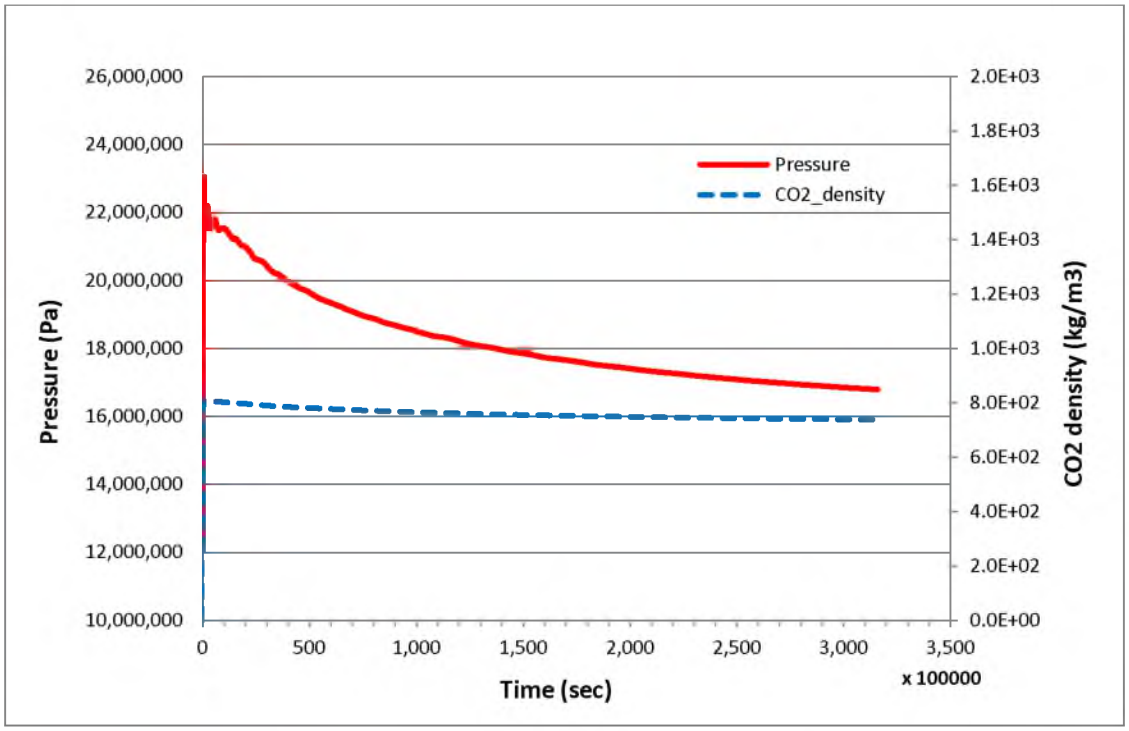


Fig. 5.26 Simulated pressure and CO_2 density distributions at the bottom of the injection well.

Table 5.9 summarizes the change of CO_2 density at the bottom of the injection well, CO_2 mass injection rates, and CO_2 volumetric injection rates for the 3.17 year leakage periods. Table 5.10 lists average brine density and viscosity in the entire system and approximates hydraulic conductivity values calculated by the average density and viscosity. The approximate hydraulic conductivity values are assigned to both aquifers and a cap rock in the single-phase model. Table 5.11 summarizes specifications of the model domain. Specific storage is calculated from Equation (5.12).

Table 5.9 CO_2 density, CO_2 mass injection rates, and CO_2 volumetric injection rates at the injection well.

Simulation time (sec)	**CO_2 density (kg/m^3)**	**CO_2 mass injection rate (kg/s)**	**CO_2 volumetric injection rate (m^3/s)**
0 – 100,000,000 (3.17 yrs)	805.4 ~ 765.4	63.4	0.0788 ~ 0.0829

Table 5.10 Brine properties and hydraulic conductivities.

Density (kg/m^3)	1,025.5	**Permeability (m^2)**	**Overlying aquifer**	$kx = ky = kz =$ 1.0e-15
			Storage aquifer	$kx = ky = kz =$ 1.0e-13
			Cap rock	$kx = ky = kz =$ 1.0e-20
Viscosity (kg/m·s)	0.00055	**Hydraulic conductivity (m/s)**	**Overlying aquifer**	$Kx = Ky = Kz =$ 1.8285e-08
			Storage aquifer	$Kx = Ky = Kz =$ 1.8285e-06
			Cap rock	$Kx = Ky = Kz =$ 1.8285e-13

Table 5.11 Specifications of the model domain.

Domain size (m)	10,100×10,100×220	**Simulation time (sec)**		0 ~ 100,000,000 (3.17 yrs)
Each cell size (m)	100×100×20	**Time step size (sec)**		10,000
Number of cells	103 × 103 × 11 (112,211 total)	**Tolerance**		1e-7
Leakage pathway size (m)	0.3 × 0.3	**Leakage pathway**	**Permeability (m^2)**	$kx = ky = kz =$ 1.0e-10
Leakage pathway location	(5250 m, 6050 m)		**Hydraulic conductivity (m/s)**	$Kx = Ky = Kz =$ 1.8285e-03
Water compressibility (Pa^{-1})	3.5×10^{-10}	**Porosity**	**Both aquifers**	0.2
			Cap rock	0.02
Pore compressibility (Pa^{-1})	0.0	**Specific storage (m^{-1})**	**Both aquifers**	7.0×10^{-7}
			Cap rock	7.0×10^{-8}

The minimum and maximum volumetric injection rates of CO_2 in Table 5.9 are assigned to the water injection rates in the developed forward model. Fig. 5.27 illustrates both the multiphase model simulated pressure by mass injection rate of 63.4 kg/s (red line) and the approximated-pressure from the single-phase (new FDM) model by the volumetric injection rate of 0.0829 m^3/s (blue line) at the leak point in the 9^{th} layer (Fig. 5.25). Comparison of the two simulators' resulting pressure distributions by volumetric injection rate of 0.0788 m^3/s is shown in Fig. 5.28. The maximum relative errors between the multiphase pressure and the approximated single-phase pressure are 0.0237 (2.37 %) and 0.0396 (3.96 %) in the two models, respectively.

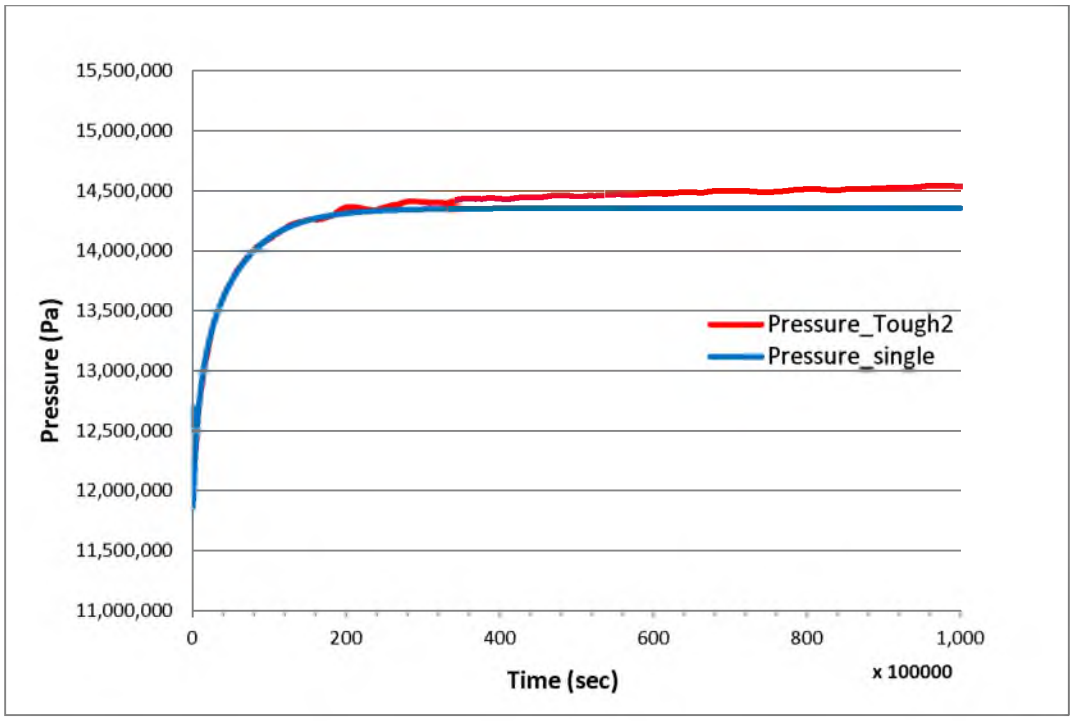


Fig. 5.27 Simulated pressure distributions at the bottom of the leakage pathway (0.0829 m^3/sec volumetric injection rate).

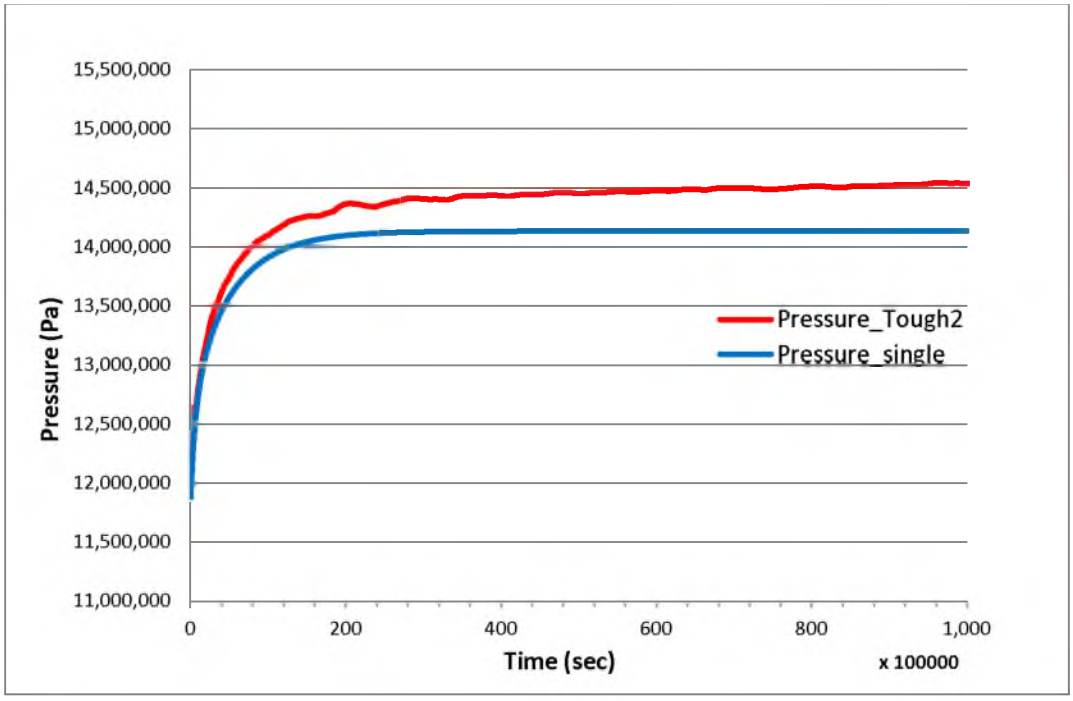


Fig. 5.28 Simulated pressure distributions at the bottom of the leakage pathway (0.0788 m^3/sec volumetric injection rate).

In spite of the errors between the multiphase and single-phase simulated pressures, both sets of results exhibit similar trends regardless of the magnitude of the two volumetric injection rates. In addition, the errors between the multiphase and single-phase pressures can be reduced by adjusting the approximate volumetric injection rates. Thus indicating that, in terms of only brine leakage, the pressure (or hydraulic head) distributions at the leak point in the storage aquifer can be approximated by the single-phase forward model using an appropriately adjusted (calibrated) value of volumetric injection rate. Based on these results, it is assumed that a single-phase inverse model may suffice for application to a multiphase domain for leakage pathway estimation. The inversion results can be improved if the volumetric injection rates are parameterized to reduce the errors between the multiphase and single-phase pressure distributions. Furthermore, the parameterized volumetric injection rates may reduce the impact of parameter uncertainties in the storage formation even if the uncertain parameters are not estimated in the inversion.

Fig. 5.29 illustrates the multiphase pressure distribution (solid red line) for 10 years and change of gaseous CO_2 saturation (dashed red line) at the leak point in the storage aquifer. In Fig. 5.29, the blue line indicates the single-phase pressure distribution with a volumetric injection rate of 0.0829 m^3/s for 3.17 years. After CO_2 reaches the bottom of the leakage pathway, the multiphase pressure increases due to capillary pressure and error associated with using single-phase (for multiphase) is significantly increased. Thus, applying the developed inverse model to the model domain after CO_2 reaches the leakage pathway will degrade the veracity of the leakage pathway estimation. Inversion using the developed single-phase model is discussed in the next section.

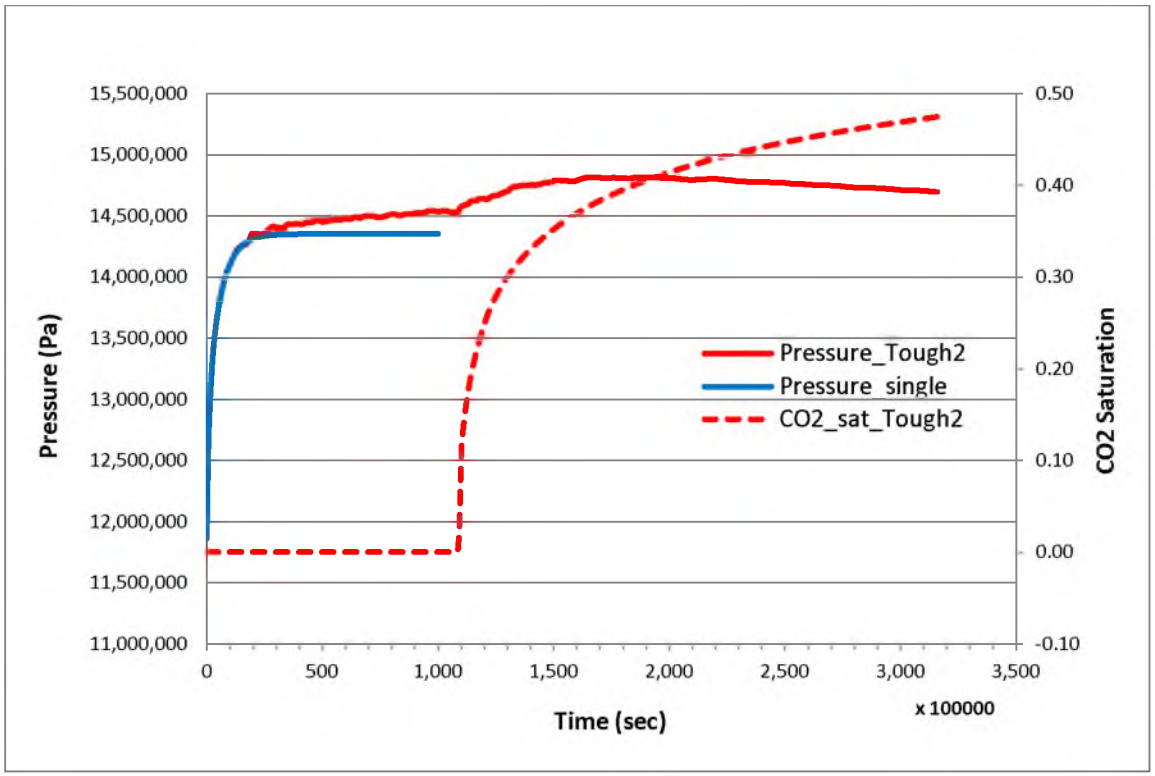


Fig. 5.29 Simulated pressure distributions at the bottom of the leakage pathway for 10 years (0.0829 m^3/sec volumetric injection rate).

5.5.2.2 Results of Inverse Modeling using New FDM Model

As mentioned before, the developed inverse model calibrates each integrated parameter of 48 initial guesses of leakage pathway, so the 48 inverse simulations were conducted to estimate each initial guess. As described in section 5.4.1.2, the bulk hydraulic conductivities characterized by average density and viscosity of brine might lead to significant errors in calculated results (hydraulic head). Therefore, the hydraulic conductivity of the overlying formation is calibrated with inverse analysis to reduce errors. As described in the previous section, the volumetric injection rate is also calibrated in the inversion to minimize errors between the multiphase and single-phase pressure distributions at the bottom of the leakage pathway. That is, the inversion simultaneously estimates three parameters: (1) each integrated parameter of both vertical

hydraulic conductivity and cross-sectional area each of the 48 initial guesses of the leakage pathway, (2) hydraulic conductivity of the overlying formation, and (3) volumetric injection rate of water. The genetic algorithm (GA) estimates the optimum combination of the three parameters to minimize the objective function.

In the inverse modeling, the forward simulator iteratively runs to generate model results, and the number of grid blocks is minimized to reduce computational expense. For effective inverse modeling, the number of cells in the domain is reduced to 14×14×11 (2,156 grid blocks total) and each cell size is varied. In the new FDM model, the objective function calculates residuals between calculated hydraulic heads and measured hydraulic heads. "Measurements" (see Fig. 5.24) were converted from pressure to hydraulic head (m). The weighting coefficient of 1 m was used for all measurements in the objective function because only the measurements in the overlying formation are used (so the magnitude of the measurements does not need to be scaled). Table 5.12 summarizes parameters of inversion for leakage pathway estimation.

Fig. 5.30 shows a contour plot of the objective function values through estimation of the three parameters. A circle in Fig. 5.30 indicates the actual leakage pathway. The minimum objective function value expresses the best fit between the measured and simulated pressures. Therefore, the initial guess with a minimum convex area of the objective function is considered to be the best estimation, i.e., the most possible location of the leakage well. In Fig. 5.30, the result of inverse analysis presents two global minima around two sets of coordinates (5250 m, 6150 m) and (5150 m, 6350 m). However, this contour plot includes some inherent error in interpolation to exhibit continuous objective function values. Exact objective function values from inversion are listed in Table 5.13.

Table 5.12 Inverse analysis parameters for leakage pathway estimation.

	Integrated parameter (m^3/s)	**Hydraulic conductivity** (m/s)	**Volumetric injection rate** (m^3/s)	**GA**	
				Generation	**Population**
Range of parameters	1.0e-8 ~ 1.0e-4	1.0e-9 ~ 1.0e-7	0.075 ~ 0.088	900	10

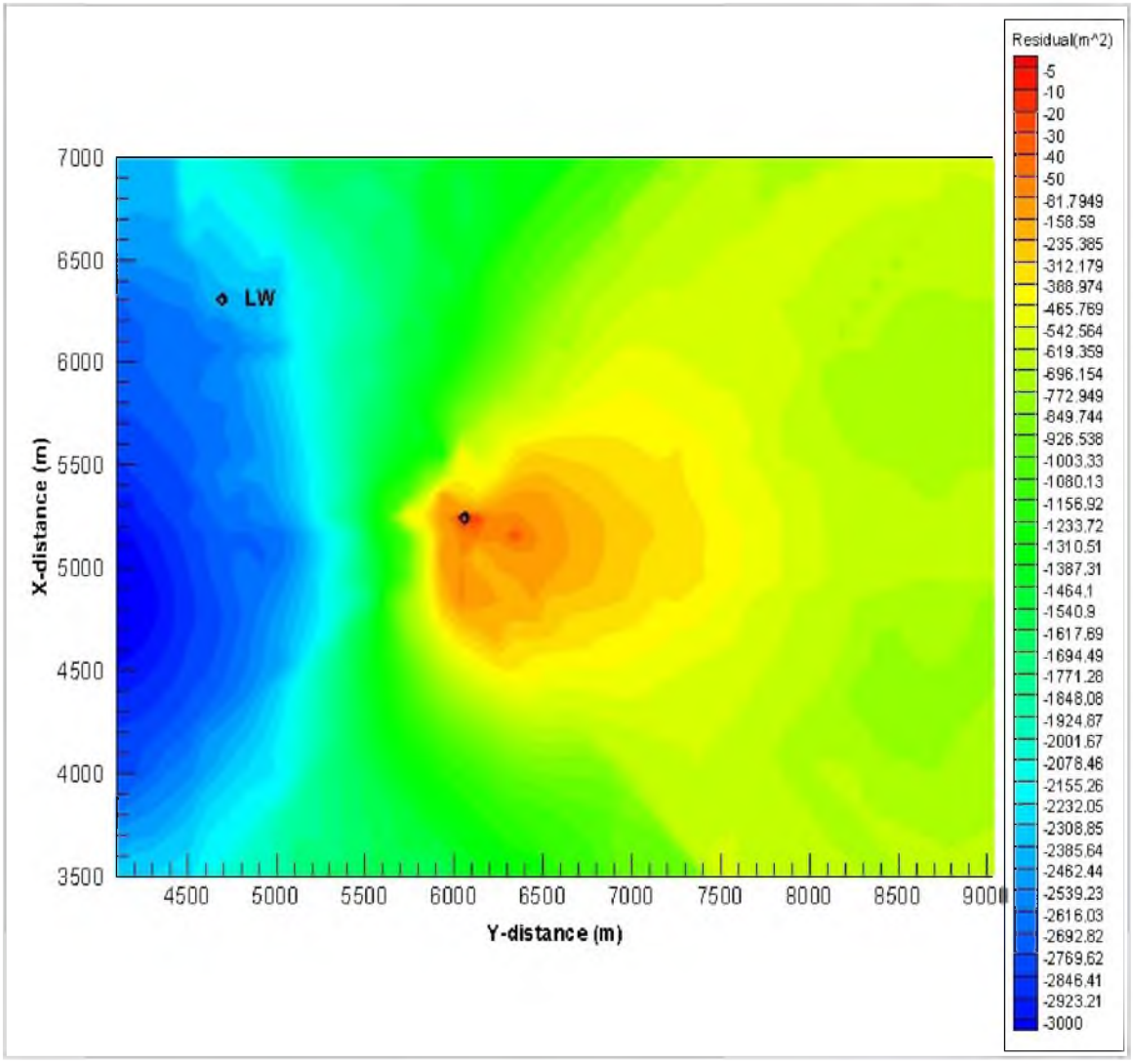


Fig. 5.30 Contour plot of the objective function values.

Table 5.13 Objective function values from estimated initial guesses of the leakage pathway.

Initial guess		**Objective function (m^2)**
Number	**Coordinate (m)**	
True	**(5250, 6050)**	-
6	(4850, 6050)	-0.42589E+02
7	(4850, 6950)	-0.30506E+03
11	(5050, 5950)	-0.28955E+02
12	(5050, 6150)	-0.27292E+03
16	(5150, 5750)	-0.10986E+04
17	(5150, 5950)	-0.13918E+03
18	(5150, 6050)	-0.62386E+02
19	(5150, 6150)	-0.39099E+02
20	(5150, 6350)	-0.20728E+02
24	(5250, 5950)	-0.83565E+02
25	**(5250, 6050)**	**-0.30843E+01**
26	(5250, 6150)	-0.11021E+02
31	(5350, 5950)	-0.50698E+02
33	(5350, 6150)	-0.38522E+03
34	(5350, 6350)	-0.13358E+03

In Table 5.13, the inversion estimated the 25th initial guess (5250 m, 6050 m) as the most possible leakage pathway location. The estimated leakage well location is identical to the actual leakage well location. Fig. 5.31 and Fig. 5.32 present residuals between the calculated and measured hydraulic heads with respect to the 20th and 25th initial guesses (see Table 5.13). As shown in Fig. 5.31 and Fig. 5.32, the residual values in the 25th initial guess are further improved on the whole, particularly at MW5.

Table 5.14 denotes the arithmetic means and the standard deviations of the objective function values and three parameters estimated by the 48 permutations.

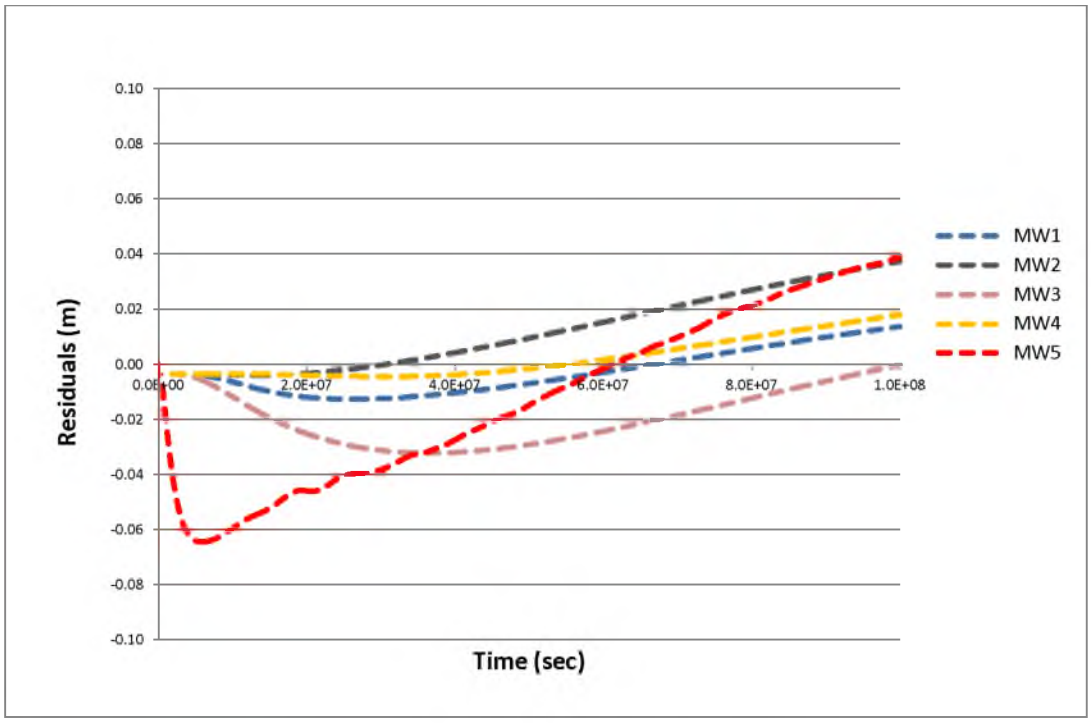


Fig. 5.31 Residuals corresponding to the 20th initial guess.

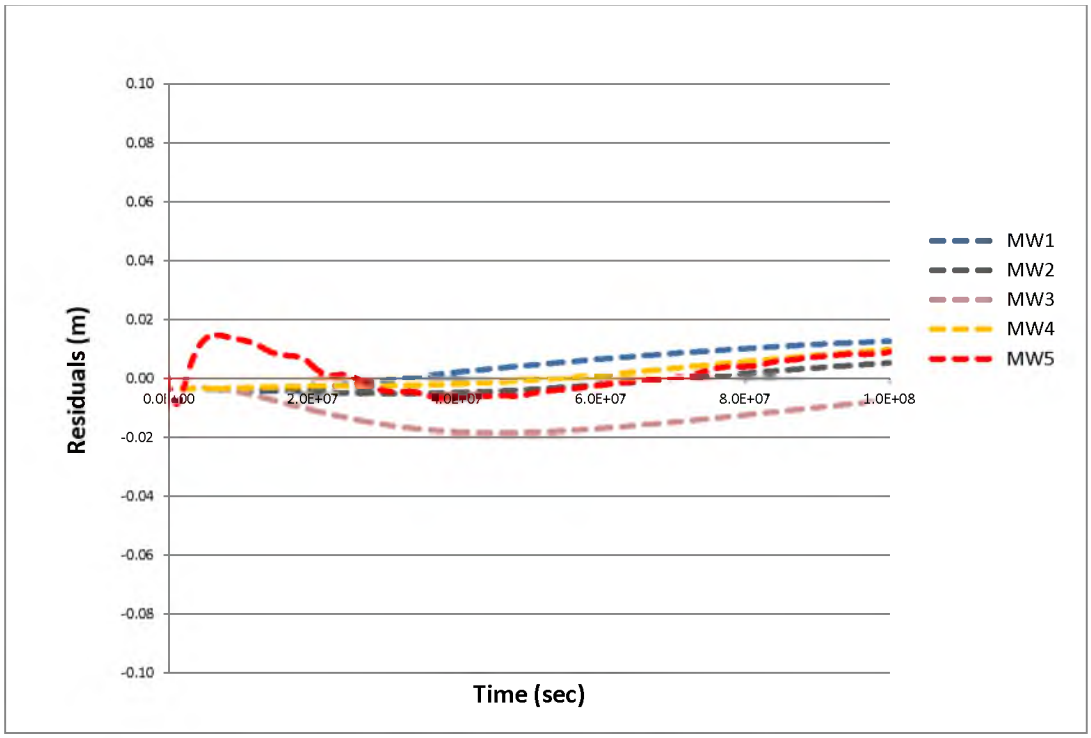


Fig. 5.32 Residuals corresponding to the 25th initial guess.

Table 5.14 Statistics of estimated parameters.

Objective function (m^2)		**Integrated parameter (m^3/s)**		**Hydraulic conductivity (m/s)**		**Volumetric injection rate (m^3/s)**	
-		0.165e-3*		0.18285e-07*		0.0788 ~ 0.0829*	
Average	Std. dev.	Average	Std. dev.	Average	Std. dev.	Average	Std. dev.
-0.947e3	0.996e3	0.119e-4	0.262e-4	0.212e-6	0.322e-6	0.818e-1	0.500e-2

* Values indicates initial approximation

This inversion result suggests that the newly-developed inverse model of single-phase fluid can be applied for leakage pathway estimation in a multiphase brine/CO_2 flow system, if the inversion focuses on only brine leaks into the overlying formation (e.g., before the CO_2 breaks through into that overlying formation), and if an appropriately-calibrated volumetric injection rate of water is used to represent CO_2 injection.

5.6 Summary and Conclusions

The new FDM model provides forward analysis of transient flows due to leakage and inverse analysis to estimate leakage pathways in a single-phase flow system. The forward model simultaneously calculates hydraulic head anomalies due to water leaks using coupled leakage points in two reservoirs. The inverse model is composed of the developed forward model and the genetic algorithm. The inverse model estimates the possible leakage pathway locations through parameters integrating vertical hydraulic conductivity and the cross- sectional area between coupled leak points. The integrated parameterization of leakage pathway properties provides three advantages for effective inverse modeling:

(1) Reducing the number of grid blocks

(2) Decreasing the impact of uncertainty in leakage pathway size in calibrations

(3) Diminishing the number of parameters to be estimated

Validation of the developed forward model was investigated through application of the multiphase model, TOUGH2 with a two-dimensional example. The relative errors between both models were less than 0.0135% at leakage and no leakage conditions.

The developed inverse model was applied to estimate location(s) of a leakage pathway(s) in a brine/CO_2 system. The inversion used pressure profiles at the overlying formation during the period of only brine leaks. The inversion calibrated three kinds of parameters as follows:

(1) Each integrated parameter of both vertical hydraulic conductivity and cross-sectional area of 48 initial guesses of leakage pathways

(2) Hydraulic conductivity of overlying formation

(3) Volumetric injection rate of water

The integrated parameters of 48 initial guesses were estimated for leakage pathway estimation. The hydraulic conductivity of the overlying formation was calibrated to reduce errors in calculated outputs due to uncertainty of brine density and viscosity. The volumetric injection rate was estimated to minimize erroneous hydraulic gradients between the overlying and the storage aquifers. The estimated leakage well location was identical to the actual leakage well location. Therefore, it was identified that the inverse model for single-phase fluid can be applied to the leakage pathway estimation in the brine/CO_2 flow system. The applicability of single-phase model to multiphase systems with leakage issues in CCUS should provide three advantages:

(1) The inverse modeling can be free from errors in capillary pressure and relative permeability functions to realize multiphase flow.

(2) The computational expense is reduced since the forward model of the single-phase has simpler logic than a multiphase simulator. Therefore, the inverse modeling of single-phase should be further effective in terms of simulation time.

(3) The goal of this application is to provide warning before CO_2 leaks, and is intended to be helpful in mitigating and managing the risk of CO_2 leaks.

5.7 References

Altevogt, A. S., and Celia, M. A. (2004). "Numerical modeling of carbon dioxide in unsaturated soils due to deep subsurface leakage." *Water Resources Research*, 40(3).

Altunin, V. (1975). *Thermophysical properties of carbon dioxide*, Publishing House of Standards, Moscow, Russia.

Anderson, M. P., and Woessner, W. W. (1992). *Applied groundwater modeling: simulation of flow and advective transport*, Elsevier Science, Oxford, UK.

Avci, C. B. (1994). "Evaluation of flow leakage through abandoned wells and boreholes." *Water Resources Research*, 30(9), 2565-2578.

Babbar, M., and Minsker, B. (2006). "Groundwater remediation design using multiscale genetic algorithms." *Journal of Water Resources Planning and Management*, 132(5), 341-350.

Beckford, O., Hilton, A. C., and Liu, X. "Development of an enhanced multi-objective robust genetic algorithm for groundwater remediation design under uncertainty." *Proc., Proc. Am. Soc. Civ. Eng*, 98.

Benjamin, J. R. (1970). *Probability, statistics, and decision for civil engineers*, McGraw-Hill, New York, NY.

Bennett, G. D. (1976). *Introduction to groundwater hydraulics*, U.S. Geological Survey, Denver, CO.

Bevington, P., and Robinson, D. (1969). *Data reduction and data analysis for the physical sciences*, McGraw-Hill, New York, NY.

Carroll, D. L. 2001. Fortran genetic algorithm driver. http://cuaerospace.com/carroll/ga.

Carroll, S., Hao, Y., and Aines, R. (2009). "Transport and detection of carbon dioxide in dilute aquifers." *Energy Procedia*, 1(1), 2111-2118.

Chan-Hilton, A. B., Iyer, S. K., Magar, V., and Kelley, M. "Optimization of natural attenuation with active remediation under uncertainty." *Proc., Seventh International In Situ and On-Site Bioremediation Symposium, Orlando, Florida, USA, 2-5 June 2003. Part H. Natural Attenuation, Long-Term Monitoring, and Site Closure*, Battelle Press, Columbus, OH.

Chen, C. S. (1989). "Solutions approximating solute transport in a leaky aquifer receiving wastewater injection." *Water Resources Research*, 25(1), 61-72.

Chen, Z., Huan, G., and Ma, Y. (2006). *Computational methods for multiphase flows in porous media*, Siam, Philadelphia, PA.

Christensen, S., and Cooley, R. (1999). "Evaluation of confidence intervals for a steady-state leaky aquifer model." *Advances in Water Resources*, 22(8), 807-817.

Cihan, A., Zhou, Q., and Birkholzer, J. T. (2011). "Analytical solutions for pressure perturbation and fluid leakage through aquitards and wells in multilayered-aquifer systems." *Water Resources Research*, 47(10), W10504.

Cobb, P., McElwee, C., and Butt, M. (1982). "Analysis of leaky aquifer pumping test data: An automated numerical solution using sensitivity analysis." *Groundwater*, 20(3), 325-333.

Collins, R. E. (1961). *Flow of fluids through porous materials*, Reinhold Publishing Corp., New York, NY.

Committee, I. F. (1967). "A formulation of the thermodynamic properties of ordinary water substance." *IFC Secretariat, Düsseldorf, Germany*, 26.

Corey, A. T. (1954). "The interrelation between gas and oil relative permeabilities." *Producers Monthly*, 19(1), 38-41.

Deming, D. (1994). "Factors necessary to define a pressure seal." *AAPG bulletin*, 78(6).

Doughty, C., and Pruess, K. (2004). "Modeling supercritical carbon dioxide injection in heterogeneous porous media." *Vadose Zone Journal*, 3(3), 837-847.

Espinoza, F. P., Minsker, B. S., and Goldberg, D. E. (2005). "Adaptive hybrid genetic algorithm for groundwater remediation design." *Journal of Water Resources Planning and Management*, 131(1), 14-24.

Finsterle, S. (2004). "Multiphase inverse modeling." *Vadose Zone Journal*, 3(3), 747-762.

Finsterle, S. (2007a). "iTOUGH2 user's guide." *Report LBNL-40040*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007b). "iTOUGH2 command reference." *Report LBNL-40041*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007c). "iTOUGH2 sample problems." *Report LBNL-40042*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2010). "iTOUGH2 V3. 2, verification and validation report." *Report LBNL-42002*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S., Moridis, G., and Pruess, K. (1994). "A tough2 equation-of-state module for the simulation of two-phase flow of air, water, and a miscible gelling liquid." *Report LBL-36086*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Garcia, J. E. (2001). "Density of aqueous solutions of CO2." *Report LBNL-49023*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Gasda, S. E., Bachu, S., and Celia, M. A. (2004). "Spatial characterization of the location of potentially leaky wells penetrating a deep saline aquifer in a mature sedimentary basin." *Environmental Geology*, 46(6-7), 707-720.

Gasda, S. E., Wang, J. Z., and Celia, M. A. (2011). "Analysis of in-situ wellbore integrity data for existing wells with long-term exposure to CO2." *Energy Procedia*, 4, 5406-5413.

Han, W. S., McPherson, B. J., Lichtner, P. C., and Wang, F. P. (2010). "Evaluation of trapping mechanisms in geologic CO2 sequestration: Case study of SACROC northern platform, a 35-year CO2 injection site." *American Journal of Science*, 310(4), 282-324.

Harbaugh, A. W. (2005). *MODFLOW-2005, the US Geological Survey modular ground-water model: The ground-water flow process*, US Department of the Interior, US Geological Survey.

Hou, Z., Murray, J. C., and Rockhold, L. M. (2012). "CO2 migration in intact caprock and leakage risk in three-dimensional heterogeneous formations." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittssburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012a). "Early detection of brine or CO2 leakage through high-permeability pathways using pressure-based monitoring data." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittsburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012b). "Impact of data uncertainty on identifying leakage pathways in CO2 geologic storage systems and estimating their hydrogeological properties by inverse modeling." *TOUGH Symposium 2012,* Lawrence Berkeley National Laboratory, Berkeley, CA.

King, P. (1989). "The use of renormalization for calculating effective permeability." *Transport in Porous Media*, 4(1), 37-58.

Ko, N.-Y., and Lee, K.-K. (2008). "Reliability and remediation cost of optimal remediation design considering uncertainty in aquifer parameters." *Journal of Water Resources Planning and Management*, 134(5), 413-421.

Kowalsky, M. B., Finsterle, S., and Rubin, Y. (2004). "Estimating flow parameter distributions using ground-penetrating radar and hydrological measurements during transient flow in the vadose zone." *Advances in Water Resources*, 27(6), 583-599.

Krevor, S., Perrin, J.-C., Esposito, A., Rella, C., and Benson, S. (2010). "Rapid detection and characterization of surface CO2 leakage through the real-time measurement of δ13 C signatures in CO2 flux from the ground." *International Journal of Greenhouse Gas Control*, 4(5), 811-815.

Liggett, J. A., and Chen, L.-C. (1994). "Inverse transient analysis in pipe networks." *Journal of Hydraulic Engineering*, 120(8), 934-955.

Metz, B., Davidson, O., de Coninck, H., Loos, M., and Meyer, L. (2005). *IPCC special report on carbon dioxide capture and storage*, Cambridge University Press, New York, NY.

Nogues, J. P., Nordbotten, J. M., and Celia, M. A. (2011). "Detecting leakage of brine or CO2 through abandoned wells in a geological sequestration operation using pressure monitoring wells." *Energy Procedia*, 4, 3620-3627.

Nordbotten, J. M., Celia, M. A., and Bachu, S. (2004). "Analytical solutions for leakage rates through abandoned wells." *Water Resources Research*, 40(4), W04204.

Nordbotten, J. M., Kavetski, D., Celia, M. A., and Bachu, S. (2008). "Model for CO2 leakage including multiple geological layers and multiple leaky wells." *Environmental Science & Technology*, 43(3), 743-749.

Onuma, T., and Ohkawa, S. (2009). "Detection of surface deformation related with CO2 injection by DInSAR at In Salah, Algeria." *Energy Procedia*, 1(1), 2177-2184.

Phillips, S. L., Igbene, A., Fair, J., Ozbek, H., and Tavana, M. (1981). *A technical databook for geothermal energy utilization*, Lawrence Berkeley Laboratory, University of California, Berkeley, CA.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in Fortran 77: The art of scientific computing second edition*, Cambridge University Press, New York, NY.

Pruess, K. (1987). "TOUGH user's guide, nuclear regulatory commission report NUREG/CR-4645." *Report LBL-20700*, Lawrence Berkeley Laboratory Berkeley, CA.

Pruess, K. (1991). "TOUGH2: A general-purpose numerical simulator for multiphase fluid and heat flow." *Report LBL-29400*, Lawrence Berkeley Laboratory, Berkeley, CA.

Pruess, K. (2004). "Numerical simulation of CO2 leakage from a geologic disposal reservoir, including transitions from super-to subcritical conditions, and boiling of liquid CO2." *Spe Journal*, 9(2), 237-248.

Pruess, K. (2005). "ECO2N: A TOUGH2 fluid property module for mixtures of water, NaCl, and CO2." *Report LBNL-57952*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Pruess, K., and García, J. (2002). "Multiphase flow dynamics during CO2 disposal into saline aquifers." *Environmental Geology*, 42(2-3), 282-295.

Pruess, K., Moridis, G., and Oldenburg, C. (1999). "TOUGH2 user's guide, version 2.0." *Report LBNL-43134*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Rao, S. S. (2009). *Engineering optimization: Theory and practice*, John Wiley & Sons, Hoboken, New Jersey.

Singh, S. K. (2009). "Simple method for quick estimation of leaky-aquifer parameters." *Journal of Irrigation and Drainage Engineering*, 136(2), 149-153.

Spycher, N., and Pruess, K. (2005). "CO2-H2O mixtures in the geological sequestration of CO2. II. Partitioning in chloride brines at 12–100° C and up to 600 bar." *Geochimica et Cosmochimica Acta*, 69(13), 3309-3320.

Sun, A. Y., Zeidouni, M., Nicot, J.-P., Lu, Z., and Zhang, D. (2013). "Assessing leakage detectability at geologic CO2 sequestration sites using the probabilistic collocation method." *Advances in Water Resources*, 56, 49-60.

Van Genuchten, M. T. (1980). "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils." *Soil Science Society of America Journal*, 44(5), 892-898.

Zheng, C., Bennett, G. D., Melton, J., and Simon, A. R. (2002). "Applied contaminant transport modeling." *Industrial and Commercial Training*, 34(7), 256-262.

Zhou, Q., Birkholzer, J. T., and Tsang, C.-F. (2009). "A semi-analytical solution for large-scale injection-induced pressure perturbation and leakage in a laterally bounded aquifer–aquitard system." *Transport in Porous Media*, 78(1), 127-148.

CHAPTER 6

SUMMARY, CONCLUSION AND RECOMMENDATION

6.1 Summary and Conclusion

The objective of this dissertation is to estimate a leakage pathway location such as abandoned wells, or other possible leakage zones, from porous media using an inverse analysis. This research is associated with the storage of CO_2 in geological saline formations. The storage of CO_2 in deep geological formations has risks of CO_2 leakage. The geological reservoirs used to store CO_2 must have a high porosity and permeability, and a low-permeability cap rock lying above the reservoir. Leakage pathways penetrating the cap rock layer may cause CO_2 to release into other reservoirs lying above the cap rock. Therefore, inverse analysis was applied to estimate the leakage pathway through pressure anomalies in an overlying aquifer induced by brine or CO_2 leaks.

In this dissertation, I assessed detectability of the leakage pathway by using an iTOUGH2 model for multiphase inverse modeling. In addition, I developed a single-phase model to perform leakage pathway estimation in a multiphase system. The application of inverse analysis was conducted based on the uncertainty of hydrogeological properties.

Chapter 1 addressed previous studies associated with CO_2 leakage detections. In this chapter, numerical approaches for leakage estimations were reviewed regardless if

they were forward or inverse modeling. This chapter also described other recent methods for the risk assessment of CO_2 leaks. In addition, the objective and conceptual framework of this dissertation were introduced.

In Chapter 2, the basic theory of multiphase CO_2 flow and the TOUGH2 and iTOUGH2 models were addressed. The basic theory of multiphase flow was described based on Chen et al. (2006). TOUGH2 program for the simultaneous flow analysis of two fluid phases in a porous medium was presented. The overview of the ECO2N module was also described in this chapter. The methodology, modeling procedure and functions of iTOUGH2 were introduced. In addition, the optimization methods to minimize objective function were investigated. In particular, I specifically described the algorithm and minimizing procedure of the Levenberg-Marquardt method, used for inversion of multiphase flow in this study.

Chapter 3 addressed sensitivity analysis to pressure difference (dP) induced by brine or CO_2 leaks and forward simulations in homogeneous and heterogeneous model domains with the leakage pathway. The sensitivity analysis identified the effect of hydrogeological properties on the pressure signals at monitoring wells, and forward simulations were performed to realize brine or CO_2.

The sensitivity of measurements in terms of the hydrogeological properties substantially influences the accuracy of the inverse solutions. Thus, the sensitivity analysis was examined in terms of pressure perturbations due to brine/CO_2 leaks to increase detectability of leakage pathways by inverse analysis. The sensitivity analysis focused on effects of three parameters of the overlying formation: (1) the permeability of the leakage pathway, (2) the permeability of the overlying formation, and (3) the

thickness of the cap rock. The results of the sensitivity analysis to the three parameters in terms of dP in the overlying formation were as follows:

(1) The lower permeability (e.g., 10^{-15} m^2 or lower) of the overlying formation increases dP and thus can increase effectiveness of leakage detection through inverse simulation.

(2) Thicker caprock can reduce diffuse leakage and thus magnify pressure anomalies due to leakage pathways. If the overlying formation is of a higher permeability ($k = 10^{-13}$ m^2), the cap rock thickness should probably be at least over 100 m.

(3) Leakage pathway permeability higher than at least 10^{-17} m^2 induces significant pressure anomalies through leakage pathways in the system.

The effects of migrations of brine/CO_2 through the leakage pathway were examined in the homogeneous model domain. The simulation had an injection condition of 20 million tons of CO_2 over 10 years into the storage formation. The injected CO_2 increased the saturation of gaseous CO_2, and the pressure gradient. The increased pressure gradient by CO_2 injection continuously induced brine leaks through the leakage pathway. CO_2 reached the bottom of the leakage pathway after approximately 3.5 years, and CO_2 rapidly migrated into the overlying formation by pressurization and buoyancy effects. Capillary effects were induced by migrations of CO_2 along the leakage pathway. Capillary effects reduced the brine/CO_2 leakage rates at the leakage pathway and pressure at the top of the leakage pathway was suddenly dropped by capillary effects. The effects were propagated into the overlying formation. This had a significant effect on MW5 in the overlying aquifer, the closest of all monitoring wells to the leakage pathway.

The heterogeneous domain was applied to simulate migrations of brine/CO_2 leaks and pressure anomalies induced by the leaks. The heterogeneous model domain was introduced from the SACROC unit. In the modeling scenario, the total amount of CO_2 injection was about 2 million tons over 10 years at one injection well. Only brine leakage lasted through the leakage pathway until the end of the simulation. Thus the sudden change of pressure by capillary effects was not induced in the overlying formation. The heterogeneous simulation exhibited various pressure distributions and pressure anomalies in the overlying formation.

In Chapter 4, iTOUGH2 was applied to estimate the leakage pathway location in the homogeneous and heterogeneous model domains. The leakage pathway was estimated by calibrating the vertical permeability values of initial guesses of the leakage pathway. In the homogeneous condition, the inversion was performed with three scenarios. The first scenario investigated the applicability of inverse analysis for leakage detection. The second scenario identified that uncertain permeability of the overlying formation can reduce inverse modeling accuracy for the leakage pathway estimation. The last simulation showed that the accuracy of the leakage pathway estimation can be improved by the parameterization of uncertain permeability in the overlying formation. From residual analysis, it was determined that pressure anomalies in the overlying formation induced by brine/CO_2 leaks are critical to estimate the possible leakage pathway. In addition, weighting factors were also one of the important factors for successful inverse results.

In the heterogeneous model, the inverse analysis was conducted with the approach of general modeling of heterogeneity. In the general modeling approach, approximated

average permeability values were assigned to each discrete cell and the cells with similar permeability values were grouped. For this process, the upscaling method was applied to simplify parameterization of permeability and to reduce the number of grid blocks in the heterogeneous domain. The inversion in heterogeneity was performed with two scenarios. The first inversion estimated only the vertical permeability of initial guesses based on the systematic error from renormalized permeability. Second, eight groups of renormalized permeability in the overlying and storage formations were parameterized in the inversion to reduce the impact of systematic error from the upscaling method. In conclusion, the calibration of renormalized permeability values could reduce systematic modeling errors, and improve the accuracy of the estimation of leakage pathway location. In addition, the results of inverse modeling identified that reasonable weighting coefficients are significantly important for well-posed inversion.

Chapter 5 addressed the developed forward and inverse models for single-phase flow analysis using FDM. The forward FDM model simultaneously calculates leakage rates based on leakage pathway properties between coupled leakage points at two formations. The forward FDM model does not need to mesh leakage pathway properties. In the inverse FDM model, cross-sectional area and the vertical hydraulic conductivity of the leakage pathway were integrated as one parameter. Therefore, the inverse FDM model estimates the possible leakage pathway locations using the integrated parameters of initial guesses. The parameterization of leakage pathway properties could be very effective for inverse modeling as follows:

(1) The number of grid blocks is reduced, so computational expenses can be saved.

(2) Estimating the integrated parameter can automatically decrease the impact on

uncertainty of leakage pathway size.

(3) The integrated parameter diminishes the number of parameters to be estimated.

The validation of the developed forward model was examined by comparison with TOUGH2. The relative errors between both models were less than 0.0135% at leakage and no leakage conditions.

The inverse FDM model was applied to the leakage pathway estimation using pressure anomalies induced by only brine leakage in the brine/CO_2 system. The inversion used pressure profiles in the overlying formation during the period of brine leakage. The leakage pathway was calibrated from three kinds of parameters: (1) each integrated parameter of the 48 initial guesses of leakage pathway, (2) hydraulic conductivity of the overlying formation, and (3) volumetric injection rate of water. The integrated parameters of 48 initial guesses were estimated for the leakage pathway detection. The hydraulic conductivity of the overlying formation was calibrated to reduce the uncertainty of brine density and viscosity. The volumetric injection rate was estimated to minimize errors in hydraulic gradients between the overlying and the storage formations. The result of inversion identified the applicability of using a single-phase model in a multiphase system. This will provide three advantages as follows:

(1) Inverse analysis does not need to take into account errors in capillary pressure and relative permeability functions in a multiphase flow system.

(2) The computational expenses can be mitigated from the simpler logic of a single-phase model than that of a multiphase model.

(3) In terms of an early warning before CO_2 leakage, it will be useful to decrease the risk of CO_2 leaks.

6.2 Recommendations from This Study

Several limitations of the application of the inverse model are addressed in this section. These limitations should be included in future works.

(1) Inverse modeling requires prior knowledge of the geologic and hydrologic properties, particularly the fundamental permeability values and heterogeneity. If that information is not known or significantly erroneous, the number of parameters to be estimated can be enormous. The inversion may estimate many sets of optimum parameter values, or the estimated parameters may significantly deviate from true parameter values (Finsterle, 2004).

(2) The developed single-phase model was applied to the overlying formation of the brine/CO_2 system before the CO_2 leaks into an overlying formation. This study identified applicability of leakage pathway detection using the single-phase model. However, this methodology can be limited depending on the location of leakage pathways. If the leakage pathway is not too far away from an injection well or CO_2 breaks through into the overlying reservoir in the short term after CO_2 is injected, pressure anomalies induced by only brine leakage may not be sufficient to be measured at monitoring wells in the overlying formation. Sensitivity analysis must be conducted to identify the extent of the single-phase model.

(3) The effect of noises in the measurements on parameter estimation was examined by one simulation case with 0.1 % random errors, which were randomly added in measured pressures at all of the monitoring wells by ± 0.1 %. However, more case studies need to be performed in order to identify the impact on the

accuracy of inverse analysis based on measurements including various magnitudes of random noises. Weighting factors can be used to reduce residuals increased by noises. The weighting factors depending on magnitude of random noises must be carried out for more effective assignment. Moreover, the noise filtering method needs additional research in order to reduce or stabilize random errors.

(4) Measured data can include random noises, pressure gauge error and electrical noises. In laboratory or field works, measurement errors must be minimized. In terms of measurement errors, the magnitude of pressure anomalies in the overlying formation will be closely related to the accuracy of leakage pathway estimation. If pressure anomalies are not more significant than the measurement errors, the pressure anomalies in the overlying formation cannot serve as the critical information for the leakage pathway estimation. Additional study associated with measurement errors must be conducted in the laboratory or field work.

6.3 Contributions to Science and Engineering

This dissertation includes different aspects with other studies associated with leakage detection using inverse analysis.

(1) A recent study by Jung et al. (2012b) examined the applicability of inverse analysis in the homogeneous single-phase system but their study is limited to single-phase flow. In this dissertation inverse analysis using iTOUGH2 was applied to the homogeneous and heterogeneous brine/CO_2 system including

characteristics of multiphase flow like capillary effects.

(2) Jung et al. (2012b) studied the impact of uncertainty in cap rock permeability. However, they did not implement the effect of reservoir permeability uncertainty on the leakage pathway estimation. This dissertation examined the impact of the homogeneous and heterogeneous permeability values. The inverse analysis focused on reducing its impact to improve the accuracy of leakage pathway estimation.

(3) In this dissertation a numerical model was developed for a single-phase flow system using a leakage term. Jung et al. (2012b) estimated the leakage pathway using initial guesses characterized by a mesh. The developed model parameterizes the properties of the leakage pathway using coupled leak points. The developed FDM model can simulate without a mesh for characterizing the leakage pathways, so computational expenses can be reduced.

(4) Jung et al. (2012b) did not take into account the impact of uncertainty of the leakage pathway size. The size of initial guesses, Jung et al. (2012b) characterized to estimate the leakage pathway, were identical with that of the leakage pathway they assigned in the model domain. In the developed inverse model, one parameter integrating cross-sectional area and the vertical hydraulic conductivity of initial guesses was used to estimate the leakage pathway location. Therefore, the inversion could automatically decrease the impact of uncertainty of the leakage pathway size on the leakage pathway estimation.

(5) This dissertation identified that the developed single-phase model can be applied to the leakage pathway estimation based on only brine leakage in a

brine/CO_2 system. This result can serve as an example or template to develop a system for early warning of actual CO_2 leaks.

6.4 Recommendations for Future Work

This section summarizes possible future studies based on the results of inverse analysis applied to estimate possible leakage pathway location.

(1) Multiple leakage pathways: This study examined the applicability of inverse analysis to estimate one leakage pathway in the generic homogeneous and heterogeneous domains. Multiple leakage pathways should be pursued to evaluate detectability of inverse method. Pressure anomalies may be superposed by brine/CO_2 leaks from the multiple leakage pathways, so those can be distributed creating further complexity. It will be a difficult problem and challenge. Furthermore, sensitivity of monitoring wells should be performed in terms of the multiple leakage pathways.

(2) Number and location of monitoring wells: In this study, nine or five pressure observation points were used for measurements depending on conditions of parameter estimation. The number of measurements can influence the accuracy of inversion. In addition, the location of monitoring wells has an effect on sensitivity of measurements in terms of location of the leakage pathways. In future study, the number and location of monitoring wells should be quantitatively examined.

(3) Uncertainties of multiple properties: Uncertainties of permeability values and leakage pathway sizes had an impact on the accuracy of leakage pathway estimation. The uncertainties were parameterized and calibrated with 0.1%

random errors to improve accuracy of inversion. However, more various errors should be introduced into the inverse modeling to examine their impact and to identify limitations on inversion.

(4) Upgrade of FDM model: The applicability of a developed single-phase model to multiphase system was identified. However, a weakness in the developed forward simulator is a linear matrix solver. The model uses the Gauss-Seidel iterative. This method is an old scheme and spends too much time on solving a large number of grid blocks in system. For a more effective model, the solver has to be replaced with modern methods like the sparse matrix method, which is a robust direct method. Moreover, the developed inverse model utilizes a genetic algorithm (GA) to minimize objective function. The GA belongs to the direct search method. The direct search method does not involve derivatives in an objective function. Instead, the calculated outputs are assigned directly to the objective function, so the method requires lots of simulation time. In general, the descent techniques (like Levenberg-Marquardt method and Gauss-Newton method) are more efficient than the direct search methods. Therefore, the FDM model should introduce the descent method for more effective optimization process, or the FDM model can be combined with iTOUGH2-PEST module, a universal optimization code (Doherty et al., 1994; Doherty, 2007).

(5) Geostatistical approach and pilot point method: The inversion in a heterogeneous field was progressed to improve accuracy of leakage pathway estimation through calibrating renormalized permeability based on known heterogeneity. On the other hand, as mentioned in section 4.2.2, the pilot point

method incorporated with inverse modeling and geostatistics had been applied to estimate heterogeneity (Kowalsky et al., 2004). Those methods can be applied for leakage pathway estimation in uncertain heterogeneous domain.

6.5 References

Anderson, M. P., and Woessner, W. W. (1992). *Applied groundwater modeling: simulation of flow and advective transport*, Elsevier Science, Oxford, UK.

Chen, Z., Huan, G., and Ma, Y. (2006). *Computational methods for multiphase flows in porous media*, Siam, Philadelphia, PA.

Cihan, A., Zhou, Q., and Birkholzer, J. T. (2011). "Analytical solutions for pressure perturbation and fluid leakage through aquitards and wells in multilayered-aquifer systems." *Water Resources Research*, 47(10), W10504.

Collins, R. E. (1961). *Flow of fluids through porous materials*, Reinhold Publishing Corp., New York, NY.

Doherty, J. (2007). "FORTRAN 90 Modules for Implementation of Parallelised, Model-Independent, Model-Based Processing." *Watermark Numerical Computing, Australia*.

Doherty, J., Brebber, L., and Whyte, P. (1994). "PEST: Model-independent parameter estimation." *Watermark Computing, Corinda, Australia*, 122.

Doughty, C., and Pruess, K. (2004). "Modeling supercritical carbon dioxide injection in heterogeneous porous media." *Vadose Zone Journal*, 3(3), 837-847.

Espinoza, F. P., Minsker, B. S., and Goldberg, D. E. (2005). "Adaptive hybrid genetic algorithm for groundwater remediation design." *Journal of Water Resources Planning and Management*, 131(1), 14-24.

Finsterle, S. (2004). "Multiphase inverse modeling." *Vadose Zone Journal*, 3(3), 747-762.

Finsterle, S. (2007a). "iTOUGH2 user's guide." *Report LBNL-40040*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007b). "iTOUGH2 command reference." *Report LBNL-40041*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2007c). "iTOUGH2 sample problems." *Report LBNL-40042*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S. (2010). "iTOUGH2 V3. 2, verification and validation report." *Report LBNL-42002*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Finsterle, S., Moridis, G., and Pruess, K. (1994). "A tough2 equation-of-state module for the simulation of two-phase flow of air, water, and a miscible gelling liquid." *Report LBL-36086*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Gasda, S. E., Bachu, S., and Celia, M. A. (2004). "Spatial characterization of the location of potentially leaky wells penetrating a deep saline aquifer in a mature sedimentary basin." *Environmental Geology*, 46(6-7), 707-720.

Gasda, S. E., Wang, J. Z., and Celia, M. A. (2011). "Analysis of in-situ wellbore integrity data for existing wells with long-term exposure to CO2." *Energy Procedia*, 4, 5406-5413.

Han, W. S., McPherson, B. J., Lichtner, P. C., and Wang, F. P. (2010). "Evaluation of trapping mechanisms in geologic CO2 sequestration: Case study of SACROC northern platform, a 35-year CO2 injection site." *American Journal of Science*, 310(4), 282-324.

Harbaugh, A. W. (2005). *MODFLOW-2005, the US Geological Survey modular ground-water model: The ground-water flow process*, US Department of the Interior, US Geological Survey.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012a). "Early detection of brine or CO2 leakage through high-permeability pathways using pressure-based monitoring data." *The Eleventh Annual Carbon Capture, Utilization & Sequestration Conference,* Pittsburgh, PA.

Jung, Y., Zhou, Q., and Birkholzer, J. T. (2012b). "Impact of data uncertainty on identifying leakage pathways in CO2 geologic storage systems and estimating their hydrogeological properties by inverse modeling." *TOUGH Symposium 2012,* Lawrence Berkeley National Laboratory, Berkeley, CA.

King, P. (1989). "The use of renormalization for calculating effective permeability." *Transport in Porous Media*, 4(1), 37-58.

Kowalsky, M. B., Finsterle, S., and Rubin, Y. (2004). "Estimating flow parameter distributions using ground-penetrating radar and hydrological measurements during transient flow in the vadose zone." *Advances in Water Resources*, 27(6), 583-599.

Liggett, J. A., and Chen, L.-C. (1994). "Inverse transient analysis in pipe networks." *Journal of Hydraulic Engineering*, 120(8), 934-955.

Metz, B., Davidson, O., de Coninck, H., Loos, M., and Meyer, L. (2005). *IPCC special report on carbon dioxide capture and storage*, Cambridge University Press, New York, NY.

Nogues, J. P., Nordbotten, J. M., and Celia, M. A. (2011). "Detecting leakage of brine or CO2 through abandoned wells in a geological sequestration operation using pressure monitoring wells." *Energy Procedia*, 4, 3620-3627.

Nordbotten, J. M., Celia, M. A., and Bachu, S. (2004). "Analytical solutions for leakage rates through abandoned wells." *Water Resources Research*, 40(4), W04204.

Nordbotten, J. M., Kavetski, D., Celia, M. A., and Bachu, S. (2008). "Model for CO2 leakage including multiple geological layers and multiple leaky wells." *Environmental Science & Technology*, 43(3), 743-749.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in Fortran 77: The art of scientific computing second edition*, Cambridge University Press, New York, NY.

Pruess, K. (1987). "TOUGH user's guide, nuclear regulatory commission report NUREG/CR-4645." *Report LBL-20700*, Lawrence Berkeley Laboratory Berkeley, CA.

Pruess, K. (1991). "TOUGH2: A general-purpose numerical simulator for multiphase fluid and heat flow." *Report LBL-29400*, Lawrence Berkeley Laboratory, Berkeley, CA.

Pruess, K. (2004). "Numerical simulation of CO2 leakage from a geologic disposal reservoir, including transitions from super-to subcritical conditions, and boiling of liquid CO2." *Spe Journal*, 9(2), 237-248.

Pruess, K. (2005). "ECO2N: A TOUGH2 fluid property module for mixtures of water, NaCl, and CO2." *Report LBNL-57952*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Pruess, K., and García, J. (2002). "Multiphase flow dynamics during CO2 disposal into saline aquifers." *Environmental Geology*, 42(2-3), 282-295.

Pruess, K., Moridis, G., and Oldenburg, C. (1999). "TOUGH2 user's guide, version 2.0." *Report LBNL-43134*, Lawrence Berkeley National Laboratory, Berkeley, CA.

Rao, S. S. (2009). *Engineering optimization: Theory and practice*, John Wiley & Sons, Hoboken, New Jersey.

Singh, S. K. (2009). "Simple method for quick estimation of leaky-aquifer parameters." *Journal of Irrigation and Drainage Engineering*, 136(2), 149-153.

Van Genuchten, M. T. (1980). "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils." *Soil Science Society of America Journal*, 44(5), 892-898.

Zhou, Q., Birkholzer, J. T., and Tsang, C.-F. (2009). "A semi-analytical solution for large-scale injection-induced pressure perturbation and leakage in a laterally bounded aquifer–aquitard system." *Transport in Porous Media*, 78(1), 127-148.

APPENDIX A

EFFECT ON UNCERTAINTY OF PERMEABILITY IN THE STORAGE FORMATION

This inversion identifies the effect of permeability uncertainty in the storage formation. This simulation was applied to the "idealized case" discussed in section 4.1 with incorrect permeability ($k = 10^{-12.5}$ m^2; true: $k = 10^{-13}$ m^2) in the storage formation. Fig. A.1 illustrates the simulation result from objective function. The estimated leaky well location is (x, y) = (5,150 m, 5,950 m). The deviation between true and estimated leaky well is 141 m.

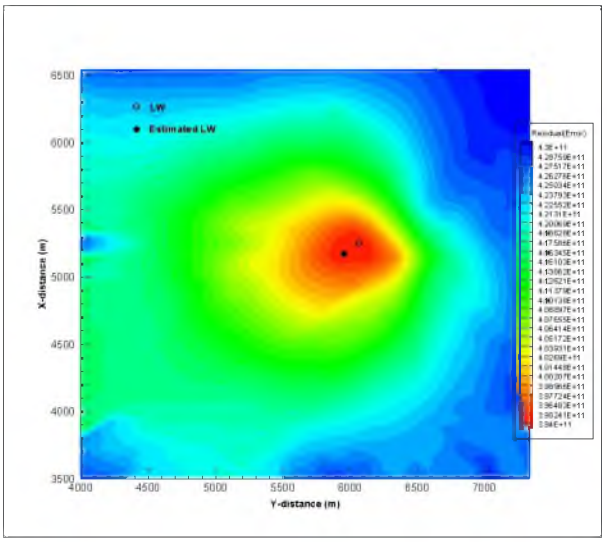


Fig. A.1 Estimated leakage well location based on the overestimated permeability in the storage formation.

APPENDIX B

RESIDUALS OF THE FIRST ADDITIONAL INVERSION

In section 4.1.2, the first additional inversion identified the impact of uncertainty in the leakage pathway size on leakage pathway estimation. Fig. B.1 illustrates residuals between measured and calculated pressures in the storage formation of the best estimation (see Fig. 4.11). Fig. B.2 presents residuals for the overlying formation of the estimated leakage pathway location. The residuals for the overlying formation in Fig. B.2 are larger than those for case 1 (a) and case 3 (d) in Fig. 4.8. This inversion has lower accuracy.

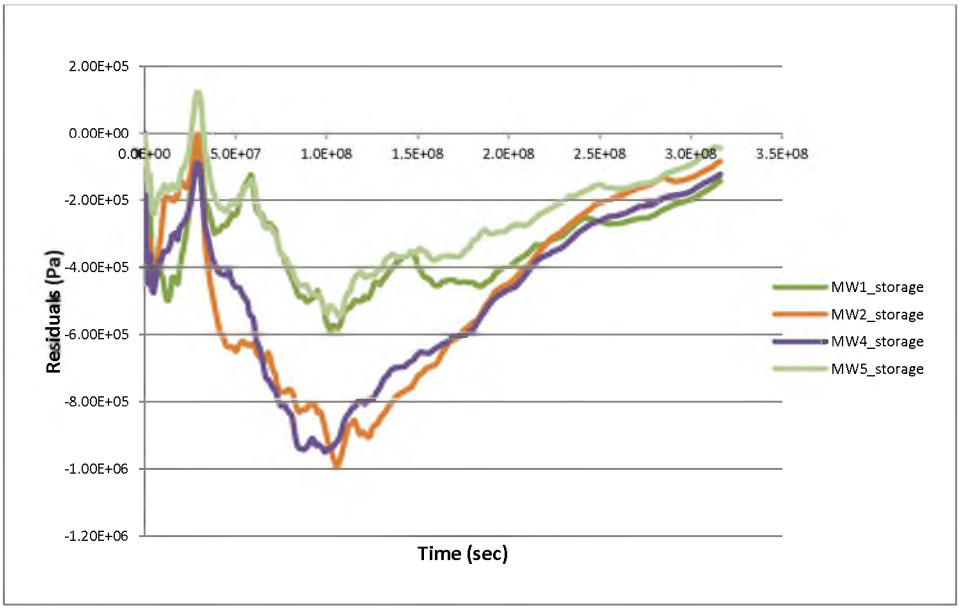


Fig. B.1 Residuals in the storage formation in the first additional inversion (effect of uncertainty of the leakage pathway size) of homogeneous model.

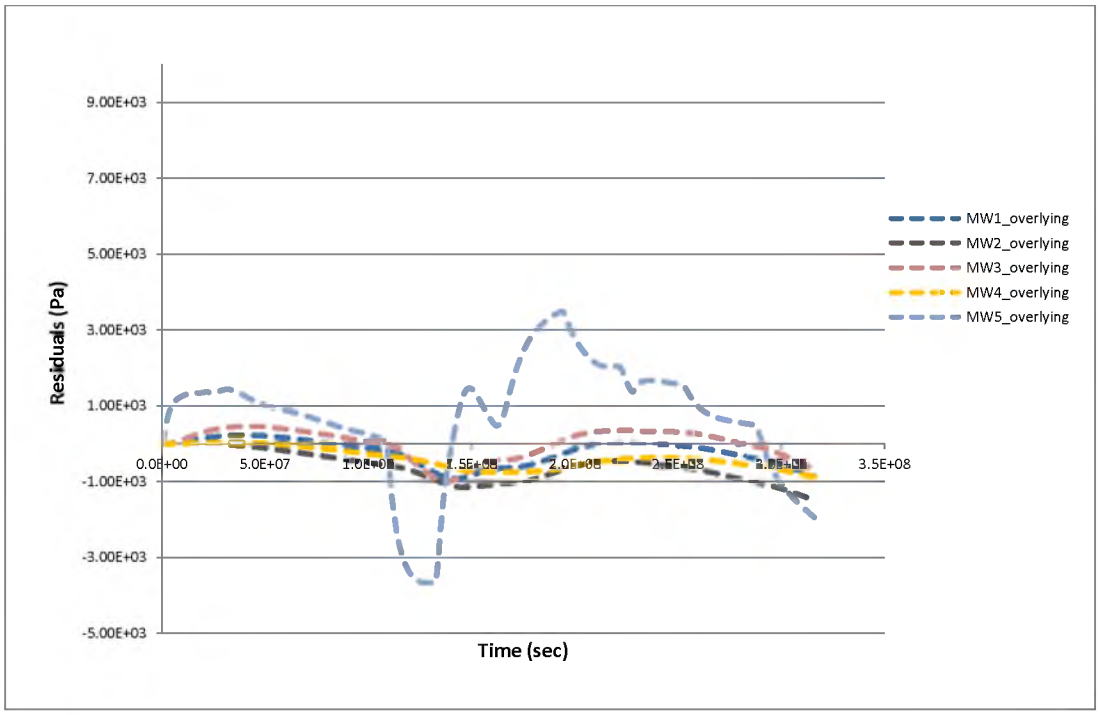


Fig. B.2 Residuals in the overlying formation in the first additional inversion (effect of uncertainty of the leakage pathway size) of homogeneous model.

APPENDIX C

RESIDUALS OF THE SECOND ADDITIONAL INVERSION

The effect of noises in the measurements on parameter estimation was examined by one simulation case with random noises of 0.1 % (the second additional inversion) in section 4.1.2. Fig. C.1 illustrates residuals for the storage formation of the leakage pathway location estimated by that inversion (see Fig. 4.14). Fig. C.2 presents residuals for the overlying formation. Fig. C.3 represents residuals at MW5 in the overlying formation. The measurements with random noises cause the fluctuation of residuals in the overlying and storage formations.

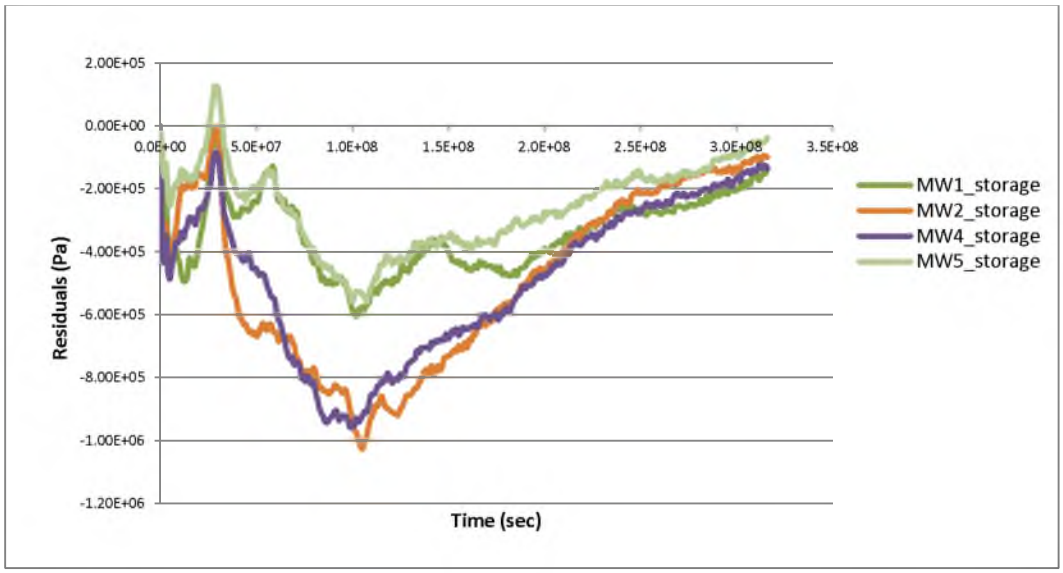


Fig. C.1 Residuals in the storage formation in the inversion for measurement noises.

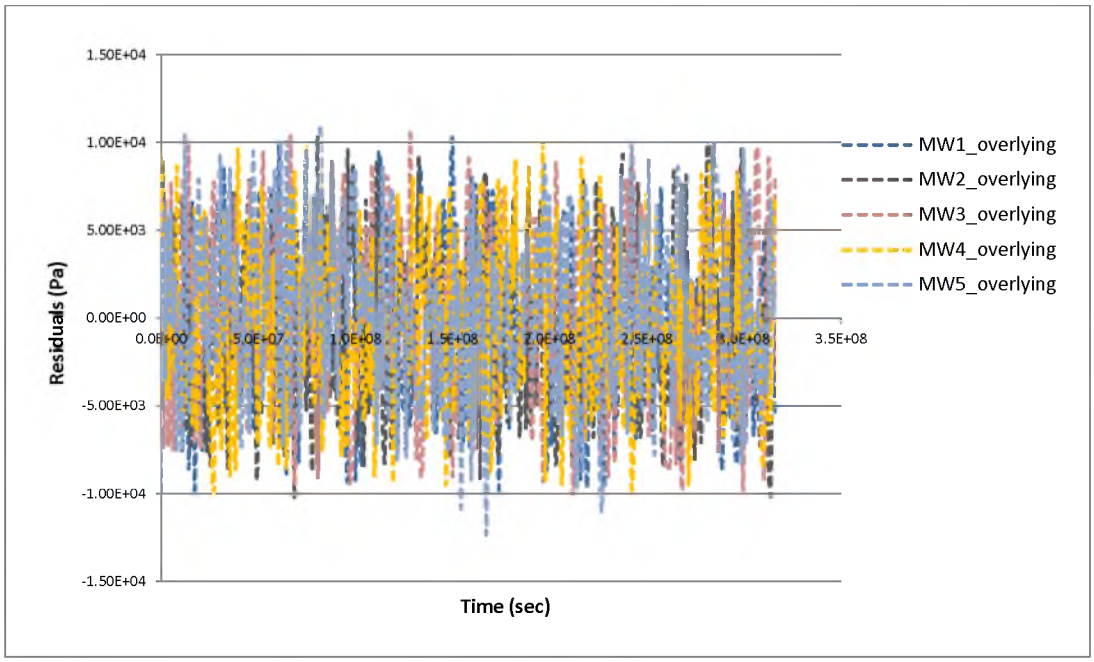


Fig. C.2 Residuals in the overlying formation in the inversion for measurement noises.

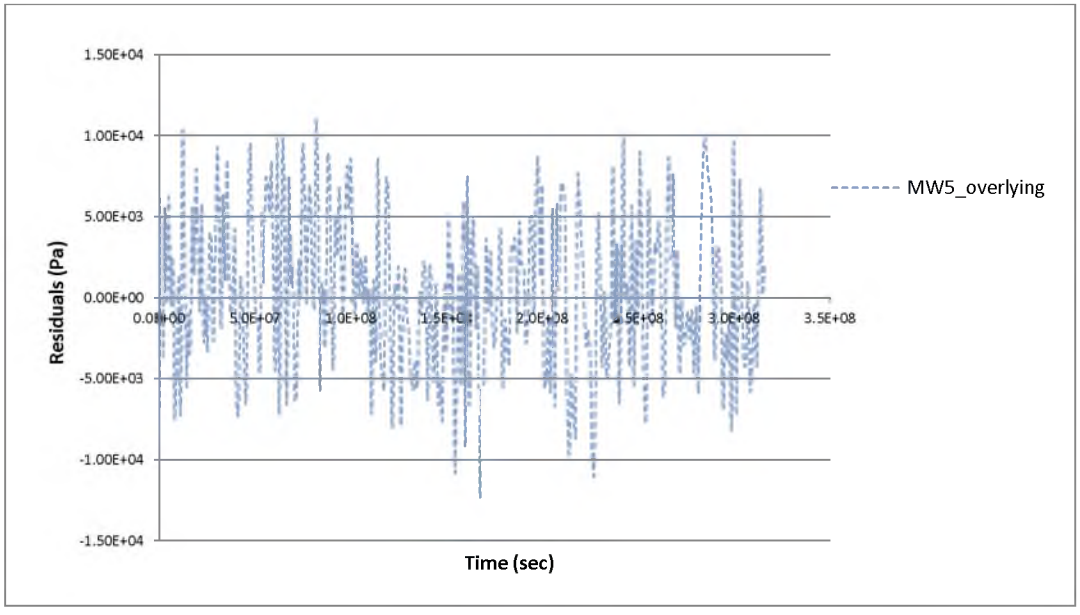


Fig. C.3 Residuals at MW5 in the overlying formation in the inversion for measurement noises.

APPENDIX D

COMPARING PRESSURE DISTRIBUTIONS BETWEEN ITOUGH2 AND DEVELOPED FDM MODEL

The new FDM model was applied to the leakage pathway estimation in the brine/CO_2 system in Chapter 5. The new FDM model uses constant hydraulic conductivities characterized by the average density and viscosity calculated from TOUGH2. Uncertain density and viscosity might cause under- or overestimated hydraulic conductivity values, inducing errors in simulation outputs of the new FDM model. Thus, the effect of uncertain hydraulic conductivity was examined by comparison of pressure distributions at the monitoring wells of the overlying formation between the developed model and TOUGH2. Fig. D.1 illustrates that the uncertainty of hydraulic conductivity in the model domain, resulting from uncertain density and viscosity of brine, induces errors in calculated pressure from the new FDM model.

In the simulation condition, the initial approximate hydraulic conductivity was assigned to 1.8e-3 m/s for the leakage pathway. When the hydraulic conductivity of the leakage pathway was assigned to 1.0e-4 m/s, error in pressure at the monitoring wells of the overlying formation could be reduced (Fig. D.2).

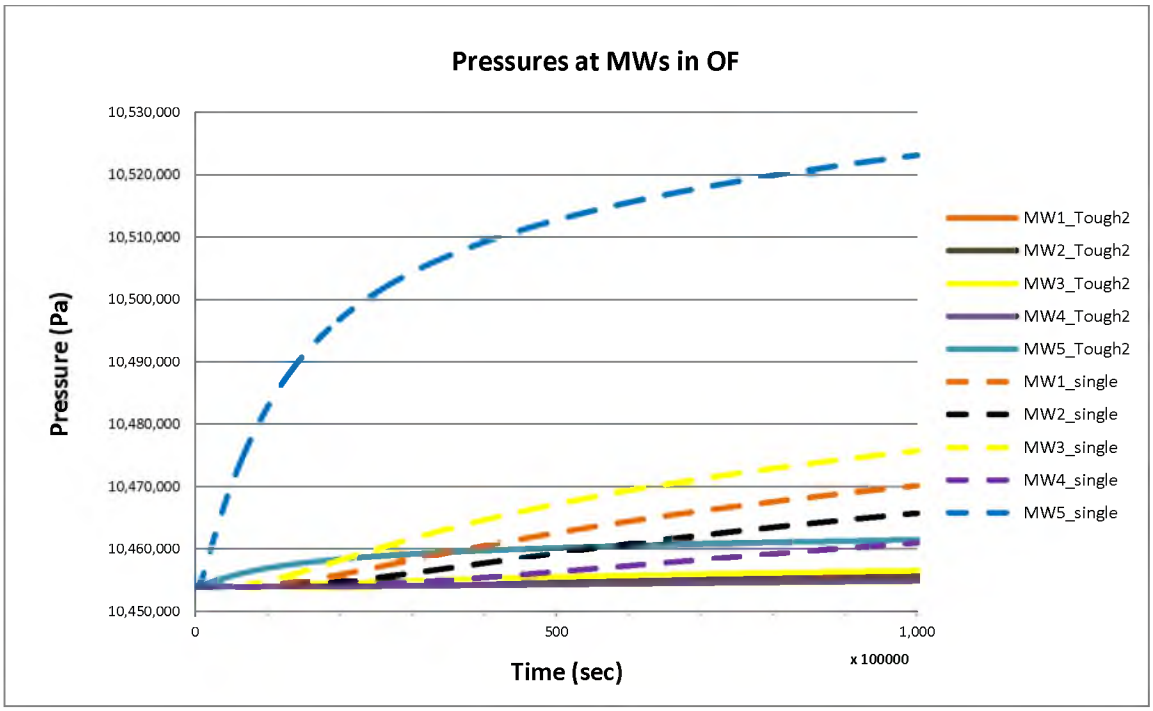


Fig. D.1 Simulated pressure distributions at monitoring wells of the overlying formation (hydraulic conductivity of the leakage pathway: 1.8e-3 m/s (KA: 1.6e-4 m^3/s).

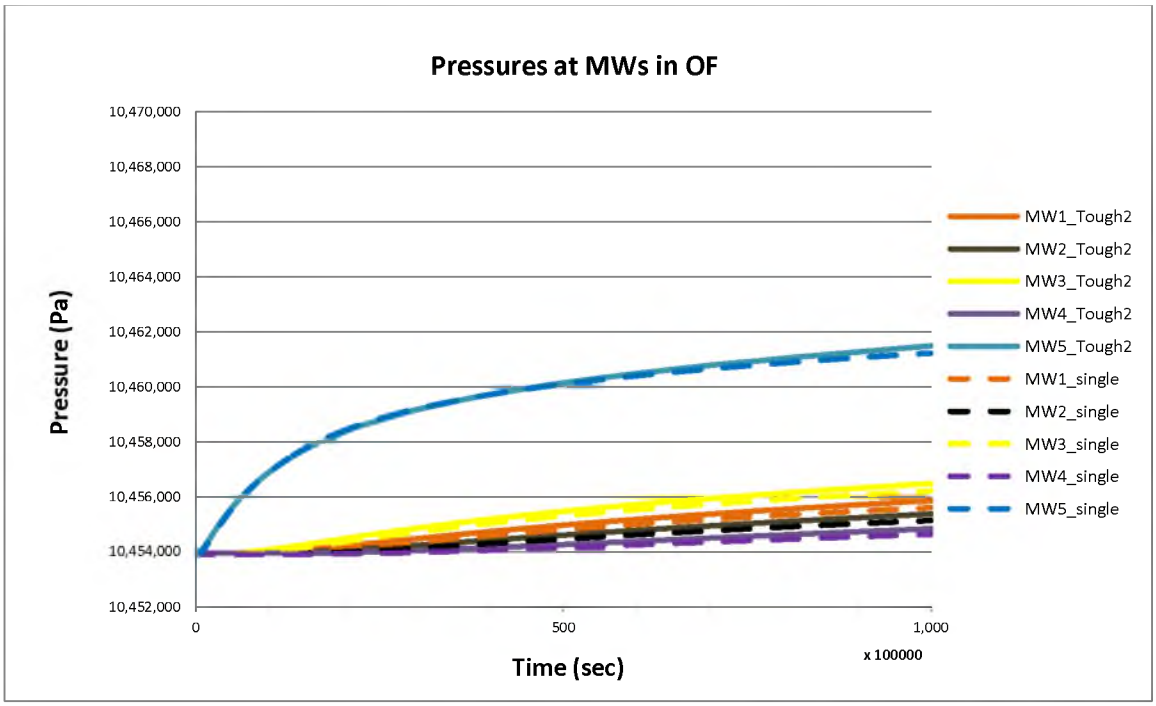


Fig. D.2 Simulated pressure distributions at monitoring wells of the overlying formation (hydraulic conductivity of the leakage pathway: 1.0e-4 m/s (KA: 0.9e-5 m^3/s).

APPENDIX E

UNCERTAINTY ANALYSIS FOR PERMEABILITY UNCERTAINTY OF THE OVERLYING FORMATION

In the second scenario of section 4.1, a simple uncertainty analysis was conducted for underestimated and overestimated permeability ($10^{-15.5}$ m^2 and $10^{-14.5}$ m^2; true: 10^{-15} m^2) of the overlying formation (see Fig. 4.3). Another uncertainty analysis was conducted with three permeability values of the overlying formation (10^{-13}, $10^{-12.5}$ and $10^{-13.5}$ m^2) as shown in Fig. E.1. In Fig. E.1 the solid, dashed and dotted lines represent pressures at MW1, MW2 and MW3 in the overlying formation, respectively.

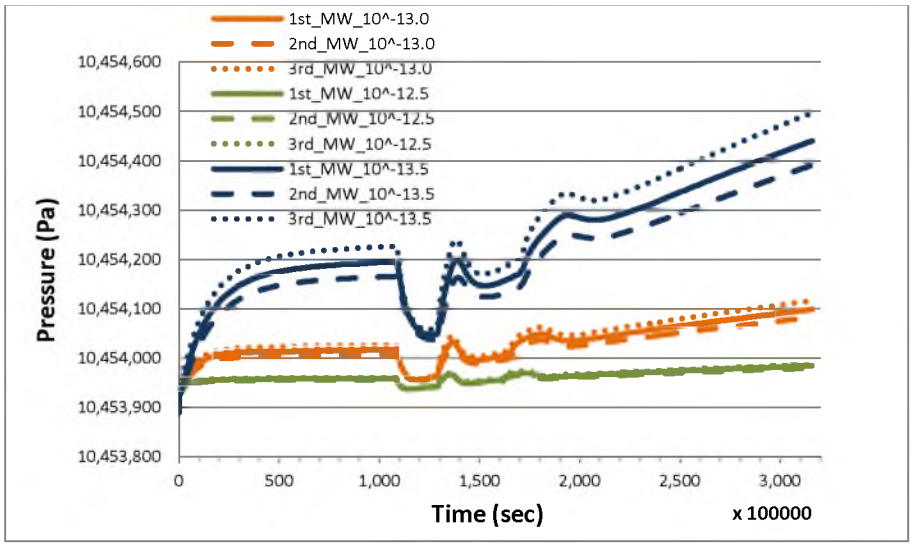


Fig. E.1 Pressure drifts among 10^{-13}, $10^{-12.5}$ and $10^{-13.5}$ m^2 permeability of the overlying formation.

APPENDIX F

COMPARISON OF PRESSURE BETWEEN REAL AND UPSCALED DOMAIN

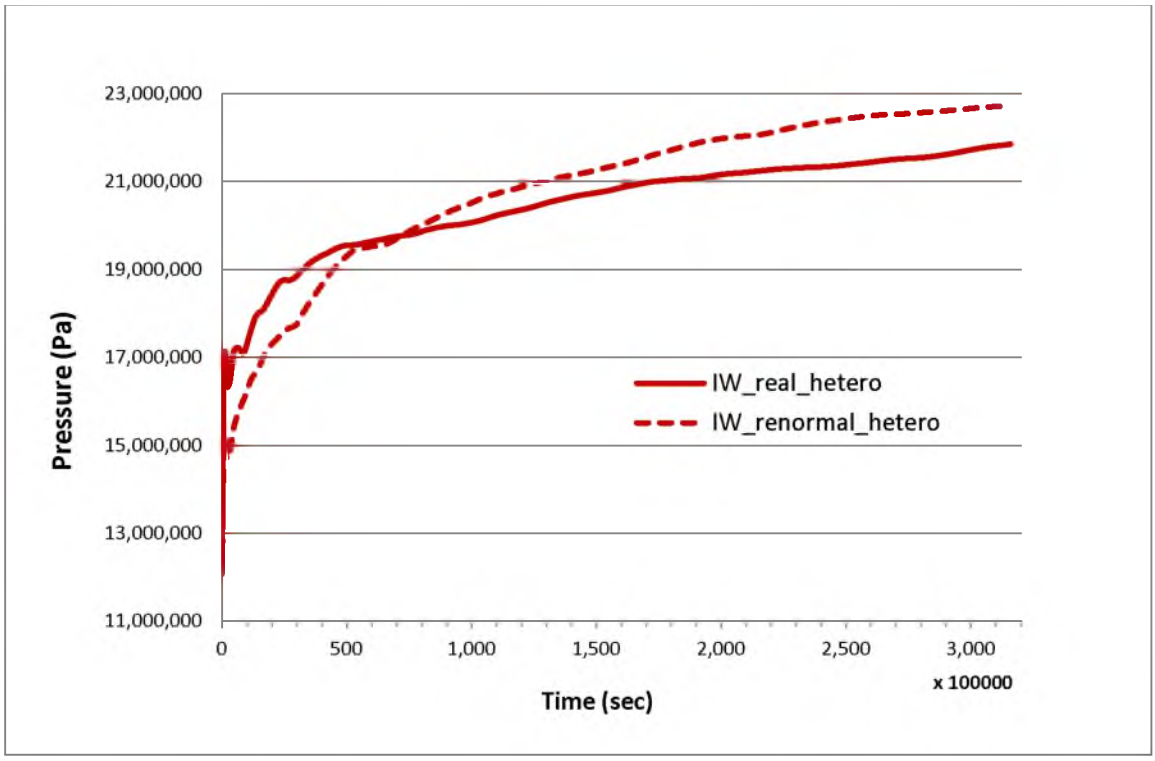


Fig. F.1 Pressure distribution at the bottom of the injection well between real and renormalized heterogeneous permeability.

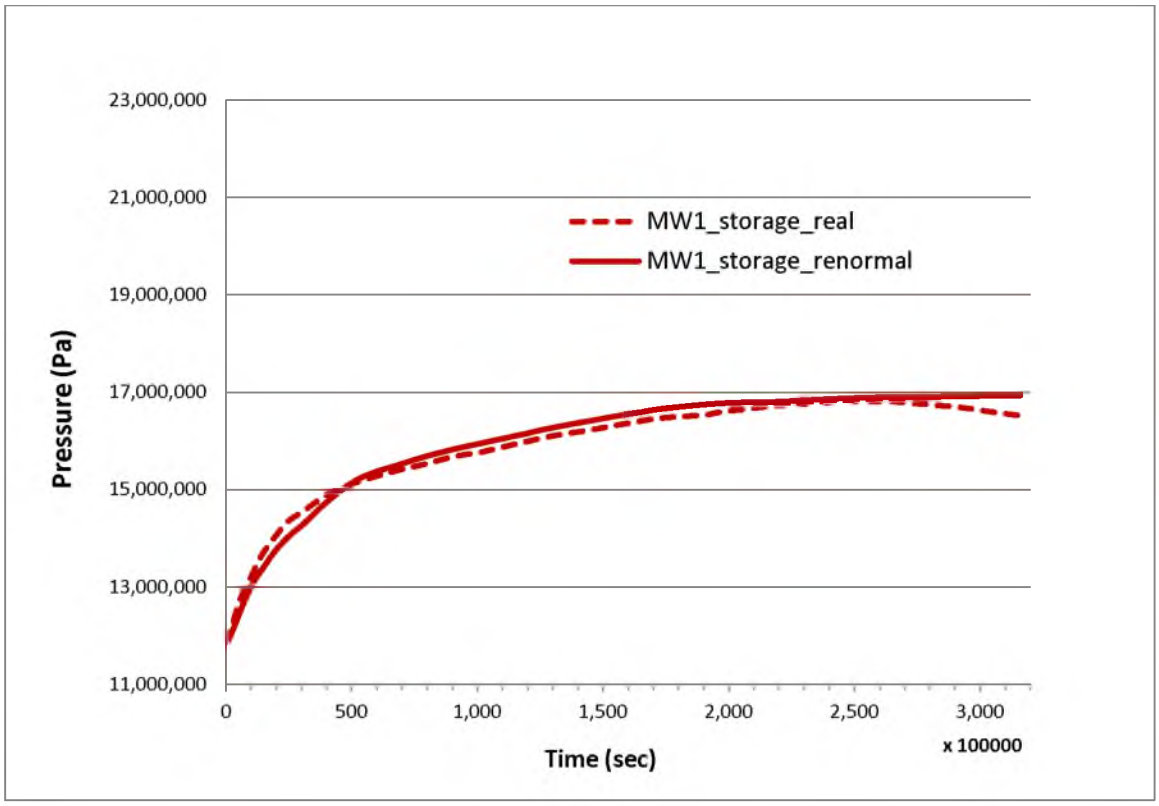


Fig. F.2 Pressure distribution at the first monitoring well of the storage formation between real and renormalized heterogeneous permeability.

APPENDIX G

ADDITIONAL SENSITIVITY ANALYSIS IN SECTION 2.3.1

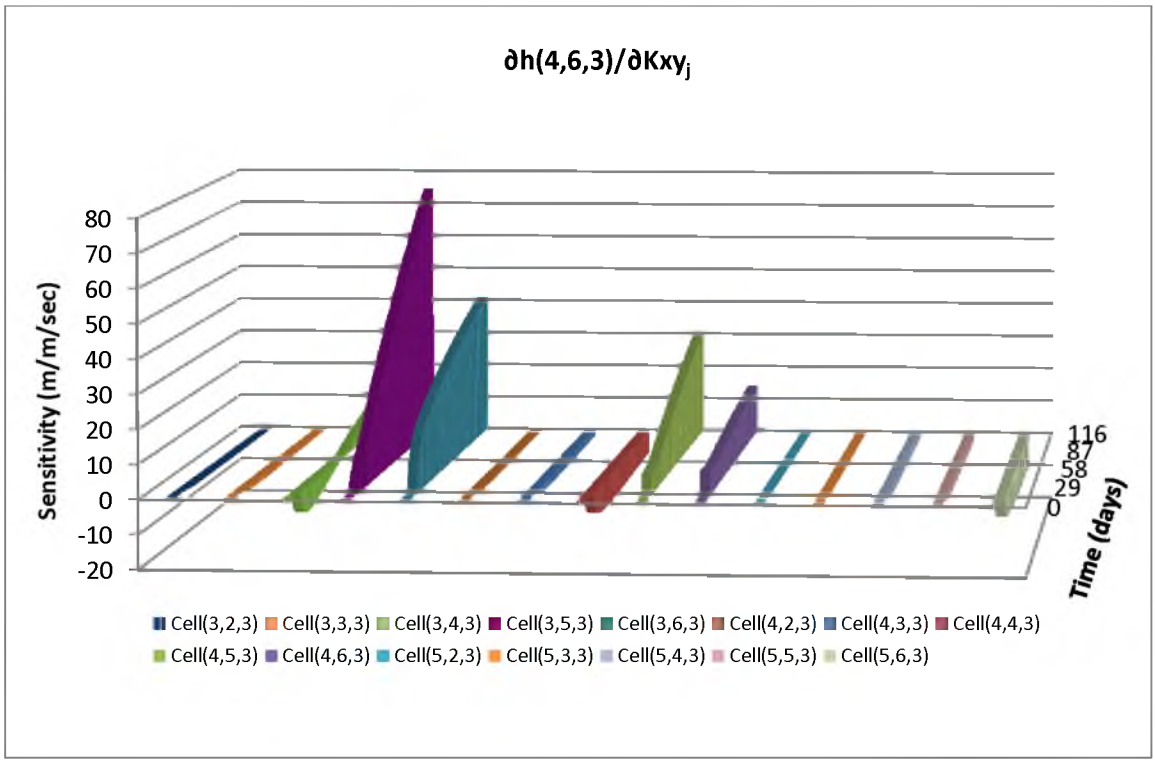


Fig. G.1 Sensitivity analysis results of the hydraulic head at cell (4, 6, 3) to lateral hydraulic conductivity change at each cell.

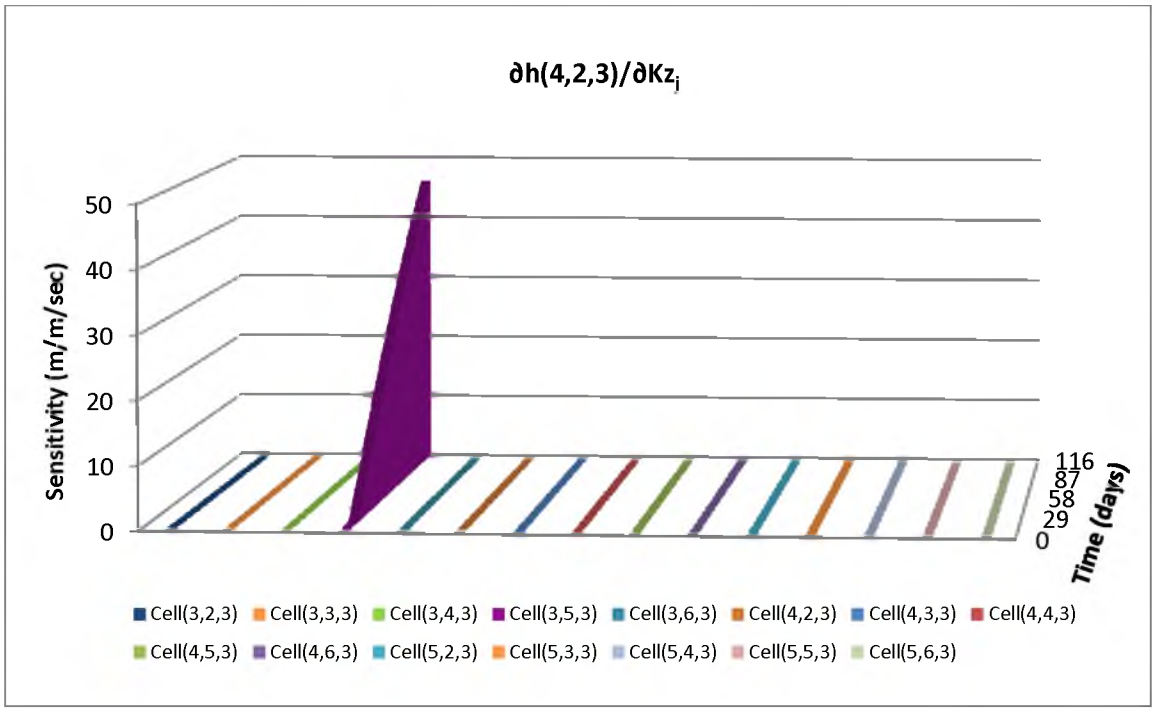


Fig. G.2 Sensitivity analysis results of the hydraulic head at cell (4, 2, 3) to vertical hydraulic conductivity change at each cell.

APPENDIX H

DEVELOPED SINGLE-PHASE SIMULATOR

```
      PROGRAM INV_GW

C  THIS PROGRAM IS COMBINED BY BOTH
ONE GA AND GROUNDWATER SIMULATOR
V2.5.
C  THIS FOCUSES ON ESTIMATION OF
VERTICAL HYDRAULIC CONDUCTIVITIES
AND
C  LEAKAGE RATE OF LEAKAGE ZONES AND
HYDRAULIC CONDUCTIVITIES OF GROUPED
C  NORMAL ZONES.
C  The inverse version 2.4.1 was
revised from v.2.4.
C  Version 2.4.1 has only one
hydraulic conductivity, i.e, x,y and
z-H.C are
C  the same. So Total kinds of
unknown parameters are just two.
C  (1) KA of leakage pathways. This
is the same as inverse version 2.4.
C  (2) x-, y- and z- Hydraulic
conductivity of each group of normal
cells


C   PROGRAM GA
C  THIS IS VERSION 1.7A, LAST
UPDATED ON 4/2/2001.
C  LAST MINOR BUG FOUND 6/14/00.
C
C  COPYRIGHT DAVID L. CARROLL; THIS
CODE MAY NOT BE REPRODUCED FOR SALE
C  OR FOR USE IN PART OF ANOTHER
CODE FOR SALE WITHOUT THE EXPRESS
C  WRITTEN PERMISSION OF DAVID L.
CARROLL.
C

      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'

      CHARACTER*40 FNAME1, FNAME2
      DIMENSION
PARENT(NPARMAX,INDMAX),CHILD(NPARMAX
,INDMAX)
      DIMENSION
FITNESS(INDMAX),NPOSIBL(NPARMAX),NIC
HFLG(NPARMAX)
      DIMENSION
IPARENT(NCHRMAX,INDMAX),ICHILD(NCHRM
AX,INDMAX)
      DIMENSION
G0(NPARMAX),G1(NPARMAX),IG2(NPARMAX)
      DIMENSION IBEST(NCHRMAX)
      DIMENSION
PARMAX(NPARMAX),PARMIN(NPARMAX),PARD
EL(NPARMAX)
      DIMENSION
GENI(1000000),GENAVG(1000000),GENMAX
(1000000)

C      REAL*4
CPU,CPU0,CPU1,TARRAY(2)
C
      DIMENSION
NXM(NXNODE),NYM(NYNODE),NZM(NZNODE)
      DIMENSION
HEADCAL(NTIMES,NNODES)
      DIMENSION
VKAOL(NPARMAX),TMLEAK(NPARMAX),HDCN(
NPARMAX)
      DIMENSION STIME(NTIMES)
      DIMENSION NPG(NNODES)
      DIMENSION WEFR(NNODES)
      INTEGER
XEL(NXNODE),YEL(NYNODE),ZEL(NZNODE),
ZELU(NZNODE)
      INTEGER
XOG(NXNODE,NEHCG),YOG(NYNODE,NEHCG),
ZOG(NZNODE,NEHCG)
      COMMON/ GWM /SITIME(NTIMES),
HEADMEA(NTIMES,NNODES)
       COMMON/GWM2/ NMD,NST,WEFR
       COMMON/GWM1/NEL,NXM,NYM,NZM
       COMMON/GWM3/NSM
       COMMON/GWM4/ELTIME
       COMMON/GWM5/XEL,YEL,ZEL,ZELU
       COMMON/GWM6/NPOPULA,NGENERA
       COMMON/GWM7/XOG,YOG,ZOG,NPG,N
TG,NOG
       COMMON/FGW/ NFI
       COMMON/GRI/GRD(NTIMES)

       COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      COMMON / GA4   / FITNESS
      COMMON / GA5   / G0,G1,IG2
      COMMON / GA6   /
PARMAX,PARMIN,PARDEL,NPOSIBL
      COMMON / GA7   / CHILD,ICHILD
      COMMON / GA8   / NICHFLG
      COMMON /INPUTGA/
PCROSS,PMUTATE,PCREEP,MAXGEN,IDUM,IR
ESTRT,
```

```
     +
ITOURNY,IELITE,ICREEP,IUNIFRM,INICHE
,
     +
ISKIP,IEND,NCHILD,MICROGA,KOUNTMX
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  INPUT VARIABLE DEFINITIONS:
C
C  ICREEP   = 0 FOR NO CREEP
MUTATIONS
C           = 1 FOR CREEP MUTATIONS;
CREEP MUTATIONS ARE RECOMMENDED.
C  IDUM     THE INITIAL RANDOM
NUMBER SEED FOR THE GA RUN.  MUST
EQUAL
C           A NEGATIVE INTEGER, E.G.
IDUM=-1000.
C  IELITE   = 0 FOR NO ELITISM (BEST
INDIVIDUAL NOT NECESSARILY
C              REPLICATED FROM ONE
GENERATION TO THE NEXT).
C           = 1 FOR ELITISM TO BE
INVOKED (BEST INDIVIDUAL REPLICATED
C              INTO NEXT
GENERATION); ELITISM IS RECOMMENDED.
C  IEND         = 0 FOR NORMAL GA
RUN (THIS IS STANDARD).
C           = NUMBER OF LAST
POPULATION MEMBER TO BE LOOKED AT IN
A SET
C             OF INDIVIDUALS.
SETTING IEND-0 IS ONLY USED FOR
DEBUGGING
C             PURPOSES AND IS
COMMONLY USED IN CONJUNCTION WITH
ISKIP.
C  INICHE   = 0 FOR NO NICHING
C           = 1 FOR NICHING; NICHING
IS RECOMMENDED.
C  IRESTRT  = 0 FOR A NEW GA RUN, OR
FOR A SINGLE FUNCTION EVALUATION
C           = 1 FOR A RESTART
CONTINUATION OF A GA RUN.
C  ISKIP    = 0 FOR NORMAL GA RUN
(THIS IS STANDARD).
C           = NUMBER IN POPULATION
TO LOOK AT A SPECIFIC INDIVIDUAL OR
C             SET OF INDIVIDUALS.
SETTING ISKIP-0 IS ONLY USED FOR
C             DEBUGGING PURPOSES.
C  ITOURNY  NO LONGER USED.  THE GA
IS PRESENTLY SET UP FOR ONLY
C           TOURNAMENT SELECTION.
C  IUNIFRM  = 0 FOR SINGLE-POINT
CROSSOVER
C           = 1 FOR UNIFORM
CROSSOVER; UNIFORM CROSSOVER IS
RECOMMENDED.
C  KOUNTMX  = THE MAXIMUM VALUE OF
KOUNT BEFORE A NEW RESTART FILE IS
C             WRITTEN; PRESENTLY SET
TO WRITE EVERY FIFTH GENERATION.
C             INCREASING THIS VALUE
WILL REDUCE I/O TIME REQUIREMENTS
C             AND REDUCE WEAR AND
TEAR ON YOUR STORAGE DEVICE
C  MAXGEN   THE MAXIMUM NUMBER OF
GENERATIONS TO RUN BY THE GA.
C           FOR A SINGLE FUNCTION
EVALUATION, SET EQUAL TO 1.
C  MICROGA  = 0 FOR NORMAL
CONVENTIONAL GA OPERATION
C           = 1 FOR MICRO-GA
OPERATION (THIS WILL AUTOMATICALLY
RESET
C             SOME OF THE OTHER
INPUT FLAGS).  I RECOMMEND USING
C             NPOPSIZ=5 WHEN
MICROGA=1.
C  NCHILD   = 1 FOR ONE CHILD PER
PAIR OF PARENTS (THIS IS WHAT I
C               TYPICALLY USE).
C           = 2 FOR TWO CHILDREN PER
PAIR OF PARENTS (2 IS MORE COMMON
C               IN GA WORK).
C  NICHFLG  = ARRAY OF 1/0 FLAGS FOR
WHETHER OR NOT NICHING OCCURS ON
C             A PARTICULAR
PARAMETER.  SET TO 0 FOR NO NICHING
ON
C             A PARAMETER, SET TO 1
FOR NICHING TO OPERATE ON PARAMETER.
C             THE DEFAULT VALUE IS
1, BUT THE IMPLEMENTATION OF NICHING
C             IS STILL CONTROLLED BY
THE FLAG INICHE.
C  NOWRITE  = 0 TO WRITE DETAILED
MUTATION AND PARAMETER ADJUSTMENTS
C           = 1 TO NOT WRITE
DETAILED MUTATION AND PARAMETER
ADJUSTMENTS
C  NPARAM   NUMBER OF PARAMETERS
(GROUPS OF BITS) OF EACH INDIVIDUAL.
C           MAKE SURE THAT NPARAM
MATCHES THE NUMBER OF VALUES IN THE
C           PARMIN, PARMAX AND
NPOSIBL INPUT ARRAYS.
```

```
C  NPOPSIZ  THE POPULATION SIZE OF A
GA RUN (TYPICALLY 100 WORKS WELL).
C           FOR A SINGLE
CALCULATION, SET EQUAL TO 1.
C  NPOSIBL  = ARRAY OF INTEGER
NUMBER OF POSSIBILITIES PER
PARAMETER.
C             FOR OPTIMAL CODE
EFFICIENCY SET NPOSIBL=2**N, I.E. 2,
4,
C             8, 16, 32, 64, ETC.
C  PARMAX   = ARRAY OF THE MAXIMUM
ALLOWED VALUES OF THE PARAMETERS
C  PARMIN   = ARRAY OF THE MINIMUM
ALLOWED VALUES OF THE PARAMETERS
C  PCREEP   THE CREEP MUTATION
PROBABILITY.  TYPICALLY SET THIS
C           =
(NCHROME/NPARAM)/NPOPSIZ.
C  PCROSS   THE CROSSOVER
PROBABILITY.  FOR SINGLE-POINT
CROSSOVER, A
C           VALUE OF 0.6 OR 0.7 IS
RECOMMENDED.  FOR UNIFORM CROSSOVER,
C           A VALUE OF 0.5 IS
SUGGESTED.
C  PMUTATE  THE JUMP MUTATION
PROBABILITY.  TYPICALLY SET =
1/NPOPSIZ.
C
C
C  FOR SINGLE FUNCTION EVALUATIONS,
SET NPOPSIZ=1, MAXGEN=1, &
IRESTRT=0.
C
C  MY FAVORITE INITIAL CHOICES OF GA
PARAMETERS ARE:
C     MICROGA=1, NPOPSIZ=5,
IUNIFRM=1, MAXGEN=200
C     MICROGA=1, NPOPSIZ=5,
IUNIFRM=0, MAXGEN=200
C  I GENERALLY GET GOOD PERFORMANCE
WITH BOTH THE UNIFORM AND SINGLE-
C  POINT CROSSOVER MICRO-GA.
C
C  FOR THOSE WISHING TO USE THE MORE
CONVENTIONAL GA TECHNIQUES,
C  MY OLD FAVORITE CHOICE OF GA
PARAMETERS WAS:
C     IUNIFRM=1, INICHE=1, IELITE=1,
ITOURNY=1, NCHILD=1
C  FOR MOST PROBLEMS I HAVE DEALT
WITH, I GET GOOD PERFORMANCE USING
C     NPOPSIZ=100, PCROSS=0.5,
PMUTATE=0.01, PCREEP=0.02, MAXGEN=26
C  OR
C     NPOPSIZ= 50, PCROSS=0.5,
PMUTATE=0.02, PCREEP=0.04, MAXGEN=51
C
C  ANY NEGATIVE INTEGER FOR IDUM
SHOULD WORK.  I TYPICALLY
ARBITRARILY
C  CHOOSE IDUM=-10000 OR -20000.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  CODE VARIABLE DEFINITIONS (THOSE
NOT DEFINED ABOVE):
C
C  BEST     = THE BEST FITNESS OF
THE GENERATION
C  CHILD    = THE FLOATING POINT
PARAMETER ARRAY OF THE CHILDREN
C  CPU      = CPU TIME OF THE
CALCULATION
C  CPU0,CPU1= CPU TIMES ASSOCIATED
WITH 'ETIME' TIMING FUNCTION
C  CREEP    = +1 OR -1, INDICATES
WHICH DIRECTION PARAMETER CREEPS
C  DELTA    = DEL/NPARAM
C  DIFFRAC  = FRACTION OF TOTAL
NUMBER OF BITS WHICH ARE DIFFERENT
C             BETWEEN THE BEST AND
THE REST OF THE MICRO-GA POPULATION.
C             POPULATION CONVERGENCE
ARBITRARILY SET AS DIFFRAC<0.05.
C  EVALS    = NUMBER OF FUNCTION
EVALUATIONS
C  FBAR     = AVERAGE FITNESS OF
POPULATION
C  FITNESS  = ARRAY OF FITNESSES OF
THE PARENTS
C  FITSUM   = SUM OF THE FITNESSES
OF THE PARENTS
C  GENAVG   = ARRAY OF AVERAGE
FITNESS VALUES FOR EACH GENERATION
C  GENI     = GENERATION ARRAY
C  GENMAX   = ARRAY OF MAXIMUM
FITNESS VALUES FOR EACH GENERATION
C  G0       = LOWER BOUND VALUES OF
THE PARAMETER ARRAY TO BE OPTIMIZED.
C             THE NUMBER OF
PARAMETERS IN THE ARRAY SHOULD MATCH
THE
C             DIMENSION SET IN THE
ABOVE PARAMETER STATEMENT.
C  G1       = THE INCREMENT BY WHICH
THE PARAMETER ARRAY IS INCREASED
```

```
C             FROM THE LOWER BOUND
VALUES IN THE G0 ARRAY.  THE MINIMUM
C             PARAMETER VALUE IS G0
AND THE MAXIMUM PARAMETER VALUE
C             EQUALS G0+G1*(2**G2-
1), I.E. G1 IS THE INCREMENTAL VALUE
C             BETWEEN MIN AND MAX.
C  IG2      = ARRAY OF THE NUMBER OF
BITS PER PARAMETER, I.E. THE NUMBER
C             OF POSSIBLE VALUES PER
PARAMETER.  FOR EXAMPLE, IG2=2 IS
C             EQUIVALENT TO 4
(=2**2) POSSIBILITIES, IG2=4 IS
EQUIVALENT
C             TO 16 (=2**4)
POSSIBILITIES.
C  IG2SUM   = SUM OF THE NUMBER OF
POSSIBILITIES OF IG2 ARRAY
C  IBEST    = BINARY ARRAY OF
CHROMOSOMES OF THE BEST INDIVIDUAL
C  ICHILD   = BINARY ARRAY OF
CHROMOSOMES OF THE CHILDREN
C  ICOUNT   = COUNTER OF NUMBER OF
DIFFERENT BITS BETWEEN BEST
C             INDIVIDUAL AND OTHER
MEMBERS OF MICRO-GA POPULATION
C  ICROSS   = THE CROSSOVER POINT IN
SINGLE-POINT CROSSOVER
C  INDMAX   = MAXIMUM # OF
INDIVIDUALS ALLOWED, I.E. MAX
POPULATION SIZE
C  IPARENT  = BINARY ARRAY OF
CHROMOSOMES OF THE PARENTS
C  ISTART   = THE GENERATION TO BE
STARTED FROM
C  JBEST    = THE MEMBER IN THE
POPULATION WITH THE BEST FITNESS
C  JELITE   = A COUNTER WHICH TRACKS
THE NUMBER OF BITS OF AN INDIVIDUAL
C             WHICH MATCH THOSE OF
THE BEST INDIVIDUAL
C  JEND     = USED IN CONJUNCTION
WITH IEND FOR DEBUGGING
C  JSTART   = USED IN CONJUNCTION
WITH ISKIP FOR DEBUGGING
C  KOUNT    = A COUNTER WHICH
CONTROLS HOW FREQUENTLY THE RESTART
C             FILE IS WRITTEN
C  KELITE   = KELITE SET TO UNITY
WHEN JELITE=NCHROME, INDICATES THAT
C             THE BEST PARENT WAS
REPLICATED AMONGST THE CHILDREN
C  MATE1    = THE NUMBER OF THE
POPULATION MEMBER CHOSEN AS MATE1
C  MATE2    = THE NUMBER OF THE
POPULATION MEMBER CHOSEN AS MATE2
C  NCHRMAX  = MAXIMUM # OF
CHROMOSOMES (BINARY BITS) PER
INDIVIDUAL
C  NCHROME  = NUMBER OF CHROMOSOMES
(BINARY BITS) OF EACH INDIVIDUAL
C  NCREEP   = # OF CREEP MUTATIONS
WHICH OCCURRED DURING REPRODUCTION
C  NMUTATE  = # OF JUMP MUTATIONS
WHICH OCCURRED DURING REPRODUCTION
C  NPARMAX  = MAXIMUM # OF
PARAMETERS WHICH THE CHROMOSOMES
MAKE UP
C  PARAMAV  = THE AVERAGE OF EACH
PARAMETER IN THE POPULATION
C  PARAMSM  = THE SUM OF EACH
PARAMETER IN THE POPULATION
C  PARENT   = THE FLOATING POINT
PARAMETER ARRAY OF THE PARENTS
C  PARDEL   = ARRAY OF THE
DIFFERENCE BETWEEN PARMAX AND PARMIN
C  RAND     = THE VALUE OF THE
CURRENT RANDOM NUMBER
C  NPOSSUM  = SUM OF THE NUMBER OF
POSSIBLE VALUES OF ALL PARAMETERS
C  TARRAY   = TIME ARRAY USED WITH
'ETIME' TIMING FUNCTION
C  TIME0    = CLOCK TIME AT START OF
RUN
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  SUBROUTINES:
C  ____________
C
C  CODE     = CODES FLOATING POINT
VALUE TO BINARY STRING.
C  CROSOVR  = PERFORMS CROSSOVER
(SINGLE-POINT OR UNIFORM).
C  DECODE   = DECODES BINARY STRING
TO FLOATING POINT VALUE.
C  EVALOUT  = EVALUATES THE FITNESS
OF EACH INDIVIDUAL AND OUTPUTS
C             GENERATIONAL
INFORMATION TO THE 'GA.OUT' FILE.
C  FUNC     = THE FUNCTION WHICH IS
BEING EVALUATED.
C  GAMICRO  = IMPLEMENTS THE MICRO-
GA TECHNIQUE.
C  INPUT    = INPUTS INFORMATION
FROM THE 'GA.INP' FILE.
C  INITIAL  = PROGRAM INITIALIZATION
AND INPUTS INFORMATION FROM THE
```

```
C             'GA.RESTART' FILE.
C  MUTATE   = PERFORMS MUTATION
(JUMP AND/OR CREEP).
C  NEWGEN   = WRITES CHILD ARRAY
BACK INTO PARENT ARRAY FOR NEW
C             GENERATION; ALSO
CHECKS TO SEE IF BEST INDIVIDUAL WAS
C             REPLICATED (ELITISM).
C  NICHE    = PERFORMS NICHING
(SHARING) ON POPULATION.
C  POSSIBL  = CHECKS TO SEE IF
DECODED BINARY STRING FALLS WITHIN
C             SPECIFIED RANGE OF
PARMIN AND PARMAX.
C  RAN3     = THE RANDOM NUMBER
GENERATOR.
C  RESTART  = WRITES THE
'GA.RESTART' FILE.
C  SELECT   = A SUBROUTINE OF
'SELECTN'.
C  SELECTN  = PERFORMS SELECTION;
TOURNAMENT SELECTION IS THE ONLY
C             OPTION IN THIS VERSION
OF THE CODE.
C  SHUFFLE  = SHUFFLES THE
POPULATION RANDOMLY FOR SELECTION.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
  10  WRITE(*,'(/A)') ' WHAT KIND OF
SIMULATION DO YOU WANT?'
      WRITE(*,'(A)') ' 1. FORWARD
SIMULATION'
      WRITE(*,'(A)') ' 2. INVERSE
SIMULATION'
      READ(*,'(I2)') NFI
      IF ((NFI.NE.1).AND.(NFI.NE.2))
THEN
       WRITE (*,*) 'YOU MUST PUT 1
OR 2 IN.'
       GOTO 10
       ENDIF

      WRITE(*,'(/A)') ' WHAT IS THE
NAME OF FORWARD INPUT FILE?'
      READ(*,'(A)') FNAME1
      OPEN(4, FILE=FNAME1,
STATUS='UNKNOWN')
      REWIND 4

       IF (NFI.EQ.1) THEN
       WRITE(*,'(/A)') ' WHAT IS THE
NAME OF OUTPUT FILE FOR FORWARD
     >SIMULATION?'
      READ(*,'(A)') FNAME2
      OPEN(5, FILE=FNAME2,
STATUS='UNKNOWN')
      REWIND 5
       CALL
GW(HEADCAL,VKAOL,TMLEAK,HDCN,STIME,N
IT,ET)
      ENDIF

C      CALL ETIME(TARRAY)
C      WRITE(6,*)
TARRAY(1),TARRAY(2)
C      CPU0=TARRAY(1)
C
C  CALL THE INPUT SUBROUTINE.
C      TIME0=SECNDS(0.0)
      CALL INPUT
C
C  PERFORM NECESSARY INITIALIZATION
AND READ THE GA.RESTART FILE.
      CALL
INITIAL(ISTART,NPOSSUM,IG2SUM)
C
C  $$$$$ MAIN GENERATIONAL
PROCESSING LOOP. $$$$$
      KOUNT=0
      DO 20 I=ISTART,MAXGEN+ISTART-1
         WRITE (6,1111) I
         WRITE (24,1111) I
         WRITE(24,1050)
C
C  EVALUATE THE POPULATION, ASSIGN
FITNESS, ESTABLISH THE BEST
C  INDIVIDUAL, AND WRITE OUTPUT
INFORMATION.
         CALL
EVALOUT(ISKIP,IEND,IBEST,FBAR,BEST)
         GENI(I)=FLOAT(I)
         GENAVG(I)=FBAR
         GENMAX(I)=BEST
         IF(NPOPSIZ.EQ.1 .OR.
ISKIP.NE.0) THEN
            CLOSE(24)
            STOP
         ENDIF
C
C  IMPLEMENT "NICHING".
         IF (INICHE.NE.0) CALL NICHE
C
C  ENTER SELECTION, CROSSOVER AND
MUTATION LOOP.
         NCROSS=0
         IPICK=NPOPSIZ
         DO 45 J=1,NPOPSIZ,NCHILD
C
```

```
C  PERFORM SELECTION.
           CALL
SELECTN(IPICK,J,MATE1,MATE2)
C
C  NOW PERFORM CROSSOVER BETWEEN THE
RANDOMLY SELECTED PAIR.
           CALL
CROSOVR(NCROSS,J,MATE1,MATE2)
 45      CONTINUE
CSJ         WRITE(6,1225) NCROSS
CSJ         WRITE(24,1225) NCROSS
C
C  NOW PERFORM RANDOM MUTATIONS.  IF
RUNNING MICRO-GA, SKIP MUTATION.
         IF (MICROGA.EQ.0) CALL
MUTATE
C
C  WRITE CHILD ARRAY BACK INTO
PARENT ARRAY FOR NEW GENERATION.
CHECK
C  TO SEE IF THE BEST PARENT WAS
REPLICATED.
         CALL
NEWGEN(IELITE,NPOSSUM,IG2SUM,IBEST)
C
C  IMPLEMENT MICRO-GA IF ENABLED.
         IF (MICROGA.NE.0) CALL
GAMICRO(I,NPOSSUM,IG2SUM,IBEST)
C
C  WRITE TO RESTART FILE.
         CALL
RESTART(I,ISTART,KOUNT)
 20   CONTINUE
C  $$$$$ END OF MAIN GENERATIONAL
PROCESSING LOOP. $$$$$
C 999  CONTINUE
      WRITE(24,3000)
      DO 100 I=1,MAXGEN

EVALS=FLOAT(NPOPSIZ)*GENI(I)
         WRITE(24,3100)
GENI(I),EVALS,GENAVG(I),GENMAX(I)
 100  CONTINUE
C      CALL ETIME(TARRAY)
C      WRITE(6,*)
TARRAY(1),TARRAY(2)
C      CPU1=TARRAY(1)
C      CPU=(CPU1-CPU0)
C      WRITE(6,1400) CPU,CPU/60.0
C      WRITE(24,1400) CPU,CPU/60.0
      CLOSE (24)
C
CSJ 1050 FORMAT(1X,' #
BINARY CODE',16X,'           PARAM1
CSJ     >   PARAM2       PARAM3
PARAM4       PARAM5      FITNESS')
1050  FORMAT(1X,' #       PARAM1
PARAM2          PARAM3
     >    FITNESS')
 1111 FORMAT(/'################
GENERATION',I5,'
################')
 1225 FORMAT(/'  NUMBER OF
CROSSOVERS      =',I5)
C 1400 FORMAT(2X,'CPU TIME FOR ALL
GENERATIONS=',E12.6,' SEC'/
C     +       2X,'
',E12.6,' MIN')
 3000 FORMAT(2X//'SUMMARY OF
OUTPUT'/
     +       2X,'GENERATION
EVALUATIONS   AVG.FITNESS   BEST
FITNESS')
 3100 FORMAT(2X,3(E10.4,4X),E11.5)
C
      STOP
      END
C
C###################################
###################################
      SUBROUTINE INPUT
C
C  THIS SUBROUTINE INPUTS
INFORMATION FROM THE GA.INP
(GAFORT.IN) FILE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'

       CHARACTER*40 FNAME3, FNAME4,
FNAME5

       DIMENSION
NPOSIBL(NPARMAX),NICHFLG(NPARMAX)
      DIMENSION
PARMAX(NPARMAX),PARMIN(NPARMAX),PARD
EL(NPARMAX)

C     DIMENSION NZ(NZNODE),
NX(NXNODE), NY(NYNODE)
      DIMENSION
NXM(NXNODE),NYM(NYNODE),NZM(NZNODE)
      DIMENSION
NZ1(NZNODE),NZ2(NZNODE)
      DIMENSION NPG(NNODES)
      DIMENSION WEFR(NNODES)
```

```
      INTEGER
XEL(NXNODE),YEL(NYNODE),ZEL(NZNODE),
ZELU(NZNODE)
      INTEGER
XOG(NXNODE,NEHCG),YOG(NYNODE,NEHCG),
ZOG(NZNODE,NEHCG)
      COMMON/ GWM /SITIME(NTIMES),
HEADMEA(NTIMES,NNODES)
      COMMON/GWM2/ NMD,NST,WEFR
      COMMON/GWM1/NEL,NXM,NYM,NZM
      COMMON/GWM3/NSM
      COMMON/GWM4/ELTIME
      COMMON/GWM5/XEL,YEL,ZEL,ZELU
      COMMON/GWM6/NPOPULA,NGENERA

COMMON/GWM7/XOG,YOG,ZOG,NPG,NTG,NOG

      COMMON/GRI/GRD(NTIMES)

      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA6   /
PARMAX,PARMIN,PARDEL,NPOSIBL
      COMMON / GA8   / NICHFLG
      COMMON /INPUTGA/
PCROSS,PMUTATE,PCREEP,MAXGEN,IDUM,IR
ESTRT,
     +
ITOURNY,IELITE,ICREEP,IUNIFRM,INICHE
,
     +
ISKIP,IEND,NCHILD,MICROGA,KOUNTMX
C
      NAMELIST / GA   /
IRESTRT,NPOPSIZ,PMUTATE,MAXGEN,IDUM,
PCROSS,
     +
ITOURNY,IELITE,ICREEP,PCREEP,IUNIFRM
,INICHE,
     +
ISKIP,IEND,NCHILD,NPARAM,PARMIN,PARM
AX,NPOSIBL,
     +
NOWRITE,NICHFLG,MICROGA,KOUNTMX
C
      KOUNTMX=5
      IRESTRT=0
      ITOURNY=0
      IELITE=0
      IUNIFRM=0
      INICHE=0
      ISKIP=0
      IEND=0
      NCHILD=1
      DO 2 I=1,NPARMAX
         NICHFLG(I)=1
 2    CONTINUE
      MICROGA=0
C
CSJ   MAKING AN INPUT FILE FOR
MEASUREMENT DATA
      WRITE(*,'(/A)') ' WHAT IS THE
NAME OF INPUT FILE FOR MEASUREMENT
     >DATA?'
      READ(*,'(A)') FNAME3
      OPEN(7, FILE=FNAME3,
STATUS='UNKNOWN')
      REWIND 7
CSJ
      WRITE(*,'(/A)') ' WHAT IS THE
NAME OF OUTPUT FILE FOR GA?'
      READ(*,'(A)') FNAME4
       OPEN (UNIT=24, FILE=FNAME4,
STATUS='UNKNOWN')
      REWIND 24
CSJ      OPEN (UNIT=23,
FILE='GA.INP', STATUS='OLD')
CSJ      READ (23, NML = GA)
CSJ      CLOSE (23)
       WRITE(*,'(/A)') ' WHAT IS THE
NAME OF OUTPUT FILE FOR RESIDUALS?'
      READ(*,'(A)') FNAME5
      OPEN(5, FILE=FNAME5,
STATUS='UNKNOWN')
      REWIND 5
C V2.7.1      WRITE(5,*)'HYDRAULIC
HEADS AT EACH EXPECTED LEAKAGE
POINTS & MEASU
C V2.7.1     >REMENT POSTS, AND
LEAKAGE RATES AT EACH EXPECTED
LEAKAGE POINTS'

C -----------READ MEASUREMENT DATA
C  NEL: NUMBER OF EXPECTED LEAKAGE
PATHWAYS
C  ELTMIN: EXPECTED MINIMUM LEAKAGE
TIME, ELTMAX: EXPECTED MAXIMUM
LEAKAGE TIME
C  ELTIME: EXPECTED LEAKAGE TIME
C  XEL,YEL,NZ1: EXPECTED LEAKAGE
POINTS OF X, Y, Z-DIRECTION AT UPPER
AQUIFER
C  XEL,YEL,NZ2: EXPECTED LEAKAGE
POINTS OF X, Y, Z-DIRECTION AT
INJECTION AQUIFER
C  NOG: NUMBER OF GROUP WITH THE
SAME HYDRAULIC CONDUCTIVITY
C  NGH: SEQUENCE NUMBER OF GROUPS
```

```
C  NPG: THE NUMBER OF NODES IN EACH
GROUP WITH THE SAME HYD. CON.
C  XOG(I,J),YOG(I,J),ZOG(I,J): X,Y
AND Z-COORD. OF J-TH NODE IN I-TH
GROUP
C  NMD : THE NUMBER OF MEASUREMENT
POSTS
C  NST: TOTAL NUMBER OF TIME STEP OF
MEASUREMENT DATA (=NIT OF SUBROUTINE
GROUNDWATER)
C  NXM: MEASUREMENT NODE NUMBER OF
X-COORD., NYM: MEASUREMENT NODE
NUMBER OF Y-COORD.,
C  NZM: MEASUREMENT NODE NUMBER OF
Z-COORD.
C  SITIME(I): SIMULATION TIME AT I-
TH TIME STEP
C  HEADMEA(I,J): MEASURED HEAD DATA
AT I-TH TIME STEP AND AT J-TH
OBSERVATION POINT
C  WEFR(I): WEIGHTING FACTOR IN
OBJECTIVE FUNCTION

C  THIS VERSION DOESN'T CONSIDER
LEAKAGE STARTING TIME
C      READ(7,*) NEL, ELTMIN, ELTMAX
C
C  AFTERWARD, MODIFY THIS PART FOR
DETECTION OF LEAKAGE PATHWAYS WITH
MULTIPLE
C  INDUCED TIME

      READ(7,*) NEL, ELTIME
      READ(7,*)
(XEL(I),YEL(I),NZ1(I),
XEL(I),YEL(I),NZ2(I), I=1,NEL)
C V2.7
      READ(7,*) GMIN, GMAX
C V2.7
      NTG=O
      READ(7,*) NOG
      DO I=1,NOG
       READ(7,*) NGH, NPG(I),
(XOG(I,J),YOG(I,J),ZOG(I,J),
J=1,NPG(I))
       NTG=NTG+NPG(I)
      ENDDO

C      IF(NOG.EQ.0)THEN
C      WRITE(*,*)
C      WRITE(*,*)'NOTICE!! IF THE
NUMBER OF GROUP OF HYDRAULIC
CONDUCTIVI
C     >TIES IS ZERO, THE INFORMATION
OF GROUP OF HYDRAULIC CONDUCTIVITIES
C     > MUST BE REMOVED'
C      ENDIF

      READ(7,*) NMD, NST
      READ(7,*) (WEFR(I), I=1,NMD)
      READ(7,*) (NXM(I), NYM(I),
NZM(I), I=1,NMD)
      DO I=1,NST
       READ(7,*) SITIME(I),
(HEADMEA(I,J), J=1,NMD)
      ENDDO
      CLOSE(7)

C  MAKE NUMBER OF EACH EXPECTED
LEAKAGE POINT TO ASCENDING ORDER OF
Z-COORD.
       DO I=1,NEL
         IF(NZ1(I).LT.NZ2(I))THEN
          ZELU(I)=NZ1(I)
          ZEL(I)=NZ2(I)

ELSEIF(NZ2(I).LT.NZ1(I))THEN
          ZELU(I)=NZ2(I)
          ZEL(I)=NZ1(I)

ELSEIF(NZ1(I).EQ.NZ2(I))THEN
          WRITE(*,*)'WARNING!!
EXPECTED LEAKAGE POINTS OF UPPER AND
INJECTION
     > AQUIFERS MUST BE DIFFERENT,
CHECK MEASUREMENT DATA!!'
          STOP
         ENDIF
       ENDDO
       NSM=1 ! COUNTING NUMBER OF
INVERSE SIMULATION

CSJ END OF MAKING INPUT FILE FOR
MEASUREMENT DATA
CSJ GA INPUT DATA
      npopsiz=10
      maxgen=300
      NPOPULA=NPOPSIZ
      NGENERA=MAXGEN
      pcreep=0.5 !
=(nchrom(15)/nparam(3))/npopsiz(10)
      pmutate=0.02
C      nparam= NEL+NOG*2 ! X AND Y-
H.C ARE SAME BUT Z-H.C IS DIFFERENT
C V2.7
      nparam= NEL+NOG+1 ! X,Y AND Z-
H.C HAVE THE SAME VALUE
C V2.7
      DO I=1,NEL !VALUES FOR ONLY
K*A(VKAOL)
```

```
        nposibl(I)=2**15
        nichflg(I)=1
        parmin(I)= 1.0E-8
        parmax(I)= 1.0E-4
c        nposibl(I)=2**1
c        nichflg(I)=1
c        parmin(I)= 0.0
c        parmax(I)= 0.1
      ENDDO
C V2.7
      DO I=NEL+1,NPARAM-NOG !VALUES
FOR INJECTION RATE
        nposibl(I)=2**15
        nichflg(I)=1
        parmin(I)= GMIN
        parmax(I)= GMAX
C V2.7
c        nposibl(I)=2**1
c        nichflg(I)=1
c        parmin(I)= 0.0001
c        parmax(I)= 0.0001
      ENDDO

      DO I=NPARAM-NOG+1,NPARAM !FOR
ONLY HYD.CON. OF NORMAL CELLS(HDCN)
        nposibl(I)=2**15
        nichflg(I)=1
        parmin(I)= 1.0E-08
        parmax(I)= 1.0E-06
C V2.7
c        nposibl(I)=2**1
c        nichflg(I)=1
c        parmin(I)= 0.0001
c        parmax(I)= 0.0001
      ENDDO
C V2.7

      irestrt=0
      microga=1
      idum=-10000
      pcross=0.5d0
      itourny=1
      ielite=1
      icreep=1
      iunifrm=1
      iniche=0
      nchild=1
      iskip= 0
      iend=  0
      nowrite=1
      kountmx=5
CSJ GA INPUT DATA

      ITOURNY=1
C      IF (ITOURNY.EQ.0) NCHILD=2
C
C  CHECK FOR ARRAY SIZING ERRORS.
      IF (NPOPSIZ.GT.INDMAX) THEN
         WRITE(6,1600) NPOPSIZ
         WRITE(24,1600) NPOPSIZ
         CLOSE(24)
         STOP
      ENDIF
      IF (NPARAM.GT.NPARMAX) THEN
         WRITE(6,1700) NPARAM
         WRITE(24,1700) NPARAM
         CLOSE(24)
         STOP
      ENDIF
C
C  IF USING THE MICROGA OPTION,
RESET SOME INPUT VARIABLES
      IF (MICROGA.NE.0) THEN
         PMUTATE=0.0D0
         PCREEP=0.0D0
         ITOURNY=1
         IELITE=1
         INICHE=0
         NCHILD=1
         IF (IUNIFRM.EQ.0) THEN
            PCROSS=1.0D0
         ELSE
            PCROSS=0.5D0
         ENDIF
      ENDIF
C
 1600 FORMAT(1X,'ERROR: NPOPSIZ >
INDMAX.  SET INDMAX = ',I6)
 1700 FORMAT(1X,'ERROR: NPARAM >
NPARMAX.  SET NPARMAX = ',I6)
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE
INITIAL(ISTART,NPOSSUM,IG2SUM)
C
C  THIS SUBROUTINE SETS UP THE
PROGRAM BY GENERATING THE G0, G1 AND
C  IG2 ARRAYS, AND COUNTING THE
NUMBER OF CHROMOSOMES REQUIRED FOR
THE
C  SPECIFIED INPUT.  THE SUBROUTINE
ALSO INITIALIZES THE RANDOM NUMBER
C  GENERATOR, PARENT AND IPARENT
ARRAYS (READS THE GA.RESTART FILE).
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
```

```
C
      INCLUDE 'PARAMS.F'
      DIMENSION
PARENT(NPARMAX,INDMAX),IPARENT(NCHRM
AX,INDMAX)
      DIMENSION NPOSIBL(NPARMAX)
      DIMENSION
G0(NPARMAX),G1(NPARMAX),IG2(NPARMAX)
      DIMENSION
PARMAX(NPARMAX),PARMIN(NPARMAX),PARD
EL(NPARMAX)
C
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      COMMON / GA5   / G0,G1,IG2
      COMMON / GA6   /
PARMAX,PARMIN,PARDEL,NPOSIBL
      COMMON /INPUTGA/
PCROSS,PMUTATE,PCREEP,MAXGEN,IDUM,IR
ESTRT,
     +
ITOURNY,IELITE,ICREEP,IUNIFRM,INICHE
,
     +
ISKIP,IEND,NCHILD,MICROGA,KOUNTMX
C
C
      DO 3 I=1,NPARAM
         G0(I)=PARMIN(I)
         PARDEL(I)=PARMAX(I)-
PARMIN(I)

G1(I)=PARDEL(I)/DBLE(NPOSIBL(I)-1)
 3    CONTINUE
      DO 6 I=1,NPARAM
         DO 7 J=1,30
            N2J=2**J
            IF (N2J.GE.NPOSIBL(I))
THEN
               IG2(I)=J
               GOTO 8
            ENDIF
            IF (J.GE.30) THEN
               WRITE(6,2000)
               WRITE(24,2000)
               CLOSE(24)
               STOP
            ENDIF
 7       CONTINUE
 8       CONTINUE
 6    CONTINUE
C
C  COUNT THE TOTAL NUMBER OF
CHROMOSOMES (BITS) REQUIRED
      NCHROME=0
      NPOSSUM=0
      IG2SUM=0
      DO 9 I=1,NPARAM
         NCHROME=NCHROME+IG2(I)
         NPOSSUM=NPOSSUM+NPOSIBL(I)
         IG2SUM=IG2SUM+(2**IG2(I))
 9    CONTINUE
      IF (NCHROME.GT.NCHRMAX) THEN
         WRITE(6,1800) NCHROME
         WRITE(24,1800) NCHROME
         CLOSE(24)
         STOP
      ENDIF
C
      IF (NPOSSUM.LT.IG2SUM .AND.
MICROGA.NE.0) THEN
         WRITE(6,2100)
         WRITE(24,2100)
      ENDIF
C
C  INITIALIZE RANDOM NUMBER
GENERATOR
      CALL RAN3(IDUM,RAND)
C
      IF(IRESTRT.EQ.0) THEN
C  INITIALIZE THE RANDOM
DISTRIBUTION OF PARAMETERS IN THE
INDIVIDUAL
C  PARENTS WHEN IRESTRT=0.
         ISTART=1
         DO 10 I=1,NPOPSIZ
            DO 15 J=1,NCHROME
               CALL RAN3(1,RAND)
               IPARENT(J,I)=1
               IF(RAND.LT.0.5D0)
IPARENT(J,I)=0
 15         CONTINUE
 10      CONTINUE
         IF (NPOSSUM.LT.IG2SUM) CALL
POSSIBL(PARENT,IPARENT)
      ELSE
C  IF IRESTRT.NE.0, READ FROM
RESTART FILE.
         OPEN (UNIT=25,
FILE='GA.RESTART', STATUS='OLD')
         REWIND 25
         READ(25,*) ISTART,NPOPSIZ
         DO 1 J=1,NPOPSIZ
            READ(25,*)
K,(IPARENT(L,J),L=1,NCHROME)
 1       CONTINUE
```

```
         CLOSE (25)
      ENDIF
C
      IF(IRESTRT.NE.0) CALL
RAN3(IDUM-ISTART,RAND)
C
 1800 FORMAT(1X,'ERROR: NCHROME >
NCHRMAX.  SET NCHRMAX = ',I6)
 2000 FORMAT(1X,'ERROR: YOU HAVE A
PARAMETER WITH A NUMBER OF '/
     +       1X,'   POSSIBILITIES >
2**30!  IF YOU REALLY DESIRE THIS,'/
     +       1X,'   CHANGE THE DO
LOOP 7 STATEMENT AND RECOMPILE.'//
     +       1X,'   YOU MAY ALSO
NEED TO ALTER THE CODE TO WORK
WITH'/
     +       1X,'   REAL NUMBERS
RATHER THAN INTEGER NUMBERS;
FORTRAN'/
     +       1X,'   DOES NOT LIKE TO
COMPUTE 2**J WHEN J>30.')
 2100 FORMAT(1X,'WARNING: FOR SOME
CASES, A CONSIDERABLE PERFORMANCE'/
     +       1X,'   REDUCTION HAS
BEEN OBSERVED WHEN RUNNING A NON-'/
     +       1X,'   OPTIMAL NUMBER
OF BITS WITH THE MICRO-GA.'/
     +       1X,'   IF POSSIBLE, USE
VALUES FOR NPOSIBL OF 2**N,'/
     +       1X,'   E.G. 2, 4, 8,
16, 32, 64, ETC.  SEE README FILE.')
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE
EVALOUT(ISKIP,IEND,IBEST,FBAR,BEST)
C
C  THIS SUBROUTINE EVALUATES THE
POPULATION, ASSIGNS FITNESS,
C  ESTABLISHES THE BEST INDIVIDUAL,
AND OUTPUTS INFORMATION.
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
       DIMENSION
PARENT(NPARMAX,INDMAX),IPARENT(NCHRM
AX,INDMAX)
      DIMENSION FITNESS(INDMAX)
      DIMENSION
PARAMSM(NPARMAX),PARAMAV(NPARMAX),IB
EST(NCHRMAX)
      DIMENSION WEFR(NNODES)
C
      COMMON/ GWM /SITIME(NTIMES),
HEADMEA(NTIMES,NNODES)
      COMMON/GWM2/ NMD,NST,WEFR
       COMMON/GWM1/NEL,NXM,NYM,NZM
       COMMON/GWM3/NSM

       COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      COMMON / GA4   / FITNESS
C
      FITSUM=0.0D0
      BEST=-1.0D10
      DO 29 N=1,NPARAM
         PARAMSM(N)=0.0D0
 29   CONTINUE
      JSTART=1
      JEND=NPOPSIZ
      IF(ISKIP.NE.0) JSTART=ISKIP
      IF(IEND.NE.0) JEND=IEND
      DO 30 J=JSTART,JEND
         CALL
DECODE(J,PARENT,IPARENT)
         IF(ISKIP.NE.0 .AND.
IEND.NE.0 .AND. ISKIP.EQ.IEND)
     +   WRITE(6,1075)
J,(IPARENT(K,J),K=1,NCHROME),
     +
(PARENT(KK,J),KK=1,NPARAM),0.0
C
C  CALL FUNCTION EVALUATOR, WRITE
OUT INDIVIDUAL AND FITNESS, AND ADD
C  TO THE SUMMATION FOR LATER
AVERAGING.
         CALL FUNC(J,FUNCVAL)
         FITNESS(J)=FUNCVAL
CSJ         WRITE(24,1075)
J,(IPARENT(K,J),K=1,NCHROME),
CSJ     +
(PARENT(KK,J),KK=1,NPARAM),FITNESS(J
)
CSJ    FORMAT FOR GA OUTPUT
         WRITE(24,'(I3,30E20.8)')
J,(PARENT(K,J),K=1,NPARAM),FITNESS(J
)
```

```
         WRITE(*,'(I3,30E20.8)')
J,(PARENT(K,J),K=1,NPARAM),FITNESS(J
)
CSJ
         FITSUM=FITSUM+FITNESS(J)
         DO 22 N=1,NPARAM

PARAMSM(N)=PARAMSM(N)+PARENT(N,J)
 22      CONTINUE
C
C  CHECK TO SEE IF FITNESS OF
INDIVIDUAL J IS THE BEST FITNESS.
         IF (FITNESS(J).GT.BEST)
THEN
            BEST=FITNESS(J)
            JBEST=J
            DO 24 K=1,NCHROME
               IBEST(K)=IPARENT(K,J)
 24         CONTINUE
         ENDIF
 30   CONTINUE
C
C  COMPUTE PARAMETER AND FITNESS
AVERAGES.
      FBAR=FITSUM/DBLE(NPOPSIZ)
      DO 23 N=1,NPARAM

PARAMAV(N)=PARAMSM(N)/DBLE(NPOPSIZ)
 23   CONTINUE
C
C  WRITE OUTPUT INFORMATION
      IF (NPOPSIZ.EQ.1) THEN
         WRITE(24,1075)
1,(IPARENT(K,1),K=1,NCHROME),
     +
(PARENT(K,1),K=1,NPARAM),FITNESS(1)
CSJ         WRITE(24,*) ' AVERAGE
VALUES:'
CSJ         WRITE(24,1275)
(PARENT(K,1),K=1,NPARAM),FBAR
CSJ      ELSE
CSJ         WRITE(24,1275)
(PARAMAV(K),K=1,NPARAM),FBAR
      ENDIF
CSJ      WRITE(6,1100) FBAR
CSJ      WRITE(24,1100) FBAR
CSJ      WRITE(6,1200) BEST
      WRITE(24,1200) BEST
C
 1075
FORMAT(I3,1X,45I1,1X,3E20.8,1X,E20.8
)
 1100 FORMAT(1X,'AVERAGE FUNCTION
VALUE OF GENERATION=',F13.8)
 1200 FORMAT(1X,'MAXIMUM FUNCTION
VALUE              =',F18.10)
 1275 FORMAT(/' AVERAGE
VALUES:',18X,1X,11F15.8/)
      RETURN
      END
C
C###################################
###################################
      SUBROUTINE NICHE
C
C  IMPLEMENT "NICHING" THROUGH
GOLDBERG'S MULTIDIMENSIONAL
PHENOTYPIC
C  SHARING SCHEME WITH A TRIANGULAR
SHARING FUNCTION.  TO FIND THE
C  MULTIDIMENSIONAL DISTANCE FROM
THE BEST INDIVIDUAL, NORMALIZE ALL
C  PARAMETER DIFFERENCES.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      DIMENSION
PARENT(NPARMAX,INDMAX),IPARENT(NCHRM
AX,INDMAX)
      DIMENSION
FITNESS(INDMAX),NPOSIBL(NPARMAX),NIC
HFLG(NPARMAX)
      DIMENSION
PARMAX(NPARMAX),PARMIN(NPARMAX),PARD
EL(NPARMAX)
C
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      COMMON / GA4   / FITNESS
      COMMON / GA6   /
PARMAX,PARMIN,PARDEL,NPOSIBL
      COMMON / GA8   / NICHFLG
C
C   VARIABLE DEFINITIONS:
C
C  ALPHA   = POWER LAW EXPONENT FOR
SHARING FUNCTION; TYPICALLY = 1.0
C  DEL     = NORMALIZED
MULTIDIMENSIONAL DISTANCE BETWEEN II
AND ALL
C            OTHER MEMBERS OF THE
POPULATION
```

```
C            (EQUALS THE SQUARE ROOT
OF DEL2)
C  DEL2    = SUM OF THE SQUARES OF
THE NORMALIZED MULTIDIMENSIONAL
C            DISTANCE BETWEEN MEMBER
II AND ALL OTHER MEMBERS OF
C            THE POPULATION
C  NNICHE  = NUMBER OF NICHED
PARAMETERS
C  SIGSHAR = NORMALIZED DISTANCE TO
BE COMPARED WITH DEL; IN SOME SENSE,
C            1/SIGSHAR CAN BE VIEWED
AS THE NUMBER OF REGIONS OVER WHICH
C            THE SHARING FUNCTION
SHOULD FOCUS, E.G. WITH SIGSHAR=0.1,
C            THE SHARING FUNCTION
WILL TRY TO CLUMP IN TEN DISTINCT
C            REGIONS OF THE PHASE
SPACE.  A VALUE OF SIGSHAR ON THE
C            ORDER OF 0.1 SEEMS TO
WORK BEST.
C  SHARE   = SHARING FUNCTION
BETWEEN INDIVIDUAL II AND J
C  SUMSHAR = SUM OF THE SHARING
FUNCTIONS FOR INDIVIDUAL II
C
C      ALPHA=1.0
      SIGSHAR=0.1D0
      NNICHE=0
      DO 33 JJ=1,NPARAM
         NNICHE=NNICHE+NICHFLG(JJ)
 33   CONTINUE
      IF (NNICHE.EQ.0) THEN
         WRITE(6,1900)
         WRITE(24,1900)
         CLOSE(24)
         STOP
      ENDIF
      DO 34 II=1,NPOPSIZ
         SUMSHAR=0.0D0
         DO 35 J=1,NPOPSIZ
            DEL2=0.0D0
            DO 36 K=1,NPARAM
               IF (NICHFLG(K).NE.0)
THEN

DEL2=DEL2+((PARENT(K,J)-
PARENT(K,II))/PARDEL(K))**2
               ENDIF
 36         CONTINUE

DEL=(DSQRT(DEL2))/DBLE(NNICHE)
            IF (DEL.LT.SIGSHAR) THEN
C               SHARE=1.0-
((DEL/SIGSHAR)**ALPHA)
               SHARE=1.0D0-
(DEL/SIGSHAR)
            ELSE
               SHARE=0.0D0
            ENDIF

SUMSHAR=SUMSHAR+SHARE/DBLE(NPOPSIZ)
 35      CONTINUE
         IF (SUMSHAR.NE.0.0D0)
FITNESS(II)=FITNESS(II)/SUMSHAR
 34   CONTINUE
C
 1900 FORMAT(1X,'ERROR: INICHE=1 AND
ALL VALUES IN NICHFLG ARRAY = 0'/
     +       1X,'       DO YOU WANT
TO NICHE OR NOT?')
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE
SELECTN(IPICK,J,MATE1,MATE2)
C
C  SUBROUTINE FOR SELECTION
OPERATOR.  PRESENTLY, TOURNAMENT
SELECTION
C  IS THE ONLY OPTION AVAILABLE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      DIMENSION
PARENT(NPARMAX,INDMAX),CHILD(NPARMAX
,INDMAX)
      DIMENSION FITNESS(INDMAX)
      DIMENSION
IPARENT(NCHRMAX,INDMAX),ICHILD(NCHRM
AX,INDMAX)
C
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      COMMON / GA4   / FITNESS
      COMMON / GA7   / CHILD,ICHILD
      COMMON /INPUTGA/
PCROSS,PMUTATE,PCREEP,MAXGEN,IDUM,IR
ESTRT,
```

```
     +
ITOURNY,IELITE,ICREEP,IUNIFRM,INICHE
,
     +
ISKIP,IEND,NCHILD,MICROGA,KOUNTMX
C
C  IF TOURNAMENT SELECTION IS CHOSEN
(I.E. ITOURNY=1), THEN
C  IMPLEMENT "TOURNAMENT" SELECTION
FOR SELECTION OF NEW POPULATION.
      IF(ITOURNY.EQ.1) THEN
         CALL SELECT(MATE1,IPICK)
         CALL SELECT(MATE2,IPICK)
C        WRITE(3,*)
MATE1,MATE2,FITNESS(MATE1),FITNESS(M
ATE2)
         DO 46 N=1,NCHROME

ICHILD(N,J)=IPARENT(N,MATE1)
            IF(NCHILD.EQ.2)
ICHILD(N,J+1)=IPARENT(N,MATE2)
 46      CONTINUE
      ENDIF
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE
CROSOVR(NCROSS,J,MATE1,MATE2)
C
C  SUBROUTINE FOR CROSSOVER BETWEEN
THE RANDOMLY SELECTED PAIR.
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      DIMENSION
PARENT(NPARMAX,INDMAX),CHILD(NPARMAX
,INDMAX)
      DIMENSION
IPARENT(NCHRMAX,INDMAX),ICHILD(NCHRM
AX,INDMAX)
C
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      COMMON / GA7   / CHILD,ICHILD
      COMMON /INPUTGA/
PCROSS,PMUTATE,PCREEP,MAXGEN,IDUM,IR
ESTRT,
     +
ITOURNY,IELITE,ICREEP,IUNIFRM,INICHE
,
     +
ISKIP,IEND,NCHILD,MICROGA,KOUNTMX
C
      IF (IUNIFRM.EQ.0) THEN
C  SINGLE-POINT CROSSOVER AT A
RANDOM CHROMOSOME POINT.
         CALL RAN3(1,RAND)
         IF(RAND.GT.PCROSS) GOTO 69
         NCROSS=NCROSS+1
         CALL RAN3(1,RAND)
         ICROSS=2+DINT(DBLE(NCHROME-
1)*RAND)
         DO 50 N=ICROSS,NCHROME

ICHILD(N,J)=IPARENT(N,MATE2)
            IF(NCHILD.EQ.2)
ICHILD(N,J+1)=IPARENT(N,MATE1)
 50      CONTINUE
      ELSE
C  PERFORM UNIFORM CROSSOVER BETWEEN
THE RANDOMLY SELECTED PAIR.
         DO 60 N=1,NCHROME
            CALL RAN3(1,RAND)
            IF(RAND.LE.PCROSS) THEN
               NCROSS=NCROSS+1

ICHILD(N,J)=IPARENT(N,MATE2)
               IF(NCHILD.EQ.2)
ICHILD(N,J+1)=IPARENT(N,MATE1)
            ENDIF
 60      CONTINUE
      ENDIF
 69   CONTINUE
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE MUTATE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      DIMENSION NPOSIBL(NPARMAX)
      DIMENSION
CHILD(NPARMAX,INDMAX),ICHILD(NCHRMAX
,INDMAX)
      DIMENSION
G0(NPARMAX),G1(NPARMAX),IG2(NPARMAX)
```

```
      DIMENSION
PARMAX(NPARMAX),PARMIN(NPARMAX),PARD
EL(NPARMAX)
C
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA5   / G0,G1,IG2
      COMMON / GA6   /
PARMAX,PARMIN,PARDEL,NPOSIBL
      COMMON / GA7   / CHILD,ICHILD
      COMMON /INPUTGA/
PCROSS,PMUTATE,PCREEP,MAXGEN,IDUM,IR
ESTRT,
     +
ITOURNY,IELITE,ICREEP,IUNIFRM,INICHE
,
     +
ISKIP,IEND,NCHILD,MICROGA,KOUNTMX
C
C  THIS SUBROUTINE PERFORMS
MUTATIONS ON THE CHILDREN
GENERATION.
C  PERFORM RANDOM JUMP MUTATION IF A
RANDOM NUMBER IS LESS THAN PMUTATE.
C  PERFORM RANDOM CREEP MUTATION IF
A DIFFERENT RANDOM NUMBER IS LESS
C  THAN PCREEP.
      NMUTATE=0
      NCREEP=0
      DO 70 J=1,NPOPSIZ
         DO 75 K=1,NCHROME
C  JUMP MUTATION
            CALL RAN3(1,RAND)
            IF (RAND.LE.PMUTATE)
THEN
               NMUTATE=NMUTATE+1
               IF(ICHILD(K,J).EQ.0)
THEN
                  ICHILD(K,J)=1
               ELSE
                  ICHILD(K,J)=0
               ENDIF
               IF (NOWRITE.EQ.0)
WRITE(6,1300) J,K
               IF (NOWRITE.EQ.0)
WRITE(24,1300) J,K
            ENDIF
 75      CONTINUE
C  CREEP MUTATION (ONE DISCRETE
POSITION AWAY).
         IF (ICREEP.NE.0) THEN
            DO 76 K=1,NPARAM
               CALL RAN3(1,RAND)
               IF(RAND.LE.PCREEP)
THEN
                  CALL
DECODE(J,CHILD,ICHILD)
                  NCREEP=NCREEP+1
                  CREEP=1.0D0
                  CALL RAN3(1,RAND)
                  IF (RAND.LT.0.5D0)
CREEP=-1.0D0

CHILD(K,J)=CHILD(K,J)+G1(K)*CREEP
                  IF
(CHILD(K,J).GT.PARMAX(K)) THEN

CHILD(K,J)=PARMAX(K)-1.0D0*G1(K)
                  ELSEIF
(CHILD(K,J).LT.PARMIN(K)) THEN

CHILD(K,J)=PARMIN(K)+1.0D0*G1(K)
                  ENDIF
                  CALL
CODE(J,K,CHILD,ICHILD)
                  IF (NOWRITE.EQ.0)
WRITE(6,1350) J,K
                  IF (NOWRITE.EQ.0)
WRITE(24,1350) J,K
               ENDIF
 76         CONTINUE
         ENDIF
 70   CONTINUE
      WRITE(6,1250) NMUTATE,NCREEP
      WRITE(24,1250) NMUTATE,NCREEP
C
 1250 FORMAT(/'  NUMBER OF JUMP
MUTATIONS  =',I5/
     +        '  NUMBER OF CREEP
MUTATIONS =',I5)
 1300 FORMAT('*** JUMP MUTATION
PERFORMED ON INDIVIDUAL  ',I4,
     +       ', CHROMOSOME ',I3,'
***')
 1350 FORMAT('*** CREEP MUTATION
PERFORMED ON INDIVIDUAL ',I4,
     +       ', PARAMETER  ',I3,'
***')
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE
NEWGEN(IELITE,NPOSSUM,IG2SUM,IBEST)
C
```

```
C  WRITE CHILD ARRAY BACK INTO
PARENT ARRAY FOR NEW GENERATION.
CHECK
C  TO SEE IF THE BEST PARENT WAS
REPLICATED; IF NOT, AND IF IELITE=1,
C  THEN REPRODUCE THE BEST PARENT
INTO A RANDOM SLOT.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      DIMENSION
PARENT(NPARMAX,INDMAX),CHILD(NPARMAX
,INDMAX)
      DIMENSION
IPARENT(NCHRMAX,INDMAX),ICHILD(NCHRM
AX,INDMAX)
      DIMENSION IBEST(NCHRMAX)
C
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      COMMON / GA7   / CHILD,ICHILD
C
      IF (NPOSSUM.LT.IG2SUM) CALL
POSSIBL(CHILD,ICHILD)
      KELITE=0
      DO 94 J=1,NPOPSIZ
         JELITE=0
         DO 95 N=1,NCHROME
            IPARENT(N,J)=ICHILD(N,J)
            IF
(IPARENT(N,J).EQ.IBEST(N))
JELITE=JELITE+1
            IF (JELITE.EQ.NCHROME)
KELITE=1
 95      CONTINUE
 94   CONTINUE
      IF (IELITE.NE.0 .AND.
KELITE.EQ.0) THEN
         CALL RAN3(1,RAND)

IRAND=1D0+DINT(DBLE(NPOPSIZ)*RAND)
         DO 96 N=1,NCHROME

IPARENT(N,IRAND)=IBEST(N)
 96      CONTINUE
CSJ         WRITE(24,1260) IRAND
      ENDIF
C
 1260 FORMAT('  ELITIST REPRODUCTION
ON INDIVIDUAL ',I4)
C
      RETURN
      END
C
C##################################
##################################
      SUBROUTINE
GAMICRO(I,NPOSSUM,IG2SUM,IBEST)
C
C  MICRO-GA IMPLEMENTATION
SUBROUTINE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      DIMENSION
PARENT(NPARMAX,INDMAX),IPARENT(NCHRM
AX,INDMAX)
      DIMENSION IBEST(NCHRMAX)
C
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
C
C  FIRST, CHECK FOR CONVERGENCE OF
MICRO POPULATION.
C  IF CONVERGED, START A NEW
GENERATION WITH BEST INDIVIDUAL AND
FILL
C  THE REMAINDER OF THE POPULATION
WITH NEW RANDOMLY GENERATED PARENTS.
C
C  COUNT NUMBER OF DIFFERENT BITS
FROM BEST MEMBER IN MICRO-POPULATION
      ICOUNT=0
      DO 81 J=1,NPOPSIZ
         DO 82 N=1,NCHROME

IF(IPARENT(N,J).NE.IBEST(N))
ICOUNT=ICOUNT+1
 82      CONTINUE
 81   CONTINUE
C
C  IF ICOUNT LESS THAN 5% OF NUMBER
OF BITS, THEN CONSIDER POPULATION
C  TO BE CONVERGED.  RESTART WITH
BEST INDIVIDUAL AND RANDOM OTHERS.
```

```
DIFFRAC=DBLE(ICOUNT)/DBLE((NPOPSIZ-
1)*NCHROME)
      IF (DIFFRAC.LT.0.05D0) THEN
      DO 87 N=1,NCHROME
         IPARENT(N,1)=IBEST(N)
 87   CONTINUE
      DO 88 J=2,NPOPSIZ
         DO 89 N=1,NCHROME
            CALL RAN3(1,RAND)
            IPARENT(N,J)=1
            IF(RAND.LT.0.5D0)
IPARENT(N,J)=0
 89      CONTINUE
 88   CONTINUE
      IF (NPOSSUM.LT.IG2SUM) CALL
POSSIBL(PARENT,IPARENT)
      WRITE(6,1375) I
      WRITE(24,1375) I
      ENDIF
C
 1375 FORMAT(//'%%%%%%%  RESTART
MICRO-POPULATION AT GENERATION',
     +       I5,'  %%%%%%%')
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE SELECT(MATE,IPICK)
C
C  THIS ROUTINE SELECTS THE BETTER
OF TWO POSSIBLE PARENTS FOR MATING.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      COMMON / GA4   / FITNESS
      DIMENSION
PARENT(NPARMAX,INDMAX),IPARENT(NCHRM
AX,INDMAX)
      DIMENSION FITNESS(INDMAX)
C
      IF(IPICK+1.GT.NPOPSIZ) CALL
SHUFFLE(IPICK)
      IFIRST=IPICK
      ISECOND=IPICK+1
      IPICK=IPICK+2

IF(FITNESS(IFIRST).GT.FITNESS(ISECON
D)) THEN
         MATE=IFIRST
      ELSE
         MATE=ISECOND
      ENDIF
C
WRITE(3,*)'SELECT',IFIRST,ISECOND,FI
TNESS(IFIRST),FITNESS(ISECOND)
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE SHUFFLE(IPICK)
C
C  THIS ROUTINE SHUFFLES THE PARENT
ARRAY AND ITS CORRESPONDING FITNESS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      COMMON / GA4   / FITNESS
      DIMENSION
PARENT(NPARMAX,INDMAX),IPARENT(NCHRM
AX,INDMAX)
      DIMENSION FITNESS(INDMAX)
C
      IPICK=1
      DO 10 J=1,NPOPSIZ-1
         CALL RAN3(1,RAND)

IOTHER=J+1+DINT(DBLE(NPOPSIZ-
J)*RAND)
         DO 20 N=1,NCHROME
            ITEMP=IPARENT(N,IOTHER)

IPARENT(N,IOTHER)=IPARENT(N,J)
            IPARENT(N,J)=ITEMP
 20      CONTINUE
         TEMP=FITNESS(IOTHER)
         FITNESS(IOTHER)=FITNESS(J)
         FITNESS(J)=TEMP
 10   CONTINUE
C
```

```
      RETURN
      END
C
C#################################
#################################
      SUBROUTINE
DECODE(I,ARRAY,IARRAY)
C
C  THIS ROUTINE DECODES A BINARY
STRING TO A REAL NUMBER.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA5   / G0,G1,IG2
      DIMENSION
ARRAY(NPARMAX,INDMAX),IARRAY(NCHRMAX
,INDMAX)
      DIMENSION
G0(NPARMAX),G1(NPARMAX),IG2(NPARMAX)
C
      L=1
      DO 10 K=1,NPARAM
         IPARAM=0
         M=L
         DO 20 J=M,M+IG2(K)-1
            L=L+1

IPARAM=IPARAM+IARRAY(J,I)*(2**(M+IG2
(K)-1-J))
 20      CONTINUE

ARRAY(K,I)=G0(K)+G1(K)*DBLE(IPARAM)
 10   CONTINUE
C
      RETURN
      END
C
C#################################
#################################
      SUBROUTINE
CODE(J,K,ARRAY,IARRAY)
C
C  THIS ROUTINE CODES A PARAMETER
INTO A BINARY STRING.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA5   / G0,G1,IG2
      DIMENSION
ARRAY(NPARMAX,INDMAX),IARRAY(NCHRMAX
,INDMAX)
      DIMENSION
G0(NPARMAX),G1(NPARMAX),IG2(NPARMAX)
C
C  FIRST, ESTABLISH THE BEGINNING
LOCATION OF THE PARAMETER STRING OF
C  INTEREST.
      ISTART=1
      DO 10 I=1,K-1
         ISTART=ISTART+IG2(I)
 10   CONTINUE
C
C  FIND THE EQUIVALENT CODED
PARAMETER VALUE, AND BACK OUT THE
BINARY
C  STRING BY FACTORS OF TWO.
      M=IG2(K)-1
      IF (G1(K).EQ.0.0D0) RETURN
      IPARAM=NINT((ARRAY(K,J)-
G0(K))/G1(K))
      DO 20 I=ISTART,ISTART+IG2(K)-1
         IARRAY(I,J)=0
         IF ((IPARAM+1).GT.(2**M))
THEN
            IARRAY(I,J)=1
            IPARAM=IPARAM-2**M
         ENDIF
         M=M-1
 20   CONTINUE
C
WRITE(3,*)ARRAY(K,J),IPARAM,(IARRAY(
I,J),I=ISTART,ISTART+IG2(K)-1)
C
      RETURN
      END
C
C#################################
#################################
C
      SUBROUTINE
POSSIBL(ARRAY,IARRAY)
C
C  THIS SUBROUTINE DETERMINES
WHETHER OR NOT ALL PARAMETERS ARE
WITHIN
C  THE SPECIFIED RANGE OF
POSSIBILITY.  IF NOT, THE PARAMETER
IS
C  RANDOMLY REASSIGNED WITHIN THE
RANGE.  THIS SUBROUTINE IS ONLY
C  NECESSARY WHEN THE NUMBER OF
POSSIBILITIES PER PARAMETER IS NOT
```

```
C  OPTIMIZED TO BE 2**N, I.E. IF
NPOSSUM < IG2SUM.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA5   / G0,G1,IG2
      COMMON / GA6   /
PARMAX,PARMIN,PARDEL,NPOSIBL
      DIMENSION
ARRAY(NPARMAX,INDMAX),IARRAY(NCHRMAX
,INDMAX)
      DIMENSION
G0(NPARMAX),G1(NPARMAX),IG2(NPARMAX)
,NPOSIBL(NPARMAX)
      DIMENSION
PARMAX(NPARMAX),PARMIN(NPARMAX),PARD
EL(NPARMAX)
C
      DO 10 I=1,NPOPSIZ
         CALL DECODE(I,ARRAY,IARRAY)
         DO 20 J=1,NPARAM
            N2IG2J=2**IG2(J)
            IF(NPOSIBL(J).NE.N2IG2J
.AND. ARRAY(J,I).GT.PARMAX(J)) THEN
               CALL RAN3(1,RAND)

IRAND=DINT(DBLE(NPOSIBL(J))*RAND)

ARRAY(J,I)=G0(J)+DBLE(IRAND)*G1(J)
               CALL
CODE(I,J,ARRAY,IARRAY)
               IF (NOWRITE.EQ.0)
WRITE(6,1000) I,J
               IF (NOWRITE.EQ.0)
WRITE(24,1000) I,J
            ENDIF
 20      CONTINUE
 10   CONTINUE
C
 1000 FORMAT('*** PARAMETER
ADJUSTMENT TO INDIVIDUAL     ',I4,
     +       ', PARAMETER  ',I3,'
***')
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE
RESTART(I,ISTART,KOUNT)
C
C  THIS SUBROUTINE WRITES RESTART
INFORMATION TO THE GA.RESTART FILE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      COMMON / GA1   /
NPOPSIZ,NOWRITE
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
      DIMENSION
PARENT(NPARMAX,INDMAX),IPARENT(NCHRM
AX,INDMAX)
      COMMON /INPUTGA/
PCROSS,PMUTATE,PCREEP,MAXGEN,IDUM,IR
ESTRT,
     +
ITOURNY,IELITE,ICREEP,IUNIFRM,INICHE
,
     +
ISKIP,IEND,NCHILD,MICROGA,KOUNTMX

      KOUNT=KOUNT+1
      IF(I.EQ.MAXGEN+ISTART-1 .OR.
KOUNT.EQ.KOUNTMX) THEN
         OPEN (UNIT=25,
FILE='GA.RESTART', STATUS='OLD')
         REWIND 25
         WRITE(25,*) I+1,NPOPSIZ
         DO 80 J=1,NPOPSIZ
            WRITE(25,1500)
J,(IPARENT(L,J),L=1,NCHROME)
 80      CONTINUE
         CLOSE (25)
         KOUNT=0
      ENDIF
C
 1500 FORMAT(I5,3X,45I2)
C
      RETURN
      END
C
C####################################
####################################
      SUBROUTINE RAN3(IDUM,RAND)
C
C  RETURNS A UNIFORM RANDOM DEVIATE
BETWEEN 0.0 AND 1.0.  SET IDUM TO
```

```
C  ANY NEGATIVE VALUE TO INITIALIZE
OR REINITIALIZE THE SEQUENCE.
C  THIS FUNCTION IS TAKEN FROM W.H.
PRESS', "NUMERICAL RECIPES" P. 199.
C
      IMPLICIT REAL*8 (A-H,M,O-Z)
      SAVE
C      IMPLICIT REAL*4(M)
      PARAMETER
(MBIG=4000000.,MSEED=1618033.,MZ=0.,
FAC=1./MBIG)
C     PARAMETER
(MBIG=1000000000,MSEED=161803398,MZ=
0,FAC=1./MBIG)
C
C  ACCORDING TO KNUTH, ANY LARGE
MBIG, AND ANY SMALLER (BUT STILL
LARGE)
C  MSEED CAN BE SUBSTITUTED FOR THE
ABOVE VALUES.
      DIMENSION MA(55)
      DATA IFF /0/
      IF (IDUM.LT.0 .OR. IFF.EQ.0)
THEN
         IFF=1
         MJ=MSEED-DBLE(IABS(IDUM))
         MJ=DMOD(MJ,MBIG)
         MA(55)=MJ
         MK=1
         DO 11 I=1,54
            II=MOD(21*I,55)
            MA(II)=MK
            MK=MJ-MK
            IF(MK.LT.MZ) MK=MK+MBIG
            MJ=MA(II)
 11      CONTINUE
         DO 13 K=1,4
            DO 12 I=1,55
               MA(I)=MA(I)-
MA(1+MOD(I+30,55))
               IF(MA(I).LT.MZ)
MA(I)=MA(I)+MBIG
 12         CONTINUE
 13      CONTINUE
         INEXT=0
         INEXTP=31
         IDUM=1
      ENDIF
      INEXT=INEXT+1
      IF(INEXT.EQ.56) INEXT=1
      INEXTP=INEXTP+1
      IF(INEXTP.EQ.56) INEXTP=1
      MJ=MA(INEXT)-MA(INEXTP)
      IF(MJ.LT.MZ) MJ=MJ+MBIG
      MA(INEXT)=MJ
      RAND=MJ*FAC
      RETURN
      END
C
C###################################
###################################
C
      SUBROUTINE FUNC(J,FUNCVAL)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE
C
      INCLUDE 'PARAMS.F'
      DIMENSION
PARENT(NPARMAX,INDMAX)
      DIMENSION
IPARENT(NCHRMAX,INDMAX)

      DIMENSION
HEADCAL(NTIMES,NNODES)
      DIMENSION
VKAOL(NPARMAX),TMLEAK(NPARMAX),HDCN(
NPARMAX)
      DIMENSION
NXM(NXNODE),NYM(NYNODE),NZM(NZNODE)
      DIMENSION STIME(NTIMES)
      DIMENSION
TCS(NTIMES),TMS(NTIMES),HHC(NTIMES,N
NODES),HHM(NTIMES,NNODES)
      DIMENSION NPG(NNODES)
      DIMENSION WEFR(NNODES)
      INTEGER
XEL(NXNODE),YEL(NYNODE),ZEL(NZNODE),
ZELU(NZNODE)
      INTEGER
XOG(NXNODE,NEHCG),YOG(NYNODE,NEHCG),
ZOG(NZNODE,NEHCG)

C      DIMENSION
PARENT2(INDMAX,NPARMAX),IPARENT2(IND
MAX,NCHRMAX)
C
      COMMON/ GWM
/SITIME(NTIMES),HEADMEA(NTIMES,NNODE
S)
      COMMON/GWM2/ NMD,NST,WEFR
      COMMON/GWM1/NEL,NXM,NYM,NZM
      COMMON/GWM3/NSM
      COMMON/GWM4/ELTIME
      COMMON/GWM5/XEL,YEL,ZEL,ZELU
      COMMON/GWM6/NPOPULA,NGENERA
      COMMON/GWM7/XOG,YOG,ZOG,NPG,N
TG,NOG
      COMMON/FGW/ NFI
      COMMON/GRI/GRD(NTIMES)
```

```
      COMMON / GA2   /
NPARAM,NCHROME
      COMMON / GA3   /
PARENT,IPARENT
C
C  THIS IS AN N-DIMENSIONAL VERSION
OF THE MULTIMODAL FUNCTION WITH
C  DECREASING PEAKS USED BY GOLDBERG
AND RICHARDSON (1987, SEE README
C  FILE FOR COMPLETE REFERENCE).  IN
N DIMENSIONS, THIS FUNCTION HAS
C  (NVALLEY-1)^NPARAM PEAKS, BUT
ONLY ONE GLOBAL MAXIMUM.  IT IS A
C  REASONABLY TOUGH PROBLEM FOR THE
GA, ESPECIALLY FOR HIGHER DIMENSIONS
C  AND LARGER VALUES OF NVALLEY.

CSJ THE NEXT PART IS FOR OBJECTIVE
FUNCTION. SO I EDITED BELOW.
C      NVALLEY=6
C      PI=4.0D0*DATAN(1.D0)
C      FUNCVAL=1.0D0
C      DO 10 I=1,NPARAM
C
F1=(SIN(5.1D0*PI*PARENT(I,J) +
0.5D0))**NVALLEY
C         F2=EXP(-
4.0D0*LOG(2.0D0)*((PARENT(I,J)-
0.0667D0)**2)/0.64D0)
C         FUNCVAL=FUNCVAL*F1*F2
C 10   CONTINUE
CSJ
      DO I=1,NEL
        TMLEAK(I)=ELTIME
       ENDDO
      DO I=1,NEL
        VKAOL(I)=PARENT(I,J)
      ENDDO
C V2.7
      NGD=0
      DO I=NEL+1,NPARAM-NOG !VALUES
FOR INJECTION RATE
        NGD=NGD+1
        GRD(NGD)=PARENT(I,J)
      ENDDO

      NC=0
      DO I=NPARAM-NOG+1,NPARAM
         NC=NC+1
         HDCN(NC)=PARENT(I,J)
       ENDDO
C V2.7
      CALL
GW(HEADCAL,VKAOL,TMLEAK,HDCN,STIME,N
IT,ET)

C ----LINEAR INTERPOLATION TO MATCH
CALCULATE AND MEASURED DATA AT SAME
TIME
      DO IJ=1,NIT
       TCS(IJ)=STIME(IJ) ! CONVERT
CALCULATED TIME SERIES (STIME) TO
TCS
      ENDDO
      DO IK=1,NST
       TMS(IK)=SITIME(IK) ! CONVERT
MEASURED TIME SERIES (SITIME) TO TMS
      ENDDO
      DO IL=1,NIT
        DO IM=1,NMD
         HHC(IL,IM)=HEADCAL(IL,IM) !
CONVERT CALCULATED HEAD
        ENDDO
      ENDDO
C      DO IN=1,NST
C        DO IP=1,NMD
C         HHM(IN,IP)=HEADMEA(IN,IP)
! CONVERT MEASURED HEAD
C        ENDDO
C      ENDDO
C  LINEAR INTERPOLATION OF
MEASUREMENT DATA TO CORRESPOND WITH
CALCULATED DATA
      DO II=1,NMD
        DO IL=1,NIT
          DO IJ=1,(NST-1)

IF(TCS(IL).EQ.TMS(IJ))THEN

HHM(IL,II)=HEADMEA(IJ,II)

ELSEIF(TCS(IL).GT.TMS(IJ).AND.TCS(IL
).LT.TMS(IJ+1))THEN

HHM(IL,II)=((HEADMEA(IJ+1,II)-
HEADMEA(IJ,II))*
     >          (TCS(IL)-
TMS(IJ)))/(TMS(IJ+1)-
TMS(IJ))+HEADMEA(IJ,II)

ELSEIF(TCS(IL).EQ.TMS(NST))THEN

HHM(IL,II)=HEADMEA(NST,II)
             ENDIF
          ENDDO
        ENDDO
      ENDDO
```

```
C -------- END OF LINEAR
INTERPOLATION
C -------- OBJECTIVE FUNCTION. HERE,
WEFR:WEIGHTING FACTOR
       SUM=0.0
       DO I=1,NMD
         DO K=1,NIT
C         LEAST SQUARE ERROR
         SUM=SUM+((HHC(K,I)-
HHM(K,I))**2.)/WEFR(I)**2.
C         RELATIVE ERROR
C         SUM=SUM+ABS(HHM(K,I)-
HHC(K,I))/HHM(K,I)
         ENDDO
       ENDDO
      FUNCVAL=-SUM
      CFC1=FUNCVAL
C V2.7  NSM: NUMBER OF INVERSE
SIMULATIONS
      IF ((NSM-1).EQ.1)THEN
       CFC2=-1.0E20
      ENDIF

      IF(CFC1.GE.CFC2)THEN
      WRITE(5,'(A)')' TIME
CALCULATED HEAD      MEASURED HEAD
     >RESIDUAL(CH-MH)'
      DO II=1,NMD
        DO JJ=1,NIT
         WRITE(5,'(F14.4,3E17.9)')
TCS(JJ),HHC(JJ,II),HHM(JJ,II),
     >(HHC(JJ,II)-HHM(JJ,II))
        ENDDO
      ENDDO
      WRITE(5,'(/,A,E17.9,//)')'
OBJECTIVE FUNCTION VALUE =', CFC1
      CFC2=CFC1
      ENDIF

      IF ((NSM-
1).NE.(NGENERA*NPOPULA))THEN
       REWIND(5)
      ELSE
       CLOSE(5)
      ENDIF
CSJ THE END OF MAKING EQUATION OF
FUNCTION VALUE

C  AS MENTIONED IN THE README FILE,
THE ARRAYS HAVE BEEN REARRANGED
C  TO ENABLE A MORE EFFICIENT
CACHING OF SYSTEM MEMORY.  IF THIS
CAUSES
C  INTERFACE PROBLEMS WITH EXISTING
FUNCTIONS USED WITH PREVIOUS
C  VERSIONS OF MY CODE, THEN YOU CAN
USE SOME TEMPORARY ARRAYS TO BRIDGE
C  THIS VERSION WITH OLDER VERSIONS.
I'VE NAMED THE TEMPORARY ARRAYS
C  PARENT2 AND IPARENT2.  IF YOU
WANT TO USE THESE ARRAYS, UNCOMMENT
THE
C  DIMENSION STATEMENT ABOVE AS WELL
AS THE FOLLOWING DO LOOP LINES.
C
C      DO 11 I=1,NPARAM
C         PARENT2(J,I)=PARENT(I,J)
C 11   CONTINUE
C      DO 12 K=1,NCHROME
C         IPARENT2(J,K)=IPARENT(K,J)
C 12   CONTINUE
C
      RETURN
      END

C
#####################################
#####################################

      SUBROUTINE
GW(HEADCAL,VKAOL,TMLEAK,HDCN,STIME,N
IT,ET)
      IMPLICIT REAL*8(A-H,O-Z)

C  THIS FORWARD SIMULATION IS
VERSION 2.5 WITH ADJUSTED LEAKAGE
TERM,
C  LEAKAGE SIMULATION HAS THREE
OPTIONS; FIRST, THE LEAKAGE PATHWAY
IS ALREADY OPENED
C  SO LEAKAGE TRAVEL TIME IS ZERO,
SECOND, THE LEAKAGE PATHWAY IS
GENERATED ON SOME TIME
C  AND LEAKAGE REACHES AT END OF
LEAKAGE PATHWAY AFTER LEAKAGE TRAVEL
TIME, THIRD OPTION IS
C  SAME AS SECOND OPTION BUT LEAKAGE
TRAVEL TIME IS EQUAL TO ZERO, I.E.,
LEAKAGE STARTING TIME OF
C  BOTH AQUIFERS IS SAME.
C
L(i,j,k)=(KZBLEAK(i,j,k)*ALEAK(i,j,k
)/DZBLEAK(I)*(H(i,j,k)-
H(i,j,ZLSU(I))))/(DX(i)*DY(j)*DZ(k))
.
C  SO
CZBLEAK(i,j,k)=KZBLEAK(i,j,k)*ALEAK(
i,j,k)/(DZBLEAK(I)*DX(i)*DY(j)*DZ(k)
),
```

```
C  INPUT FILE AND INTERPOLATION FOR
CALIBRATION POINT OF MEASUREMENT
DATA.
C  I REMOVED TIME DIMENSION OF
VARIABLES HEAD,Q,V.
C  LINEAR FINITE DIFFERENCE METHOD
TO SOLVE THREE-DIMENSIONAL
GROUNDWATER
C  FLOW EQUATION WITH LEAKAGE TERM.
THIS PROGRAM IS FOR STEADY AND
UNSTEADY
C  CONDITION BY BLOCK-CENTERED
METHOD. ALSO, IT IS POSSIBLE FOR
HETEROGENEOUS AND
C  ISOTHERMAL CONDITION. TO SOLVE
LINEAR SYSTEM, I USED GAUSS-SEIDAL
METHOD.
C  BASICALLY, THIS PROGRAM USES
SOLUTIONS OF STEADY STATE CONDITION
FOR
C  INITIAL CONDITION OF UNSTEADY
(TRANSIENT) CONDITION.
C
C
C  NSIMCON: 1 OR 2 FOR FLOW
CONDITION. 1:STEADY STATE CONDITION,
2:UNSTEADY CONDITION
C  NX : NUMBER OF X-DIRECTION NODES
FROM 1 TO NX (NODE 1 AND NX MUST BE
BOUNDARY
C  NODES).
C  NY : NUMBER OF Y-DIRECTION NODES
FROM 1 TO NY (1 AND NY MUST BE
BOUNDARY
C  NODES).
C  NH : NUMBER OF Z-DIRECTION NODES
FROM 1 TO NH (1 AND NH MUST BE
BOUNDARY
C  NODES).
C  ST : STARTING TIME OF SIMULATION
C  ET : ENDING TIME OF SIMULATION
C  NT : TOTAL NUMBER OF TIME STEPS
C  DT : TIME STEP SIZE FOR
SIMULATION EXCEPT LEAKAGE STARTING
AND APPROACHING TIME
C  NLC : NUMBER OF LEAKAGE COLUMNS
C  TOL : TOLERANCE OF LINEAR MATRIX
SOLVER
C  STIME(T) : SIMULATION TIME AT T-
TH TIME STEP
C  DENS : DENSITY OF WATER
C  VISCO : DYNAMIC VISCOSITY OF
WATER
C  ELEV : ELEVATION OF ORIGIN
C  IPRINT1 : PRINTING OPTION. PUT 0
OR 1. 0 : NO PRINTING OUT HYDRAULIC
HEADS
c  (PRESSURE) AND FLUX AT ALL CELLS,
1 : PRINTING OUT THEM BY TIME
INTERVAL OF IPTIT.
C  IPTIT :IN CASE IPTIT IS 1, PUT
EVERY TIME STEP INTERVAL TO PRINT
OUT.
C       EX) IPTIT = 100, OUTPUT DATA
FILE PRINTS OUT HYDRAULIC HEADS AND
FLUX BY
C                   TIME STEP
INTERVAL OF 100
C  IPRINT2 : PRINTING OPTION. PUT 0
OR 1. 0 : NO PRINTING OUT HYDRAULIC
HEAD OR FLUX
C        AT SPECIFIC CELLS BY
INTERVAL OF TIME STEP SIZE DT. 1 :
PRINTING OUT
C        HYDRAULIC HEAD OR FLUX AT
SPECIFIC CELLS BY INTERVAL OF TIME
STEP SIZE DT
C  NCPH : NUMBER OF CELLS TO PRINT
OUT HYDRAULIC HEAD BY DT
C  NPH : CELL NUMBER TO PRINT OUT
HYDRAULIC HEAD. NPXH(NCPH):X-
COORDINATE NUMBER,
C        NPYH(NCPH):Y-COORDINATE
NUMBER, NPZH(NCPH):Z-COORDINATE
NUMBER.
C        THIS SAVES X,Y,Z-COORDINATE
NUMBERS TO PRINT OUT
C       EX) IF NCPH : 2 AND NPH :
2,3,5  7,4,5,
NPXH(1)=2,NPYH(1)=3,NPZH(1)=5,
C
NPXH(2)=7,NPYH(2)=4,NPZH(2)=5
C  NCPQ : NUMBER OF CELLS TO PRINT
OUT FLUX BY DT
C  NPQ : CELL NUMBER TO PRINT OUT
FLUX.
NPXQ(NCPQ),NPYQ(NCPQ),NPZQ(NCPQ).
C        IN SAME WAY WITH NPH
C  DX(I) : X-DIRECTIONAL LENGTH OF
I-TH CELL
C  DY(J) : Y-DIRECTIONAL LENGTH OF
J-TH CELL
C  DZ(K) : Z-DIRECTIONAL LENGTH OF
K-TH CELL
C  X(I) : X-DIRECTIONAL DISTANCE OF
NODE(I,J,K) FROM ORIGIN
C  Y(J) : Y-DIRECTIONAL DISTANCE OF
NODE(I,J,K) FROM ORIGIN
```

```
C  Z(K) : Z-DIRECTIONAL DISTANCE OF
NODE(I,J,K) FROM ORIGIN
C  KX(I,J,K) : X-DIRECTIONAL
HYDRAULIC CONDUCTIVITY AT
NODE(I,J,K)(UNIT:M/S)
C  KY(I,J,K) : Y-DIRECTIONAL
HYDRAULIC CONDUCTIVITY AT
NODE(I,J,K)(UNIT:M/S)
C  KZ(I,J,K) : Z-DIRECTIONAL
HYDRAULIC CONDUCTIVITY AT
NODE(I,J,K)(UNIT:M/S),
C        VERTICAL HYDRAULIC
CONDUCTIVITY FOR SOME CELLS
INCLUDING LEAKAGE SECTION
C        (ABANDONED WELL) MUST BE
USED TO KZBLEAK(I,J,K)
C  PORO(I,J,K) : POROSITY OF EACH
CELL
C  Ss(I,J,K) : SPECIFIC STORAGE
COEFFICIENT, UNIT:1/M
C  NBC(I,J,K) : NUMBER OF BOUNDARY
CONDITION OF EACH CELL. 0:NORMAL
CELL,
C        1:CONSTANT HEAD BOUNDARY
CONDITION, 2:NO FLOW BOUNDARY
CONDITION,
C        3:SINK/SOURCE TERM,
4:LEAKAGE POINTS AT UPPER AQUIFER,
5:LEAKAGE POINTS AT
C        INJECTION AQUIFER,. Even if
THERE ARE MULTI LEAKAGE POINTS, USE
JUST 4 and 5.
C  KZBLEAK(I,J,K) : EFFECTIVE
VERTICAL HYDRAULIC CONDUCTIVITY IN
AND AROUND LEAKAGE
C        SECTION (ABANDONED WELL)
BETWEEN LEAKAGE OUTFLOW POINT AT
INJECTION AQUIFER
C        AND LEAKAGE INFLOW POINT AT
UPPER AQUIFER. JUST FOR NBC(I,J,K)=4
and 5 TO
C        GET LEAKAGE RATE AT
CELL(I,J,K). NO CELLS IN LEAKAGE
SECTIONS (NORMAL CELLS)
C        HAVE 0(ZERO).
C  ALEAK(I,J,K) : EFFECTIVE LEAKAGE
AREA. IN GENERAL, CROSS SECTIONAL
AREA OF
C        ABANDONED WELL IS USED FOR
ALEAK(I,J,K). JUST FOR NBC(I,J,K)=4
and 5.
C  LSTIME(I,J,K) : LEAKAGE STARTING
TIME. IF NOL IS 1, LSTIME MEANS
STARTING TIME
C        OF INGOING LEAKAGE INTO
LEAKAGE PATHWAY. IF NOL IS 2, LSTIME
MEANS STARTING
C        TIME OF OUTGOING LEAKAGE
FROM LEAKAGE PATHWAY.
C  NOL : LEAKAGE OPTION NUMBER. IF
NOL IS 1, LEAKAGE TRAVEL TIME IS
CONSIDER BECAUSE
C        LSTIME MEANS LEAKAGE
INGOING TIME INTO LEAKAGE PATHWAY,
SO LEAKAGE OUTGOING
C        TIME IS SUM OF LSTIME AND
LEAKAGE TRAVEL TIME. IF NOL IS 2,
LEAKAGE TRAVEL
C        TIME IS NOT CONSIDER
BECAUSE LSTIME MEANS THAT LEAKAGE
OUTGOING TIME INTO
C        UPPER AQUIFERS FROM LEAKAGE
PATHWAY. THAT IS, LEAKAGE TRAVEL
TIME IN THE
C        LEAKAGE PATHWAY IS IGNORED
BECAUSE THIS AMOUNT IN THE LEAKAGE
PATHWAY IS
C        SO LITTLE AND ACTUAL
LEAKAGE TRAVEL TIME CAN HAVE A LOT
OF UNCERTAINTIES
C        TO BE CALCULATED.
C  OLST(I) : ONE-DIMENSIONAL LEAKAGE
STARTING TIME, LEAKAGE TIME OF ITH
NODE
C  XLI(I) : X-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER
C  YLI(I) : Y-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER
C  ZLI(I) : Z-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER
C  ZLIU(I) : Z-COORDINATE OF ITH
LEAKAGE POINT AT UPPER AQUIFER
C        XLUA(I)=XLI(I) AND
YLUA(I)=YLI(I), SO XLUA AND YLUA ARE
NOT USED
C  XLS(I) : X-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER.
IN CASE OF
C        OCCURING OF LEAKAGE AT
SIMULATION STARTING TIME (ST), I IS
FROM 1 TO NCST
C        AND I IS FROM NCST+1 TO NLC
FOR LEAKAGE GENERATED AFTER ST
C  YLS(I) : Y-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER.
IN CASE OF
C        OCCURING OF LEAKAGE AT
SIMULATION STARTING TIME (ST), I IS
FROM 1 TO NCST
```

```
C        AND I IS FROM NCST+1 TO NLC
FOR LEAKAGE GENERATED AFTER ST
C  ZLS(I) : Z-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER.
IN CASE OF
C        OCCURING OF LEAKAGE AT
SIMULATION STARTING TIME (ST), I IS
FROM 1 TO NCST
C        AND I IS FROM NCST+1 TO NLC
FOR LEAKAGE GENERATED AFTER ST
C  ZLSU(I) : Z-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER.
IN CASE OF
C        OCCURING OF LEAKAGE AT
SIMULATION STARTING TIME (ST), I IS
FROM 1 TO NCST
C        AND I IS FROM NCST+1 TO NLC
FOR LEAKAGE GENERATED AFTER ST
C  XCL(I) : X-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER.
IN CASE OF
C        OCCURING OF LEAKAGE AT
SIMULATION STARTING TIME (ST), I IS
FROM 1 TO NCST
C        AND I IS ASCENDING TIME
ORDER FOR LEAKAGE GENERATED AFTER ST
C  YCL(I) : Y-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER.
IN CASE OF
C        OCCURING OF LEAKAGE AT
SIMULATION STARTING TIME (ST), I IS
FROM 1 TO NCST
C        AND I IS ASCENDING TIME
ORDER FOR LEAKAGE GENERATED AFTER ST
C  ZCL(I) : Z-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER.
IN CASE OF
C        OCCURING OF LEAKAGE AT
SIMULATION STARTING TIME (ST), I IS
FROM 1 TO NCST
C        AND I IS ASCENDING TIME
ORDER FOR LEAKAGE GENERATED AFTER ST
C  ZCLU(I) : Z-COORDINATE OF ITH
LEAKAGE POINT AT INJECTION AQUIFER.
IN CASE OF
C        OCCURING OF LEAKAGE AT
SIMULATION STARTING TIME (ST), I IS
FROM 1 TO NCST
C        AND I IS FROM NCST+1 TO NLC
FOR LEAKAGE GENERATED AFTER ST
C  APTOLK(I) : AT ITH LEAKAGE POINT,
LEAKAGE APPROACHING TIME INTO SOME
CELLS AT
C        UPPER AQUIFER
C  NCHB : TOTAL NUMBER OF CELLS WITH
CONSTANT HEAD BOUNDARY CONDITION
C  NTB1 : NUMBER OF TIME STEP FOR
CONSTANT HEAD BOUNDARY CONDITIONS
C  NXHB(I) : X-DIRECTIONAL BOUNDARY
NUMBER
C  NYHB(J) : Y-DIRECTIONAL BOUNDARY
NUMBER
C  NZHB(K) : Z-DIRECTIONAL BOUNDARY
NUMBER
C  TSBC1(I) : TIME SERIES FOR
CONSTANT HEAD BOUNDARY CONDITION
C  BHEAD(I,J) : CONSTANT HEAD
BOUNDARY CONDITION,I:TIME STEP,J:Jth
HEAD AT Ith ROW
C  NSST : TOTAL NUMBER OF CELLS WITH
SINK/SOURCE TERM EXCEPT FOR LEAKAGE
C  NTB2 : NUMBER OF TIME STEP FOR
SINK/SOURCE BOUNDARY CONDITION
C  NXSS(I) : X-DIRECTIONAL
SINK/SOURCE TERM NUMBER
C  NYSS(J) : Y-DIRECTIONAL
SINK/SOURCE TERM NUMBER
C  NZSS(K) : Z-DIRECTIONAL
SINK/SOURCE TERM NUMBER
C  TSBC2(I) :TIME SERIES FOR
SINK/SOURCE CONDITION
C  BFSS(I,J) : FLUX(M**3/S) FOR
SINK/SOURCE CELLS,I:TIME STEP,J:Jth
FLUX AT Ith ROW
C  BFTI(I,J) : IN GOVERNING EQ, UNIT
IS T**(-1), SO BFSS(M**3/S) HAS TO
CONVERT TO
C        BFTI(1/S).
BFTI=BFSS/(DX*DY*DZ)), I:TIME
STEP,J:Jth FLUX
C  W(I,J,K) : SINK/SOURCE TERM AT
CELL(I,J,K). UNIT:1/T,
C
W(NXHB(J),NYHB(J),NZHB(J))=BFTI(I,J)
C  CZBLEAK(I,J,K) : VERTICAL
CONDUCTANCE AT VERTICAL LEAKAGE
PATHWAY. I,J,K MEANS
C        LEAKAGE POINT AT INJECTION
AQUIFER, II MEANS II-TH LEAKAGE
PATHWAY
C
(=KZBLEAK(I,J,K)*ALEAK(I,J,K)/(DX(I)
*DY(J)*DZ(K)*DZBLEAK(II)))
C  DZBLEAK(I) : DISTANCE OF iTH
VERTICAL LEAKAGE PATHWAY. IF
IMPERMEABLE LAYER EXISTS
C        AMONG K, K-1 AND K-2
LAYERS, iTH
```

```
DZBLEAK(I)=(DZ(K)/2+DZ(K-1)+DZ(K-
2)/2
C  NCST : NUMBER OF COUNTING LEAKAGE
POINTS WITH LEAKAGE AT SIMULATION
STARTING TIME
C  NCNS : NUMBER OF COUNTING LEAKAGE
POINTS WITH NO LEAKAGE AT SIMULATION
ST. IT COUNTS
C        FROM NCST+1 TO NLC
C  IT : ITERATION NUMBER IN ONE TIME
STEP TO BE CONVERGENT
C  NIT : COUNTING NUMBER FOR TIME
STEP
C  HEAD(I,J,K) : HYDRAULIC HEAD AT
NODE(I,J,K)
C  PSHEAD(I,J,K) : PREVIOUS STEP
HYDRAULIC HEAD AT NODE(I,J,K)
C  WW: OMEGA OF SOR METHOD
C  DAZL(I) : FLOWING LENGTH OF ITH
LEAKAGE DURING THE DT IN LEAKAGE
PATHWAY
C  AZL(I) : FLOWING LENGTH OF ITH
LEAKAGE IN LEAKAGE PATHWAY. =
DZBLEAK(I)
C  VOLK(I) : VELOCITY OF ITH LEAKAGE
IN LEAKAGE PATHWAY
C  VXF(I,J,K) : X-DIRECTIONAL
VELOCITY OF INFLOW INTO CELL(I,J,K)
C  VXB(I,J,K) : X-DIRECTIONAL
VELOCITY OF OUTFLOW FROM CELL(I,J,K)
C  VYF(I,J,K) : Y-DIRECTIONAL
VELOCITY OF INFLOW INTO CELL(I,J,K)
C  VYB(I,J,K) : Y-DIRECTIONAL
VELOCITY OF OUTFLOW FROM CELL(I,J,K)
C  VZF(I,J,K) : Z-DIRECTIONAL
VELOCITY OF INFLOW INTO CELL(I,J,K)
C  VZB(I,J,K) : Z-DIRECTIONAL
VELOCITY OF OUTFLOW FROM CELL(I,J,K)
C  QXF(I,J,K) : X-DIRECTIONAL FLUX
OF INFLOW INTO CELL(I,J,K)
C  QXB(I,J,K) : X-DIRECTIONAL FLUX
OF OUTFLOW FROM CELL(I,J,K)
C  QYF(I,J,K) : Y-DIRECTIONAL FLUX
OF INFLOW INTO CELL(I,J,K)
C  QYB(I,J,K) : Y-DIRECTIONAL FLUX
OF OUTFLOW FROM CELL(I,J,K)
C  QZF(I,J,K) : Z-DIRECTIONAL FLUX
OF INFLOW INTO CELL(I,J,K)
C  QZB(I,J,K) : Z-DIRECTIONAL FLUX
OF OUTFLOW FROM CELL(I,J,K)
C  QLEAK(I,J,K) : LEAKAGE RATE AT
CELL(I,J,K)
C  VKAOL(NPARMAX) : RANDOM VALUES
FROM GA,
C                  THIS MEANS
VERTICAL KZBLEAK*ALEAK OF LEAKAGE
PATHWAYS
C  TMLEAK(NPARMAX) : RANDOM VALUES
FROM GA, THIS MEANS LSTIME
C  HDCN(NPARMAX) : RANDOM VALUES
FROM GA, THIS MEANS HYDRAULIC
C                  CONDUCTIVITIES OF
EACH GROUPED NORMAL CELLS IN DOMAIN
C  AKL(I,J,K) : IT IS EQUAL TO
VKAOL, i.e., KZBLEAK*ALEAK

C      PARAMETER (NNODES=20,
NTIMES=50)
      INCLUDE 'PARAMS.F'

C      CHARACTER*40 FNAME1, FNAME2
      CHARACTER*120 TITLE, GROUP1,
GROUP2, GROUP3, GROUP4, GROUP5
      CHARACTER*120 DEF1,DEF4
C     CHARACTER*120 DEF3
      CHARACTER*180 DEF2
      COMMON/GWM2/ NMD,NST,WEFR
       COMMON/GWM1/NEL,NXM,NYM,NZM
       COMMON/GWM3/NSM
       COMMON/GWM4/ELTIME
       COMMON/GWM5/XEL,YEL,ZEL,ZELU
       COMMON/GWM6/NPOPULA,NGENERA
       COMMON/GWM7/XOG,YOG,ZOG,NPG,N
TG,NOG
      COMMON/FGW/ NFI
      COMMON/GRI/GRD(NTIMES)

      INTEGER
XEL(NXNODE),YEL(NYNODE),ZEL(NZNODE),
ZELU(NZNODE)
      INTEGER
XOG(NXNODE,NEHCG),YOG(NYNODE,NEHCG),
ZOG(NZNODE,NEHCG)
      DIMENSION
NXM(NXNODE),NYM(NYNODE),NZM(NZNODE)
      DIMENSION
HEADCAL(NTIMES,NNODES)
      DIMENSION
HEADUPP(NTIMES,NNODES)
      DIMENSION
HEADINJ(NTIMES,NNODES)
      DIMENSION
VKAOL(NPARMAX),TMLEAK(NPARMAX),HDCN(
NPARMAX)
      DIMENSION
AKL(NXNODE,NYNODE,NZNODE)
      DIMENSION
QLUPP(NTIMES,NNODES),QLINJ(NTIMES,NN
ODES)
```

```
      DIMENSION
DX(NXNODE),DY(NYNODE),DZ(NZNODE)
      DIMENSION
NXC(NXNODE),NYC(NYNODE),NZC(NZNODE)
      DIMENSION
PORO(NXNODE,NYNODE,NZNODE),SS(NXNODE
,NYNODE,NZNODE)
      DIMENSION
NBC(NXNODE,NYNODE,NZNODE)
      DIMENSION
ALEAK(NXNODE,NYNODE,NZNODE)
      DIMENSION OLST(NNODES*NNODES)
      DIMENSION
STIME(NTIMES),X(NXNODE),Y(NYNODE),Z(
NZNODE)
      DIMENSION
W(NXNODE,NYNODE,NZNODE)
      DIMENSION
DXF(NXNODE),DXB(NXNODE),DYF(NYNODE),
DYB(NYNODE)
      DIMENSION
DZF(NZNODE),DZB(NZNODE)
      DIMENSION
CXF(NXNODE,NYNODE,NZNODE),CXB(NXNODE
,NYNODE,NZNODE)
      DIMENSION
CYF(NXNODE,NYNODE,NZNODE),CYB(NXNODE
,NYNODE,NZNODE)
      DIMENSION
CZF(NXNODE,NYNODE,NZNODE),CZB(NXNODE
,NYNODE,NZNODE)
      DIMENSION
CZBLEAK(NXNODE,NYNODE,NZNODE)
      DIMENSION DZBLEAK(NNODES)
      DIMENSION
PSHEAD(NXNODE,NYNODE,NZNODE)
      DIMENSION
HYH(NXNODE,NYNODE,NZNODE)
      DIMENSION
HEAD(NXNODE,NYNODE,NZNODE)
      DIMENSION
VXF(NXNODE,NYNODE,NZNODE)
      DIMENSION
VXB(NXNODE,NYNODE,NZNODE)
      DIMENSION
VYF(NXNODE,NYNODE,NZNODE)
      DIMENSION
VYB(NXNODE,NYNODE,NZNODE)
      DIMENSION
VZF(NXNODE,NYNODE,NZNODE)
      DIMENSION
VZB(NXNODE,NYNODE,NZNODE)
      DIMENSION
QXF(NXNODE,NYNODE,NZNODE)
      DIMENSION
QXB(NXNODE,NYNODE,NZNODE)
      DIMENSION
QYF(NXNODE,NYNODE,NZNODE)
      DIMENSION
QYB(NXNODE,NYNODE,NZNODE)
      DIMENSION
QZF(NXNODE,NYNODE,NZNODE)
      DIMENSION
QZB(NXNODE,NYNODE,NZNODE)
      DIMENSION
QLEAK(NXNODE,NYNODE,NZNODE)
      DIMENSION
NPXH(NXNODE),NPYH(NYNODE),NPZH(NZNOD
E)
      DIMENSION
NPXQ(NXNODE),NPYQ(NYNODE),NPZQ(NZNOD
E)
      DIMENSION NXHB(NNODES*NNODES)
      DIMENSION NYHB(NNODES*NNODES)
      DIMENSION NZHB(NNODES*NNODES)
      DIMENSION
TSBC1(NTIMES),BHEAD(NTIMES,NNODES*NN
ODES)
      DIMENSION NXSS(NNODES*NNODES)
      DIMENSION NYSS(NNODES*NNODES)
      DIMENSION NZSS(NNODES*NNODES)
      DIMENSION
TSBC2(NTIMES),BFSS(NTIMES,NNODES*NNO
DES)
      DIMENSION
BFTI(NTIMES,NNODES*NNODES)
      DIMENSION TLS(NNODES*NNODES)
      DIMENSION
SUM1(NNODES*NNODES),SUM2(NNODES*NNOD
ES)
      DIMENSION
APTOLK(NNODES*NNODES)
      DIMENSION HPO(NTIMES,NNODES)
      DIMENSION QPXF(NTIMES,NXNODE)
      DIMENSION QPXB(NTIMES,NXNODE)
      DIMENSION QPYF(NTIMES,NYNODE)
      DIMENSION QPYB(NTIMES,NYNODE)
      DIMENSION QPZF(NTIMES,NZNODE)
      DIMENSION QPZB(NTIMES,NZNODE)
      DIMENSION
NPNLA2(NNODES*NNODES)
      DIMENSION NPG(NNODES)
      DIMENSION WEFR(NNODES)

      REAL
KX(NXNODE,NYNODE,NZNODE),KY(NXNODE,N
YNODE,NZNODE)
      REAL KZ(NXNODE,NYNODE,NZNODE)
```

```
      REAL
KZBLEAK(NXNODE,NYNODE,NZNODE)
      REAL
LSTIME(NXNODE,NYNODE,NZNODE)
C      REAL
KXF(NXNODE,NYNODE,NZNODE),KXB(NXNODE
,NYNODE,NZNODE)
C      REAL
KYF(NXNODE,NYNODE,NZNODE),KYB(NXNODE
,NYNODE,NZNODE)
C      REAL
KZF(NXNODE,NYNODE,NZNODE),KZB(NXNODE
,NYNODE,NZNODE)
      REAL MU1(NXNODE,NYNODE,NZNODE)
      REAL MU2(NXNODE,NYNODE,NZNODE)

C      INTEGER H
      INTEGER
XLI(NXNODE),YLI(NYNODE),ZLI(NZNODE),
ZLIU(NZNODE)
      INTEGER
XLS(NXNODE),YLS(NYNODE),ZLS(NZNODE),
ZLSU(NZNODE)
      INTEGER
XCL(NXNODE),YCL(NYNODE),ZCL(NZNODE),
ZCLU(NZNODE)
      INTEGER
XAT(NXNODE),YAT(NYNODE),ZAT(NZNODE),
ZATU(NZNODE)

C     WRITE(*,'(/A)') ' WHAT IS THE
NAME OF INPUT DATA FILE ?'
C      READ(*,'(A)') FNAME1
C      OPEN(4, FILE=FNAME1,
STATUS='OLD')

C      WRITE(*,'(/A)') ' WHAT IS THE
NAME OF OUTPUT FILE ?'
C      READ(*,'(A)') FNAME2
C      OPEN(5, FILE=FNAME2,
STATUS='UNKNOWN')

C ------------------- CONSTITUTION
OF FORWARD INPUT DATA --------------
---------
C      ! SKIP FORWARD INPUT DATA
FROM 2TH INVERSE ITERATION
C      IF(NFI.EQ.2)THEN
C        DO I=1,NGENERA
C          NCSM=1+(I-1)*NPOPULA
C          IF(NCSM.EQ.NSM) GOTO 101
C        ENDDO
C         GOTO 1001
C      ENDIF
C

 101   READ(4,'(A80)') TITLE
c      WRITE(5,'(A)') TITLE

C ----------------- CONTROL
PARAMETERS
      READ(4,'(A80)') GROUP1
      READ(4,*) NSIMCON, NX, NY, NH,
ET, DT, NLC, TOL
      READ(4,'(A80)') GROUP2
      READ(4,*) DENS, VISCO, ELEV

C ----------------- PRINT OPTIONS
      READ(4,'(A80)') GROUP3
      READ(4,*) IPRINT1, IPTIT
      READ(4,'(A120)') DEF1
      READ(4,*) IPRINT2, NCPH,
(NPXH(I),NPYH(I),NPZH(I),I=1,NCPH),
     >           NCPQ,
(NPXQ(J),NPYQ(J),NPZQ(J),J=1,NCPQ)

C ----------------- CELL INFORMATION
      READ(4,'(A80)') GROUP4
      READ(4,'(A180)') DEF2
      NL1=0
      NL2=0
      NL3=0
      NL4CB=0
      DO I=1,NX
        DO J=1,NY
          DO K=1,NH
      READ(4,*)
NXC(I),NYC(J),NZC(K),DX(I),DY(J),DZ(
K),HYH(I,J,K),

>KX(I,J,K),KY(I,J,K),KZ(I,J,K),PORO(
I,J,K),SS(I,J,K),NBC(I,J,K),

>KZBLEAK(I,J,K),ALEAK(I,J,K),LSTIME(
I,J,K)

       IF(NBC(I,J,K).EQ.4) THEN
         NL1=NL1+1
C         XLI(NL1)=I  ! X-COORDINATE
(SAME X-COORDINATE AT NBC=5)
C         YLI(NL1)=J  ! Y-COORDINATE
(SAME Y-COORDINATE AT NBC=5)
         ZLIU(NL1)=K  ! Z-COORDINATE
OF LEAKAGE INFLOW LOCATION
       ENDIF
       IF(NBC(I,J,K).EQ.5) THEN
         NL2=NL2+1
         OLST(NL2)=LSTIME(I,J,K) !
CONVERT LEAKAGE STARTING TIME
         XLI(NL2)=I  ! X-COORDINATE
OF LEAKAGE OUTFLOW LOCATION
```

```
         YLI(NL2)=J  ! Y-COORDINATE
OF LEAKAGE OUTFLOW LOCATION
         ZLI(NL2)=K  ! Z-COORDINATE
OF LEAKAGE OUTFLOW LOCATION
       ENDIF
       IF(NBC(I,J,K).EQ.2) THEN
         KX(I,J,K)=0.0
         KY(I,J,K)=0.0
         KZ(I,J,K)=0.0
         PORO(I,J,K)=0.0
         SS(I,J,K)=0.0
       ENDIF
       IF(LSTIME(I,J,K).NE.0.0 .AND.
NLC.NE.0) THEN
         NL3=NL3+1
        ENDIF
C V2.5  COUNTING CONSTANT HEAD
BOUNDARY CELLS
       IF(NBC(I,J,K).EQ.1) THEN
        NL4CB=NL4CB+1
       ENDIF
C V2.5
          ENDDO
         ENDDO
       ENDDO
       IF
((NL1.NE.NLC).OR.(NL2.NE.NLC)) THEN
      WRITE(*,*) ' WARNING! NUMBER
OF NBC (4) AND NBC (5), AND NLC FOR
     >LEAKAGE FEATURES MUST BE SAME.
CHECK UP LEAKAGE FEATURES IN INPUT
     >DATA !! '
      WRITE(5,*) ' WARNING! NUMBER
OF NBC (4) AND NBC (5), AND NLC
     >FOR LEAKAGE FEATURES MUST BE
SAME. CHECK UP LEAKAGE FEATURES IN
     >INPUT DATA !! '
      STOP
      ENDIF
C   OPTION OF LEAKAGE FEATURES
      IF (NL3.NE.0 .AND. NFI.EQ.1)
THEN
 11   WRITE(*,'(/A)') ' WOULD YOU
LIKE TO CONSIDER LEAKAGE TRAVEL TIME
'
      WRITE(*,'(A)') ' THROUGH
LEAKAGE PATHWAY OR NOT? '
       WRITE(*,'(A)') ' 1. YES,
LEAKAGE STARTING TIME IN INPUT WILL
BE '
      WRITE(*,'(A)') '    LEAKAGE
INGOING TIME INTO LEAKAGE PATHWAY '
       WRITE(*,'(A)') ' 2. NO,
LEAKAGE STARTING TIME IN INPUT WILL
BE '
       WRITE(*,'(A)') '    LEAKAGE
OUTGOING TIME FROM LEAKAGE PATHWAY '
       WRITE(*,'(A)') '
(RECOMMENDED)'
      READ(*,'(I2)') NOL
        IF
((NOL.NE.1).AND.(NOL.NE.2)) THEN
          WRITE (*,*) 'YOU MUST PUT
1 OR 2 IN.'
          GOTO 11
         ENDIF
      ELSEIF (ELTIME.NE.0.0 .AND.
NFI.EQ.2) THEN
        NOL=2 ! IN CASE OF INVERSE,
SIMULATOR CONSIDERS THAT LEAKAGE
      ENDIF   ! TRAVEL TIME IS ZERO

C ---------------- CONSTANT HEAD
BOUNDARY CONDITION
C V2.5 I CONSIDER CONSTANT HEAD
BOUNDARY IS NOT CHANGE WITH TIME.
C  SO, IN VERSION 2.5 INITIAL HEADS
ARE KEPT CONSTANTLY WITH TIME
C  AT CONSTANT BOUNDARY. SO IT IS
REVISED.
C START OLD VERSION
C      READ(4,'(A80)') GROUP5
C      READ(4,'(A120)') DEF3
C      READ(4,*) NCHB, NTB1
C      IF (NCHB.EQ.0) GOTO 66
C      READ(4,*)
(NXHB(I),NYHB(I),NZHB(I),I=1,NCHB)
C      DO I=1,NTB1
C        READ(4,*) TSBC1(I),
(BHEAD(I,J),J=1,NCHB)
C         IF (I.EQ.NTB1)
THL=TSBC1(I)
C      ENDDO
C      IF (NTB1.NE.0 .AND.
THL.LT.ET) THEN
C       WRITE(*,*) ' LAST TIME OF
BOUNDARY CONDITION IS LESS THAN
ENDING
C     >TIME OF SIMULATION '
C       WRITE(5,*) ' LAST TIME OF
BOUNDARY CONDITION IS LESS THAN
ENDING
C     >TIME OF SIMULATION '
C       STOP
C      ENDIF
C END OLD VERSION
C START VERSION 2.5
       READ(4,'(A80)') GROUP5
       NCHB=NL4CB
       NTB1=1
```

```
       TSBC1(NTB1)=ET
       NL5CB=0
       IF (NCHB.EQ.0) GOTO 66
       DO I=1,NX
        DO J=1,NY
          DO K=1,NH
          IF(NBC(I,J,K).EQ.1) THEN
            NL5CB=NL5CB+1
            NXHB(NL5CB)=I
            NYHB(NL5CB)=J
            NZHB(NL5CB)=K

BHEAD(NTB1,NL5CB)=HYH(I,J,K)
          ENDIF
          ENDDO
        ENDDO
       ENDDO
C       WRITE(*,*) ' CONSTANT
BOUNDARY = ', NL5CB
       IF (NCHB.NE.NL5CB) THEN
       WRITE(*,*) ' PLEASE CHECK!!
CONSTANT HEAD BOUNDARY CONDITION '
       WRITE(5,*) ' PLEASE CHECK!!
CONSTANT HEAD BOUNDARY CONDITION '
       STOP
      ENDIF
C END VERSION 2.5

C -------------- FLUX OF
SINK/SOURCE(EXCEPT FOR LEAKAGE)
 66   READ(4,'(A120)') DEF4
      READ(4,*) NSST, NTB2
      IF (NSST.EQ.0) GOTO 1001
      READ(4,*)
(NXSS(I),NYSS(I),NZSS(I),I=1,NSST)
      DO I=1,NTB2
        READ(4,*) TSBC2(I),
(BFSS(I,J),J=1,NSST)
        IF (I.EQ.NTB2) TFL=TSBC2(I)
C V2.7.1
        IF (NFI.EQ.2) THEN
         IF (I.GE.2) THEN
C          TFL=TSBC2(I)
          BFSS(I,NSST)=GRD(1)
         ENDIF
        ENDIF
C V2.7.1
      ENDDO
      IF (NTB2.NE.0 .AND. TFL.LT.ET)
THEN
       WRITE(*,*) ' LAST TIME OF
BOUNDARY CONDITION IS LESS THAN
ENDING
     >TIME OF SIMULATION '
       WRITE(5,*) ' LAST TIME OF
BOUNDARY CONDITION IS LESS THAN
ENDING
     >TIME OF SIMULATION '
       STOP
      ENDIF
C CHANGE UNIT(M**3/S) OF FLUX OF
SINK/SOURCE BOUNDARY CONDITION TO
1/S.
C UNIT OF BFSS (INFLOW(+), OUTFLOW(-
)) IS L**3T**(-1). UNIT OF BFTI IS
C T**(-1). IN GOVERNING EQ, UNIT IS
T**(-1), SO BFSS HAS TO CONVERT TO
C BFTI.
      DO I=1,NTB2
        DO J=1,NSST

BFTI(I,J)=BFSS(I,J)/(DX(NXSS(J))*DY(
NYSS(J))*DZ(NZSS(J)))
        ENDDO
      ENDDO
      REWIND(4)
C ----------------------- END OF
INPUT DATA -------------------------

C --------------------- REVISE INPUT
FOR INVERSE ANALYSIS
1001  IF (NFI.EQ.2) THEN
       NLC=NEL
       NOF=0
      DO I=1,NX
        DO J=1,NY
          DO K=1,NH
       IF(NBC(I,J,K).EQ.4) THEN
         NBC(I,J,K)=0
         KZBLEAK(I,J,K)=0.
         ALEAK(I,J,K)=0.
         LSTIME(I,J,K)=0.
       ENDIF
       IF(NBC(I,J,K).EQ.5) THEN
         NBC(I,J,K)=0
         KZBLEAK(I,J,K)=0.
         ALEAK(I,J,K)=0.
         LSTIME(I,J,K)=0.
       ENDIF
       IF(NBC(I,J,K).NE.2) THEN
C         KX(I,J,K)=0.0
C         KY(I,J,K)=0.0
C         KZ(I,J,K)=0.0
         NOF=NOF+1  ! COUNTING CELLS
EXCEPT FOR FLOW BOUNDARY CELLS
       ENDIF
          ENDDO
        ENDDO
      ENDDO
```

```
C NTG: TOTAL NUMBER OF CELLS FOR
GROUPING HYDRAULIC CONDUCTIVITY OF
NORMAL CELLS
      IF
(NOF.NE.NTG.AND.NSM.EQ.1)THEN
      WRITE(*,*)'NOTICE!! TOTAL
NUMBER OF ELEMENTS IN EACH GROUP OF
HYDR
     >AULIC HEAD IN MEASUREMENT
INPUT IS DIFFERENT WITH TOTAL NUMBER
OF
     >NORMAL CELLS, THE HYDRAULIC
CONDUCTIVITIES OF NO SPECIFIED CELLS
A
     >RE USED FROM THOSE IN FORWARD
INPUT FILE '
      ENDIF

      DO K=1,NLC
         OLST(K)=0.0 ! CONVERT
LEAKAGE STARTING TIME
         XLI(K)=0  ! X-COORDINATE OF
LEAKAGE OUTFLOW LOCATION
         YLI(K)=0  ! Y-COORDINATE OF
LEAKAGE OUTFLOW LOCATION
         ZLI(K)=0  ! Z-COORDINATE OF
LEAKAGE OUTFLOW LOCATION
         ZLIU(K)=0  ! Z-COORDINATE
OF LEAKAGE INFLOW LOCATION
      ENDDO

C  DEFINE KBLEAK*ALEAK(AKL) OF
EXPECTED LEAKAGE PATHWAYS AND
HYDRAULIC CONDUCTIVITY
C  OF EACH NODES BELONG TO GROUPS
      DO I=1,NX
        DO J=1,NY
          DO K=1,NH
            DO IL=1,NLC

IF(I.EQ.XEL(IL).AND.J.EQ.YEL(IL).AND
.K.EQ.ZELU(IL))THEN
        NBC(I,J,K)=4
        AKL(I,J,K)=VKAOL(IL)
C        LSTIME(I,J,K)=TMLEAK(IL)
        ZLIU(IL)=K

ELSEIF(I.EQ.XEL(IL).AND.J.EQ.YEL(IL)
.AND.K.EQ.ZEL(IL))THEN
        NBC(I,J,K)=5
        AKL(I,J,K)=VKAOL(IL)
        LSTIME(I,J,K)=TMLEAK(IL)
        OLST(IL)=TMLEAK(IL) !
CONVERT LEAKAGE STARTING TIME
        XLI(IL)=I  ! X-COORDINATE OF
LEAKAGE OUTFLOW LOCATION
        YLI(IL)=J  ! Y-COORDINATE OF
LEAKAGE OUTFLOW LOCATION
        ZLI(IL)=K  ! Z-COORDINATE OF
LEAKAGE OUTFLOW LOCATION
      ENDIF
            ENDDO

            DO IG=1,NOG
              DO IC=1,NPG(IG)

IF(I.EQ.XOG(IG,IC).AND.J.EQ.YOG(IG,I
C).AND.K.EQ.ZOG(IG,IC))THEN
C        KX(I,J,K)=HDCN(2*IG-1) ! X
AND Y-HYDRAULIC CONDUCTIVITY ARE
SAME
C        KY(I,J,K)=HDCN(2*IG-1)
C        KZ(I,J,K)=HDCN(2*IG) ! BUT
Z-HYDRAULIC CONDUCTIVITY IS
DIFFERENT
        KX(I,J,K)=HDCN(IG) ! X,Y AND
Z-HYD. CON. ARE THE SAME
        KY(I,J,K)=HDCN(IG)
        KZ(I,J,K)=HDCN(IG)
      ENDIF
              ENDDO
            ENDDO
          ENDDO
        ENDDO
      ENDDO

C V.2.7.1        IF
(NSM.GT.(NGENERA-1)*NPOPULA)THEN
C V.2.7.1           WRITE(5,*)
C V.2.7.1
WRITE(5,'(/,I10)') NSM
C V.2.7.1        ENDIF
      ENDIF
C ---------------- END OF REVISE
INPUT FOR INVERSE ANALYSIS

C  DEFINITION OF
DXF,DXB,DYF,DYB,DZF,DZB
      DO I=2,NX
       DXF(I)=DX(I-1)/2.+DX(I)/2.
      ENDDO
      DO I=1,NX-1
       DXB(I)=DX(I)/2.+DX(I+1)/2.
      ENDDO
      DO I=2,NY
       DYF(I)=DY(I-1)/2.+DY(I)/2.
      ENDDO
      DO I=1,NY-1
       DYB(I)=DY(I)/2.+DY(I+1)/2.
```

```
      ENDDO
      DO I=2,NH
       DZF(I)=DZ(I-1)/2.+DZ(I)/2.
      ENDDO
      DO I=1,NH-1
       DZB(I)=DZ(I)/2.+DZ(I+1)/2.
      ENDDO

C  DEFINITION OF
KXF,KXB,KYF,KYB,KZF,KZB AND
CXF,CXB,CYF,CYB,CZF,CZB
      DO I=2,NX-1
        DO J=2,NY-1
          DO K=2,NH-1
C HYDRAULIC CONDUCTIVITY BETWEEN
NODES IS CALCULATED BY WEIGHTED
HARMONIC MEAN.
C TO REDUCE CALCULATION PROCESS,
HYDRAULIC CONDUCTIVITY IS CALCULATED
IN
C CONDUCTANCE.
C 1.HYDAULIC CONDUCTIVITY BY
WEIGHTED HARMONIC MEAN
C      KXF(I,J,K)=(2*DXF(I)*KX(I-
1,J,K)*KX(I,J,K))/(KX(I,J,K)*DX(I-
1)+
C     >KX(I-1,J,K)*DX(I))
C
KXB(I,J,K)=(2*DXB(I)*KX(I,J,K)*KX(I+
1,J,K))/(KX(I+1,J,K)*DX(I)+
C     >KX(I,J,K)*DX(I+1))
C      KYF(I,J,K)=(2*DYF(J)*KY(I,J-
1,K)*KY(I,J,K))/(KY(I,J,K)*DY(J-1)+
C     >KY(I,J-1,K)*DY(J))
C
KYB(I,J,K)=(2*DYB(J)*KY(I,J,K)*KY(I,
J+1,K))/(KY(I,J+1,K)*DY(J)+
C     >KY(I,J,K)*DY(J+1))
C
KZF(I,J,K)=(2*DZF(K)*KZ(I,J,K-
1)*KZ(I,J,K))/(KZ(I,J,K)*DZ(K-1)+
C     >KZ(I,J,K-1)*DZ(K))
C
KZB(I,J,K)=(2*DZB(K)*KZ(I,J,K)*KZ(I,
J,K+1))/(KZ(I,J,K+1)*DZ(K)+
C     >KZ(I,J,K)*DZ(K+1))
C 2.CONDUCTANCE
C
CXF(I,J,K)=KXF(I,J,K)/(DX(I)*DXF(I))
C
CXB(I,J,K)=KXB(I,J,K)/(DX(I)*DXB(I))
C
CYF(I,J,K)=KYF(I,J,K)/(DY(J)*DYF(J))
C
CYB(I,J,K)=KYB(I,J,K)/(DY(J)*DYB(J))
C
CZF(I,J,K)=KZF(I,J,K)/(DZ(K)*DZF(K))
C
CZB(I,J,K)=KZB(I,J,K)/(DZ(K)*DZB(K))

C IF PUTTING ABOVE
KXF,KXB,KYF,KYB,KZF AND KZB IN
CONDUCTANCE, WE CAN
C GET MORE SIMPLIFIED CONDUCTANCE AS
FOLLOWS
      CXF(I,J,K)=(2*KX(I-
1,J,K)*KX(I,J,K))/((KX(I,J,K)*DX(I-
1)+
     >KX(I-1,J,K)*DX(I))*DX(I))

CXB(I,J,K)=(2*KX(I,J,K)*KX(I+1,J,K))
/((KX(I+1,J,K)*DX(I)+
     >KX(I,J,K)*DX(I+1))*DX(I))
      CYF(I,J,K)=(2*KY(I,J-
1,K)*KY(I,J,K))/((KY(I,J,K)*DY(J-1)+
     >KY(I,J-1,K)*DY(J))*DY(J))

CYB(I,J,K)=(2*KY(I,J,K)*KY(I,J+1,K))
/((KY(I,J+1,K)*DY(J)+
     >KY(I,J,K)*DY(J+1))*DY(J))
      CZF(I,J,K)=(2*KZ(I,J,K-
1)*KZ(I,J,K))/((KZ(I,J,K)*DZ(K-1)+
     >KZ(I,J,K-1)*DZ(K))*DZ(K))

CZB(I,J,K)=(2*KZ(I,J,K)*KZ(I,J,K+1))
/((KZ(I,J,K+1)*DZ(K)+
     >KZ(I,J,K)*DZ(K+1))*DZ(K))
          ENDDO
        ENDDO
      ENDDO

C  DEFINITION OF X,Y,Z MEANING THE
DISTANCE OF EACH NODE FROM ORIGIN
      X(1)=DX(1)/2.
      Y(1)=DY(1)/2.
      Z(1)=DZ(1)/2.
      DO I=2,NX
       X(I)=X(I-1)+DXF(I)
      ENDDO
      DO J=2,NY
       Y(J)=Y(J-1)+DYF(J)
      ENDDO
      DO K=2,NH
       Z(K)=Z(K-1)+DZF(K)
      ENDDO

C  DEFINE INITIAL CZBLEAK,W,QLEAK
      DO I=1,NX
        DO J=1,NY
          DO K=1,NH
```

```
          CZBLEAK(I,J,K)=0.0 ! THIS
MEANS NO LEAKAGE AT ALL CELLS
          W(I,J,K)=0.0 ! ASSUMING
INITIAL VALUES OF W(I,J,K)
          QLEAK(I,J,K)=0.0 !
ASSUMING INITIAL VALUES OF
QLEAK(I,J,K)
          ENDDO
        ENDDO
      ENDDO
C  CALCULATE DISTANCE OF LEAKAGE
PATH (DZBLEAK) AND CONDUCTANCE OF
C  LEAKAGE PATH (CZBLEAK)
C  CALCULATE DZBLEAK AND CZBLEAK AT
ALL LEAKAGE POINTS
      DO I=1,NLC
       SUM=0.0
        DO J=1,(ZLI(I)-ZLIU(I)-1)
          SUM=DZ(ZLI(I)-J)+SUM !
THICKNESS OF AN IMPERMEABLE LAYER
        ENDDO

DZBLEAK(I)=DZ(ZLI(I))/2.+DZ(ZLIU(I))
/2.+SUM
        IF(NFI.EQ.1)THEN

CZBLEAK(XLI(I),YLI(I),ZLIU(I))=KZBLE
AK(XLI(I),YLI(I),ZLIU(I))*

>ALEAK(XLI(I),YLI(I),ZLIU(I))/(DX(XL
I(I))*DY(YLI(I))*DZ(ZLIU(I))*
     >DZBLEAK(I))

CZBLEAK(XLI(I),YLI(I),ZLI(I))=KZBLEA
K(XLI(I),YLI(I),ZLI(I))*

>ALEAK(XLI(I),YLI(I),ZLI(I))/(DX(XLI
(I))*DY(YLI(I))*DZ(ZLI(I))*
     >DZBLEAK(I))
        ELSEIF(NFI.EQ.2)THEN

CZBLEAK(XLI(I),YLI(I),ZLIU(I))=AKL(X
LI(I),YLI(I),ZLIU(I))/

>(DX(XLI(I))*DY(YLI(I))*DZ(ZLIU(I))*
DZBLEAK(I))

CZBLEAK(XLI(I),YLI(I),ZLI(I))=AKL(XL
I(I),YLI(I),ZLI(I))/

>(DX(XLI(I))*DY(YLI(I))*DZ(ZLI(I))*D
ZBLEAK(I))
        ENDIF
      ENDDO
C  STARTING SIMULATION TIME
      ST=0.0 ! STARTING TIME OF
SIMULATION PUTS ON ZERO
      STIME(1)=ST  ! STIME(1) :
SIMULATION TIME AT 1ST TIME STEP
C  MAXIMUM NUMBER OF TIME STEP.
2*NLC MEANS LSTIME AND APTOLK
C      NT=DINT((ET-ST)/DT+1)+2*NLC !
MAXIMUM NUMBER OF TIME STEP

C  FIND SOME LEAKAGE POINTS THAT
LEAKAGE STARTS AT ST OR NOT STARTS
AT ST.
      NCST=0 ! COUNTING LEAKAGE
POINTS WITH LEAKAGE AT STARTING TIME
      NNK=0 ! COUNTING LEAKAGE
POINTS WITH NO LEAKAGE AT STARTING
TIME
      DO I=1,NLC
       IF (OLST(I).EQ.STIME(1)) THEN
! IN CASE OF LEAKAGE AT ST
        NCST=NCST+1
        TLS(NCST)=OLST(I) ! LEAKAGE
STARTING TIME
        XLS(NCST)=XLI(I) ! X-COORD.
OF LEAKAGE OCCURING AT INJEC. AQUI.
        YLS(NCST)=YLI(I) ! Y-COORD.
OF LEAKAGE OCCURING AT INJEC. AQUI.
        ZLS(NCST)=ZLI(I) ! Z-COORD.
OF LEAKAGE OCCURING AT INJEC. AQUI.
        ZLSU(NCST)=ZLIU(I) ! Z-COORD
OF LEAKAGE OCCURING AT UP. AQUI.
        APTOLK(NCST)=0.0 !DEFINITION
OF APTOLK WHEN LEAKAGE STARTS AT ST
C  CHANGE TO COINCIDE PARAMETERS IN
DIFFERENCE EQUATION
        XCL(NCST)=XLS(NCST) ! X-
COORD. OF LEAKAGE STARTING AT INJ.
AQ.
        YCL(NCST)=YLS(NCST)
        ZCL(NCST)=ZLS(NCST)
        ZCLU(NCST)=ZLSU(NCST) ! Z-
COORD. OF LEAKAGE STARTING AT UP.
AQ.
        XAT(NCST)=XLS(NCST) ! X-
COORD. OF LEAKAGE APPROACH AT UP.
AQ.
        YAT(NCST)=YLS(NCST)
        ZAT(NCST)=ZLS(NCST)
        ZATU(NCST)=ZLSU(NCST) ! Z-
COORD. OF LEAKAGE APPROACH AT UP.
AQ.
       ENDIF
      ENDDO
      DO J=1,NLC
```

```
       IF (OLST(J).NE.STIME(1)) THEN
! IN CASE OF NO LEAKAGE AT ST
        NNK=NNK+1
        NCNS=NNK+NCST
        TLS(NCNS)=OLST(J) ! LEAKAGE
STARTING TIME AT NO ST
        XLS(NCNS)=XLI(J) ! X-COORD
OF LEAKAGE AT INJ. AQUI AT NO ST
        YLS(NCNS)=YLI(J)
        ZLS(NCNS)=ZLI(J)
        ZLSU(NCNS)=ZLIU(J) ! Z-COORD
OF LEAKAGE AT UP. AQUI.
       ENDIF
      ENDDO

C ------------------- SOLVING
DIFFERENCE EQUATION ----------------
--------
C  SOLVING GOVERNING EQUATIONS TO
GET HYDRAULIC HEAD AND FLUX OF EACH
C  NODE(I,J,K), AND LEAKAGE RATE AT
LEAKAGE POINTS

C  INITIAL HEAD VALUES OF INTERNAL
NODES FOR GAUSS SEIDAL METHOD TO
SOLVE
C  LINEAR MATRIX (HEAD(I,J,K)=0.).

      DO I=2,NX-1
        DO J=2,NY-1
          DO K=2,NH-1
            IF (NBC(I,J,K).NE.2)
THEN
            HEAD(I,J,K)=HYH(I,J,K)
            ENDIF
          ENDDO
        ENDDO
      ENDDO

C  CONSTANT HEAD BOUNDARY CONDITION
AND SINK/SOURCE VALUES FOR STEADY
C  STATE CONDITION. THAT IS TO SAY,
THEY ARE FOR VALUES AT STIME(1)(=ST,
C  I.E. ST=0.0)
      DO J=1,NCHB

HEAD(NXHB(J),NYHB(J),NZHB(J))=BHEAD(
1,J) ! BHEAD AT ST(TIME=0.0)
      ENDDO
      DO J=1,NSST

W(NXSS(J),NYSS(J),NZSS(J))=BFTI(1,J)
! S/S AT ST(TIME=0.0)
      ENDDO
      NLA1=0
      NLA2=0
      NPNLA2(1)=0
C      NPNLA1=0
      TMTC1=-1.0E5
C      TMTC2=-1E5
      NLT1=NCST
      NLT2=NCST
       IT=1 ! ITERATION NUMBER IN
ONE TIME STEP
      NIT=1 ! COUNTING NUMBER OF
TIME STEP.
      MAXIT=10000
      NSUM=1
      DO I=1,NLC
       SUM1(I)=0.0
       SUM2(I)=0.0
      ENDDO

      DO 40 WHILE (MAXIT.GE.IT)
       VMAX=0.0

      DO I=2,NX-1
       DO J=2,NY-1
        DO K=2,NH-1
        IF(NBC(I,J,K).NE.2) THEN
         IF (NIT.EQ.1) THEN !SOLVING
TO STEADY STATE CONDITION
C           IF (NCST.NE.0) THEN ! IN
CASE OF LEAKAGE STARTING AT ST
           DO IL=1,NCST

IF((I.EQ.XAT(IL)).AND.(J.EQ.YAT(IL))
.AND.(K.EQ.ZATU(IL)))THEN

MU1(I,J,K)=CXF(I,J,K)+CXB(I,J,K)+CYF
(I,J,K)+CYB(I,J,K)+CZF(I,J,K)+
     >CZB(I,J,K)+CZBLEAK(I,J,K)
      VA=(CXF(I,J,K)*HEAD(I-
1,J,K)+CXB(I,J,K)*HEAD(I+1,J,K)+CYF(
I,J,K)*
     >HEAD(I,J-
1,K)+CYB(I,J,K)*HEAD(I,J+1,K)+CZF(I,
J,K)*HEAD(I,J,K-1)+

>CZB(I,J,K)*HEAD(I,J,K+1)+W(I,J,K)+C
ZBLEAK(I,J,K)*
     >HEAD(I,J,ZAT(IL)))/MU1(I,J,K)
!FOR UPPER AQUIFER WITH LEAKAGE
              GOTO 201

ELSEIF((I.EQ.XCL(IL)).AND.(J.EQ.YCL(
IL)).AND.(K.EQ.ZCL(IL)))
     >THEN
```

```
MU1(I,J,K)=CXF(I,J,K)+CXB(I,J,K)+CYF
(I,J,K)+CYB(I,J,K)+CZF(I,J,K)+
     >CZB(I,J,K)+CZBLEAK(I,J,K)
      VA=(CXF(I,J,K)*HEAD(I-
1,J,K)+CXB(I,J,K)*HEAD(I+1,J,K)+CYF(
I,J,K)*
     >HEAD(I,J-
1,K)+CYB(I,J,K)*HEAD(I,J+1,K)+CZF(I,
J,K)*HEAD(I,J,K-1)+

>CZB(I,J,K)*HEAD(I,J,K+1)+W(I,J,K)+C
ZBLEAK(I,J,K)*
     >HEAD(I,J,ZCLU(IL)))/MU1(I,J,K)
!FOR INJECTION AQUIFER WITH LEAKAGE
              GOTO 201
C            ELSEIF (IL.EQ.NCST) GOTO
101
           ENDIF
           ENDDO
C          ELSEIF (NCST.EQ.0) THEN !
IN CASE OF NO LEAKAGE AT ST

MU1(I,J,K)=CXF(I,J,K)+CXB(I,J,K)+CYF
(I,J,K)+CYB(I,J,K)+CZF(I,J,K)+
     >CZB(I,J,K)
      VA=(CXF(I,J,K)*HEAD(I-
1,J,K)+CXB(I,J,K)*HEAD(I+1,J,K)+CYF(
I,J,K)*
     >HEAD(I,J-
1,K)+CYB(I,J,K)*HEAD(I,J+1,K)+CZF(I,
J,K)*HEAD(I,J,K-1)+

>CZB(I,J,K)*HEAD(I,J,K+1)+W(I,J,K))/
MU1(I,J,K)
C          ENDIF
         ENDIF
C  END OF CALCULATION OF STEADY
STATE CONDITION

C  START OF CALCULATION OF UNSTEADY
CONDITION
         IF (NIT.NE.1) THEN !
SOLVING TO UNSTEADY CONDITION
C           IF (NLT2.NE.0) THEN ! IN
CASE UP.AQU. HAS INFLOW BY LEAKAGE
             DO IJ=1,NLT2      !
FROM UPP.AQU.
              IF
((I.EQ.XAT(IJ)).AND.(J.EQ.YAT(IJ)).A
ND.(K.EQ.ZATU(IJ)))
     >THEN ! IN CASE OF INFLOW INTO
UPPER AQUIFER

MU2(I,J,K)=CXF(I,J,K)+CXB(I,J,K)+CYF
(I,J,K)+CYB(I,J,K)+CZF(I,J,K)+

>CZB(I,J,K)+SS(I,J,K)/(STIME(NIT)-
STIME(NIT-1))+CZBLEAK(I,J,K)
      VA=(CXF(I,J,K)*HEAD(I-
1,J,K)+CXB(I,J,K)*HEAD(I+1,J,K)+CYF(
I,J,K)*
     >HEAD(I,J-
1,K)+CYB(I,J,K)*HEAD(I,J+1,K)+CZF(I,
J,K)*HEAD(I,J,K-1)+

>CZB(I,J,K)*HEAD(I,J,K+1)+W(I,J,K)+(
SS(I,J,K)*PSHEAD(I,J,K))/
     >(STIME(NIT)-STIME(NIT-
1))+CZBLEAK(I,J,K)*HEAD(I,J,ZAT(IJ))
)/
     >MU2(I,J,K)
                GOTO 201
C              ELSEIF (IJ.EQ.NLT2)
GOTO 301 ! FOR SOME NO INFLOW CELLS
             ENDIF
! IN UP.AQ.
            ENDDO
C           ELSEIF (NLT2.EQ.0) GOTO
301 ! FOR NO INFLOW AT ALL CELL IN
C           ENDIF
! UP.AQ.
C           IF (NLT1.NE.0) THEN ! IN
CASE INJ.AQU. HAS OUTFLOW BY LEAKAGE
             DO JL=1,NLT1      !
FROM INJ.AQU.

IF((I.EQ.XCL(JL)).AND.(J.EQ.YCL(JL))
.AND.(K.EQ.ZCL(JL)))
     >THEN ! IN CASE OF OUTFLOW BY
LEAKAGE FROM INJECTION AQUIFER

MU2(I,J,K)=CXF(I,J,K)+CXB(I,J,K)+CYF
(I,J,K)+CYB(I,J,K)+CZF(I,J,K)+

>CZB(I,J,K)+SS(I,J,K)/(STIME(NIT)-
STIME(NIT-1))+CZBLEAK(I,J,K)
      VA=(CXF(I,J,K)*HEAD(I-
1,J,K)+CXB(I,J,K)*HEAD(I+1,J,K)+CYF(
I,J,K)*
     >HEAD(I,J-
1,K)+CYB(I,J,K)*HEAD(I,J+1,K)+CZF(I,
J,K)*HEAD(I,J,K-1)+

>CZB(I,J,K)*HEAD(I,J,K+1)+W(I,J,K)+(
SS(I,J,K)*PSHEAD(I,J,K))/
```

```
     >(STIME(NIT)-STIME(NIT-
1))+CZBLEAK(I,J,K)*HEAD(I,J,ZCLU(JL)
))/
     >MU2(I,J,K)
                  GOTO 201
C               ELSEIF (JL.EQ.NLT1)
GOTO 301 ! FOR SOME NO LEAKAGE CELLS
              ENDIF
! IN INJECTION AQUIFER
             ENDDO
C           ELSEIF
((IJ.EQ.NLT2).EQ.(JL.EQ.NLT1)) THEN
C             GOTO 301
C            ENDIF
C          ELSEIF (NLC.EQ.0) THEN
!IN CASE OF NO LEAKAGE DURING ALL
TIME


MU2(I,J,K)=CXF(I,J,K)+CXB(I,J,K)+CYF
(I,J,K)+CYB(I,J,K)+CZF(I,J,K)+

>CZB(I,J,K)+SS(I,J,K)/(STIME(NIT)-
STIME(NIT-1))
      VA=(CXF(I,J,K)*HEAD(I-
1,J,K)+CXB(I,J,K)*HEAD(I+1,J,K)+CYF(
I,J,K)*
     >HEAD(I,J-
1,K)+CYB(I,J,K)*HEAD(I,J+1,K)+CZF(I,
J,K)*HEAD(I,J,K-1)+

>CZB(I,J,K)*HEAD(I,J,K+1)+W(I,J,K)+(
SS(I,J,K)*PSHEAD(I,J,K))/
     >(STIME(NIT)-STIME(NIT-
1)))/MU2(I,J,K)
C               GOTO 201
C           ELSEIF (NCST.EQ.NLC) THEN
! IN CASE OF NO LEAKAGE AFTER ST
C           ENDIF
         ENDIF
 201         IF (ABS((VA-
HEAD(I,J,K))/VA).GT.VMAX) THEN
               VMAX=ABS((VA-
HEAD(I,J,K))/VA)
             ENDIF
               HEAD(I,J,K)=VA
        ENDIF
        ENDDO
       ENDDO
      ENDDO

      IF (VMAX.LE.TOL) THEN
C      WRITE(*,*) NIT, STIME(NIT)
C ----------------- OPTION TO PRINT
OUT
         IF
((IPRINT1.NE.0).AND.(IPRINT1.NE.1))
THEN
        WRITE(*,*) ' WARNING!
IPRINT1 OF PRINT OPTION MUST BE 0 OR
1 '
        WRITE(5,*) ' WARNING!
IPRINT1 OF PRINT OPTION MUST BE 0 OR
1 '
          STOP
         ENDIF
         IF
((IPRINT2.NE.0).AND.(IPRINT2.NE.1))
THEN
        WRITE(*,*) ' WARNING!
IPRINT2 OF PRINT OPTION MUST BE 0 OR
1 '
        WRITE(5,*) ' WARNING!
IPRINT2 OF PRINT OPTION MUST BE 0 OR
1 '
          STOP
         ENDIF

        IF(NFI.EQ.1) THEN


WRITE(*,'(/,A,I6,5X,A,F14.3,5X,A,I7)
') '### TIME STEP # = ', NIT,
     >'SIMULATION TIME =',
STIME(NIT), ' ITERATION # FOR
CONVERGENCE =',
     >IT

          IF (IPRINT1.EQ.1) THEN
            IF (NIT.EQ.NSUM) THEN
       WRITE(5,'(/,A,F12.3,A)')
'################## SIMULATION TIME
=',
     >STIME(NIT), '
###################'
       WRITE(5,'(A,I5)')' TIME STEP
# = ', NIT
        WRITE(5,'(A,I5)')' ITERATION
# FOR CONVERGENCE =', IT
       WRITE(5,'(A)')'  I  J  K
X(I)   Y(J)   Z(K)      HEAD
     >VXF          VXB          VYF
VYB          VZF
     >VZB          QXF          QXB
QYF          QYB
     >QZF          QZB
QLEAK'
            ENDIF
          ENDIF
        ENDIF
```

```
      DO I=2,NX-1
       DO J=2,NY-1
        DO K=2,NH-1
        IF(NBC(I,J,K).NE.2) THEN
         PSHEAD(I,J,K)=HEAD(I,J,K) !
SAVE PREVIOUS STEP HYDRAULIC HEAD

C ---------------- CALCULATING
VELOCITY (M/S)

VXF(I,J,K)=CXF(I,J,K)*DX(I)*(HEAD(I-
1,J,K)-HEAD(I,J,K))

VXB(I,J,K)=CXB(I,J,K)*DX(I)*(HEAD(I+
1,J,K)-HEAD(I,J,K))

VYF(I,J,K)=CYF(I,J,K)*DY(J)*(HEAD(I,
J-1,K)-HEAD(I,J,K))

VYB(I,J,K)=CYB(I,J,K)*DY(J)*(HEAD(I,
J+1,K)-HEAD(I,J,K))

VZF(I,J,K)=CZF(I,J,K)*DZ(K)*(HEAD(I,
J,K-1)-HEAD(I,J,K))

VZB(I,J,K)=CZB(I,J,K)*DZ(K)*(HEAD(I,
J,K+1)-HEAD(I,J,K))

C ---------------- CALCULATING FLUX
(M**3/S)

QXF(I,J,K)=VXF(I,J,K)*DY(J)*DZ(K)

QXB(I,J,K)=VXB(I,J,K)*DY(J)*DZ(K)

QYF(I,J,K)=VYF(I,J,K)*DX(I)*DZ(K)

QYB(I,J,K)=VYB(I,J,K)*DX(I)*DZ(K)

QZF(I,J,K)=VZF(I,J,K)*DX(I)*DY(J)

QZB(I,J,K)=VZB(I,J,K)*DX(I)*DY(J)

C ----------------- CALCULATE
LEAKAGE RATE (M**3/S)
         DO KM=1,NLT2      ! LEAKAGE
INFLOW INTO UPPER AQUIFER
           IF
((I.EQ.XAT(KM)).AND.(J.EQ.YAT(KM)).A
ND.(K.EQ.ZATU(KM)))
     >
THEN

QLEAK(I,J,K)=CZBLEAK(I,J,K)*DX(I)*DY
(J)*DZ(K)*(HEAD(I,J,K)
     >                -
HEAD(I,J,ZAT(KM)))
          QLUPP(NIT,KM)=QLEAK(I,J,K)
           ENDIF
         ENDDO
         DO KN=1,NLT1      ! LEAKAGE
OUTFLOW FROM INJECTION AQUIFER

IF((I.EQ.XCL(KN)).AND.(J.EQ.YCL(KN))
.AND.(K.EQ.ZCL(KN)))
     >
THEN

QLEAK(I,J,K)=CZBLEAK(I,J,K)*DX(I)*DY
(J)*DZ(K)*(HEAD(I,J,K)
     >                -
HEAD(I,J,ZCLU(KN)))
          QLINJ(NIT,KN)=QLEAK(I,J,K)
           ENDIF
         ENDDO

C ------------------ FORWARD OPTION
TO PRINT OUT
         IF (NFI.EQ.1) THEN
           IF (IPRINT1.EQ.1) THEN
             IF (NIT.EQ.NSUM) THEN

WRITE(5,'(3I3,1x,3F10.3,X,F13.5,13F1
3.9)') I, J, K, X(I), Y(J),
     >Z(K),HEAD(I,J,K), VXF(I,J,K),
VXB(I,J,K), VYF(I,J,K), VYB(I,J,K),
     >VZF(I,J,K), VZB(I,J,K),
QXF(I,J,K), QXB(I,J,K), QYF(I,J,K),
     >QYB(I,J,K), QZF(I,J,K),
QZB(I,J,K), QLEAK(I,J,K)
              IF (I.EQ.NX-
1.AND.J.EQ.NY-1.AND.K.EQ.NH-1)
NSUM=NSUM+IPTIT
             ENDIF
            ENDIF

           IF (IPRINT2.EQ.1) THEN
             DO IH=1,NCPH

IF(I.EQ.NPXH(IH).AND.J.EQ.NPYH(IH).A
ND.K.EQ.NPZH(IH))THEN

HPO(NIT,IH)=HEAD(I,J,K)
              ENDIF
            ENDDO
            DO JH=1,NCPQ
```

```
IF(I.EQ.NPXQ(JH).AND.J.EQ.NPYQ(JH).A
ND.K.EQ.NPZQ(JH))THEN

QPXF(NIT,JH)=QXF(I,J,K)

QPXB(NIT,JH)=QXB(I,J,K)

QPYF(NIT,JH)=QYF(I,J,K)

QPYB(NIT,JH)=QYB(I,J,K)

QPZF(NIT,JH)=QZF(I,J,K)

QPZB(NIT,JH)=QZB(I,J,K)
              ENDIF
             ENDDO
            ENDIF
          ENDIF
C ---------------INVERSE CALCULATED
& PRINTOUT DATA
C    OBTAIN CALCULATED HEAD DATA AT
MEASUREMENT POSTS
           IF (NFI.EQ.2) THEN
             DO IN=1,NMD ! NUMBER OF
MEASUREMENT DATA

IF(I.EQ.NXM(IN).AND.J.EQ.NYM(IN).AND
.K.EQ.NZM(IN))THEN

HEADCAL(NIT,IN)=HEAD(I,J,K) ! OBTAIN
CALCULATED DATA
               ENDIF
             ENDDO
C    OBTAIN HEAD DATA AT EXPECTED
LEAKAGE POINTS AFTER LAST GENERATION
             IF (NSM.GE.(MAXGEN-
1)*NPOPSIZ)THEN
               DO IM=1,NEL

IF(I.EQ.XEL(IM).AND.J.EQ.YEL(IM).AND
.K.EQ.ZELU(IM))THEN

HEADUPP(NIT,IM)=HEAD(I,J,K)
                 ENDIF

IF(I.EQ.XEL(IM).AND.J.EQ.YEL(IM).AND
.K.EQ.ZEL(IM))THEN

HEADINJ(NIT,IM)=HEAD(I,J,K)
                 ENDIF
               ENDDO
             ENDIF
           ENDIF

C ---------------END OF OBTAINING
INVERSE CALCULATED & PRINTOUT DATA
         ENDIF
         ENDDO
       ENDDO
      ENDDO

C ---------------- FORWARD PRINT
OPTION
       IF(NFI.EQ.1)THEN
         IF (NSIMCON.EQ.1)THEN ! END
OF STEADY CONDITION SIMULATION
            STOP
         ELSEIF (STIME(NIT).GE.ET)
THEN ! END OF UNSTEADY SIMULATION
C ------------------ OPTION TO PRINT
OUT
C BEFOR FINISHING SIMULATION,
CONDUCT PRINT OPTION
           IF (IPRINT2.EQ.1) THEN
             IF (NCPH.NE.0) THEN
           WRITE(5,'(A)') ' TIME
HYDRAULIC HEAD DISTRIBUTION'
C          WRITE(5,'(30I3)')
(NPXH(I),NPYH(I),NPZH(I),I=1,NCPH)
           DO K=1,NIT

WRITE(5,'(F12.3,200F18.10)')
STIME(K),(HPO(K,IH),IH=1,NCPH)
           ENDDO
             ENDIF

             IF (NCPQ.NE.0) THEN
           WRITE(5,'(A)') ' TIME
FLUX DISTRIBUTION (QXF QXB
     >QYF QYB QZF QZB) AT EACH CELL'
C          WRITE(5,'(30I3)')
(NPXQ(J),NPYQ(J),NPZQ(J),J=1,NCPQ)
           DO K=1,NIT

WRITE(5,'(F12.3,30F14.10)')
STIME(K),(QPXF(K,JH),QPXB(K,JH),
     >
QPYF(K,JH),QPYB(K,JH),QPZF(K,JH),QPZ
B(K,JH),JH=1,NCPQ)
           ENDDO
             ENDIF
           ENDIF
            STOP
         ENDIF
       ENDIF
C ------------------ INVERSE PRINT
OPTION

       IF(NFI.EQ.2)THEN
```

```
C  PRINT HEAD DATA & LEAKAGE RATE AT
EXPECTED LEAKAGE POINTS AFTER LAST
GENERATION
         IF (STIME(NIT).GE.ET)THEN
C V2.7           IF
(NSM.GT.(NGENERA-1)*NPOPULA)THEN
         ! PRINT OUT HEAD DATA AT
EXPECTED LEAKAGE POINTS
C V2.7
WRITE(5,'(7X,60I4)')
(XEL(IK),YEL(IK),ZELU(IK),XEL(IK),
C V2.7     >YEL(IK),ZEL(IK),
IK=1,NEL), (NXM(IN),NYM(IN),NZM(IN),
IN=1,NMD)
C V2.7             DO K=1,NIT
C V2.7
WRITE(5,'(F12.4,20F15.8)')
STIME(K),(HEADUPP(K,IL),
C V2.7     >HEADINJ(K,IL),
IL=1,NEL), (HEADCAL(K,IN),IN=1,NMD)
C V2.7             ENDDO
         ! PRINT OUT LEAKAGE RATES
AT EXPECTED LEAKAGE POINTS
C V2.7
WRITE(5,'(7X,30I4)')
(XAT(II),YAT(II),ZATU(II),XCL(II),
C V2.7     >YCL(II),ZCL(II),
II=1,NEL)
C V2.7             DO L=1,NIT
C V2.7
WRITE(5,'(F12.4,20F15.8)')
STIME(L),(QLUPP(L,IJ),
C V2.7     >QLINJ(L,IJ), IJ=1,NEL)
C V2.7             ENDDO
C V2.7           ENDIF
         NSM=NSM+1 ! COUNTING TOTAL
NUMBER OF SIMULATION
         RETURN
         ENDIF
       ENDIF
C -----------------END OF INVERSE
PRINT OPTION

C   CALCULATE TIME STEP
         CALL
TIMESTEP(NIT,STIME,DT,NLA1,NLA2,NCST
,NLC,TMTC1,NLT1,

>NLT2,DZBLEAK,KZBLEAK,HEAD,TLS,XLS,Y
LS,ZLS,ZLSU,XAT,YAT,ZAT,ZATU,
     >XCL,YCL,ZCL,ZCLU,NOL)

C   CALCULATE BOUNDARY CONDITION AT
TIME STEP
         CALL
INTERPOL(NTB1,NTB2,TSBC1,TSBC2,NXHB,
NYHB,NZHB,NXSS,

>NYSS,NZSS,BHEAD,BFTI,HEAD,W,NCHB,NS
ST,DT,ET,NLC,NIT,STIME,ST)

        IT=0
      ENDIF

       IF (IT.EQ.MAXIT) THEN
        WRITE(*,'(A,I5)')'PROGRAM
STOPED BECAUSE THE NUMBER OF
ITERATION
     > EXCEEDED MAXIMUM ALLOWABLE
ITERATION NUMBER AT TIME STEP', NIT
        WRITE(5,'(A,I5)')'PROGRAM
STOPED BECAUSE THE NUMBER OF
ITERATION
     > EXCEEDED MAXIMUM ALLOWABLE
ITERATION NUMBER AT TIME STEP', NIT
         STOP
        ENDIF

        IT=IT+1
  40  ENDDO

      END

C
#####################################
#####################################
#
      SUBROUTINE
TIMESTEP(NIT,STIME,DT,NLA1,NLA2,NCST
,NLC,TMTC1,NLT1,

>NLT2,DZBLEAK,KZBLEAK,HEAD,TLS,XLS,Y
LS,ZLS,ZLSU,XAT,YAT,ZAT,ZATU,
     >XCL,YCL,ZCL,ZCLU,NOL)
C  CALCULATE NEXT TIME STEP BY
COMPARISON AMONG NORMAL SIMULATION
TIME BY DT,
C  LEAKAGE STARTING TIME FROM
INJECTION AQUIFER, AND LEAKAGE
INFLOW TIME INTO
C  UPPER AQUIFER. LEAKAGE STARTING
TIME AND LEAKAGE INFLOW TIME INTO
UPPER
C  AQUIFER ARE DETERMINED TO TIME
STEP. BECAUSE THE SIMULATOR MUST BE
C  SIMULATED AT THAT TIME WHEN
LEAKAGE INFLOWS OR OUTFLOWS INTO/OUT
EACH AQUIFER
C
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      INCLUDE 'PARAMS.F'
C      COMMON /FS/ NIT,NLT2
      DIMENSION STIME(NTIMES)
      DIMENSION
IMIN(NNODES*NNODES),NIMIN(NNODES*NNO
DES)
      DIMENSION DZBLEAK(NNODES)
      DIMENSION
HEAD(NXNODE,NYNODE,NZNODE)
      DIMENSION TLS(NNODES*NNODES)
      DIMENSION
SUM1(NNODES*NNODES),SUM2(NNODES*NNOD
ES)
      DIMENSION
NPNLA2(NNODES*NNODES)

      REAL
KZBLEAK(NXNODE,NYNODE,NZNODE)
      INTEGER
XLS(NXNODE),YLS(NYNODE),ZLS(NZNODE),
ZLSU(NZNODE)
C      INTEGER
XLA(NNODES),YLA(NNODES),ZLA(NNODES),
ZLAU(NNODES)
      INTEGER
XAT(NXNODE),YAT(NYNODE),ZAT(NZNODE),
ZATU(NZNODE)
      INTEGER
XTL(NXNODE),YTL(NYNODE),ZTL(NZNODE),
ZLTL(NZNODE)
      INTEGER
XCL(NXNODE),YCL(NYNODE),ZCL(NZNODE),
ZCLU(NZNODE)

      NIT=NIT+1
      STIME(NIT)=STIME(NIT-1)+DT
      IF
((NLC.NE.NCST).AND.(NIT.EQ.2)) THEN
C        NPNLA1=NLA1
        CALL
STRTIME(NLA1,NCST,NLC,TMTC1,IMIN,TLS
,XLS,YLS,ZLS,ZLSU,XTL,
     >YTL,ZTL,ZLTL)
      ENDIF

      IF ((STIME(NIT-1).LT.TMTC1)
.AND. (STIME(NIT).GE.TMTC1)) THEN
          STIME(NIT)=TMTC1
        IF(NOL.EQ.1) THEN  ! NO FOR
INVERSE ANALYSIS (AT INVERSE NOL=2)
         CALL
APTIME(NLA2,NLT1,NCST,NIMIN,DZBLEAK,
KZBLEAK,HEAD,XLS,

>YLS,ZLS,ZLSU,XAT,YAT,ZAT,ZATU,SUM1,
SUM2,STIME,NPNLA2,NIT,NLT2)
        ENDIF
         IF (STIME(NIT).EQ.TMTC1)
THEN
            NLT1=NLA1+NCST
            IF(NOL.EQ.2) NLT2=NLT1
          DO JK=1,NLA1
            NIMIN(JK)=IMIN(JK)
          ENDDO
          DO IK=NCST+1,NLT1
            XCL(IK)=XTL(IK-NCST)
            YCL(IK)=YTL(IK-NCST)
            ZCL(IK)=ZTL(IK-NCST)
            ZCLU(IK)=ZLTL(IK-NCST)
             IF(NOL.EQ.2) THEN
               XAT(IK)=XCL(IK)
               YAT(IK)=YCL(IK)
               ZAT(IK)=ZCL(IK)
               ZATU(IK)=ZCLU(IK)
             ENDIF
          ENDDO
          IF (NLT1.LT.NLC) THEN
            CALL
STRTIME(NLA1,NCST,NLC,TMTC1,IMIN,TLS
,XLS,YLS,ZLS,ZLSU,
     >XTL,YTL,ZTL,ZLTL)
            ENDIF
         ENDIF
      ELSEIF (NLT1.GT.NLT2) THEN
C          NPNLA1=NLA1
C          NPTMTC1=TMTC1
        IF (NOL.EQ.1) THEN
          CALL
APTIME(NLA2,NLT1,NCST,NIMIN,DZBLEAK,
KZBLEAK,HEAD,XLS,

>YLS,ZLS,ZLSU,XAT,YAT,ZAT,ZATU,SUM1,
SUM2,STIME,NPNLA2,NIT,NLT2)
        ENDIF
      ENDIF

      RETURN
      END
C
#####################################
#####################################
#
      SUBROUTINE
STRTIME(NLA1,NCST,NLC,TMTC1,IMIN,TLS
,XLS,YLS,ZLS,ZLSU,
     >XTL,YTL,ZTL,ZLTL)
```

```
C  THIS IS TO FIND LEAKAGE STARTING
TIME AT INJECTION AQUIFER TO
ASCENDING ORDER
C  IN TIME. THIS SUBROUTINE
DETERMINES MINIMUM LEAKAGE STARTING
TIME AFTER PRESENT
C  TIME STEP (EXEPT PREVIOUS
SELECTED LEAKAGE STARTING TIME). THE
MINIMUM LEAKAGE
C  STARTING TIME IS DETERMINED TO
TIME STEP IN SUBROUTINE TIMESTEP.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INCLUDE 'PARAMS.F'
      DIMENSION TLS(NNODES*NNODES)
      DIMENSION
IMIN(NNODES*NNODES),NIDMIN(NNODES*NN
ODES)

      INTEGER
XTL(NXNODE),YTL(NYNODE),ZTL(NZNODE),
ZLTL(NZNODE)
      INTEGER
XLS(NXNODE),YLS(NYNODE),ZLS(NZNODE),
ZLSU(NZNODE)
      INTEGER
DXTL(NXNODE),DYTL(NYNODE),DZTL(NZNOD
E),DZLTL(NZNODE)

      NC1=0
      NLA1=NLA1+1
      TLMIN1=1.0E30
C      TEMIN1=-1.0E10
      DO I=NCST+1,NLC
        IF (TLS(I).GT.TMTC1) THEN
          IF (TLS(I).LT.TLMIN1) THEN
            TLMIN1=TLS(I)
            IMIN(NLA1)=I
            XTL(NLA1)=XLS(I)
            YTL(NLA1)=YLS(I)
            ZTL(NLA1)=ZLS(I)
            ZLTL(NLA1)=ZLSU(I)
          ELSEIF (TLS(I).EQ.TLMIN1)
THEN
            NC1=NC1+1
            TEMIN1=TLMIN1
            NIDMIN(NC1)=I
            DXTL(NC1)=XLS(I)
            DYTL(NC1)=YLS(I)
            DZTL(NC1)=ZLS(I)
            DZLTL(NC1)=ZLSU(I)
          ENDIF
        ENDIF
      ENDDO
        TMTC1=TLMIN1
      IF (TEMIN1.EQ.TLMIN1) THEN
        DO II=1,NC1
          NLA1=NLA1+1
          IMIN(NLA1)=NIDMIN(II)
          XTL(NLA1)=DXTL(II)
          YTL(NLA1)=DYTL(II)
          ZTL(NLA1)=DZTL(II)
          ZLTL(NLA1)=DZLTL(II)
        ENDDO
      ENDIF

      RETURN
      END

C
#####################################
#####################################
#
      SUBROUTINE
APTIME(NLA2,NLT1,NCST,NIMIN,DZBLEAK,
KZBLEAK,HEAD,XLS,

>YLS,ZLS,ZLSU,XAT,YAT,ZAT,ZATU,SUM1,
SUM2,STIME,NPNLA2,NIT,NLT2)

C  THIS IS TO CALCULATE LEAKAGE
TRAVEL TIME INTO UPPER AQUIFER. THE
C  SUBROUTINE DETERMINES EACH
LEAKAGE TRAVEL LENGTH FROM NEXT TIME
STEP
C  AFTER CURRENT TIME STEP WHEN
LEAKAGE STARTS FROM INJECTION
AQUIFER
C  (THE CURRENT TIME STEP MEANS THE
RIGHT TIME WHEN LEAKAGE OCCURS AT
INJECTION
C  AQUIFER TIME FOR TIME STEP). IF
LEAKAGE TRAVEL LENGTH IS GREATER
THAN
C  LEAKAGE PATHWAY (DZBLEAK(I)),
LEAKAGE TREVEL LENGTH IS CORRECTED
TO
C  DZBLEAK EQUALLY AND THEN TIME
STEP IS DETERMINED.
C
C  THIS SUBROUTINE IS USED FOR THE
FIRST OPTION (NOL=1, CALCULATION OF
TRAVEL
C  TIME) OF FORWARD SIMULATION. THE
INVERSE ANALYSIS DOESN'T USE THIS
ROUTINE
C  BECAUSE TRAVEL TIME CAN'T BE
CALCULATED. IN INVERSE ANALYSIS KA
VALUES ARE
```

```
C  INTEGRATED, SO KA VALUES SHOULD
BE SEPARATED TO CALCULATE TRAVEL
TIME.
C  TRAVEL TIME IS CALCULATED FROM
DZBLENTH/(KI) SO WE CANN'T CONSIDER
LEAKAGE
C  TRAVEL TIME IN INVERSE ANALYSIS.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INCLUDE 'PARAMS.F'
C      COMMON /FS/NIT,NLT2
      DIMENSION NIMIN(NNODES*NNODES)
      DIMENSION DZBLEAK(NNODES)
      DIMENSION
HEAD(NXNODE,NYNODE,NZNODE)
      DIMENSION
DAZL(NNODES*NNODES),AZL(NNODES*NNODE
S)
      DIMENSION
SUM1(NNODES*NNODES),SUM2(NNODES*NNOD
ES)
      DIMENSION
STIME(NTIMES),NPNLA2(NNODES*NNODES)
      DIMENSION
APTOLK(NNODES*NNODES),VOLK(NNODES*NN
ODES)
C      REAL
LAPTIM(NNODES*NNODES),LVEL(NNODES*NN
ODES)
      REAL
KZBLEAK(NXNODE,NYNODE,NZNODE)
      INTEGER
XLS(NXNODE),YLS(NYNODE),ZLS(NZNODE),
ZLSU(NZNODE)
      INTEGER
XAT(NXNODE),YAT(NYNODE),ZAT(NZNODE),
ZATU(NZNODE)

C      TLMIN2=1E10
C      NC2=0
      DO K=1,(NLT1-NCST)
         I=NIMIN(K)
         DO JJ=1,NLA2
          IF (NPNLA2(JJ).EQ.K) GOTO
301
         ENDDO

VOLK(I)=KZBLEAK(XLS(I),YLS(I),ZLS(I)
)*

>ABS(HEAD(XLS(I),YLS(I),ZLS(I))-
HEAD(XLS(I),YLS(I),ZLSU(I)))/
     >DZBLEAK(I)
       DAZL(I)=VOLK(I)*(STIME(NIT)-
STIME(NIT-1))

SUM1(I)=SUM1(I)+(DAZL(I)/VOLK(I))
       SUM2(I)=SUM2(I)+DAZL(I)
       AZL(I)=SUM2(I)

            IF
(AZL(I).GE.DZBLEAK(I)) THEN
              SUM1(I)=SUM1(I)-
(DAZL(I)/VOLK(I))
              SUM2(I)=SUM2(I)-
DAZL(I)
              DAZL(I)=DZBLEAK(I)-
SUM2(I)
              STIME(NIT)=STIME(NIT-
1)+(DAZL(I)/VOLK(I))

APTOLK(I)=SUM1(I)+(DAZL(I)/VOLK(I))
              NLA2=NLA2+1
              NLT2=NLA2+NCST
              XAT(NLT2)=XLS(I)
              YAT(NLT2)=YLS(I)
              ZAT(NLT2)=ZLS(I)
              ZATU(NLT2)=ZLSU(I)
              NPNLA2(NLA2)=K
            ENDIF
  301  ENDDO

      RETURN
      END

C
#####################################
#####################################
#
      SUBROUTINE
INTERPOL(NTB1,NTB2,TSBC1,TSBC2,NXHB,
NYHB,NZHB,NXSS,

>NYSS,NZSS,BHEAD,BFTI,HEAD,W,NCHB,NS
ST,DT,ET,NLC,NIT,STIME,ST)
C  THIS IS FOR INTERPOLATION OF
CONSTANT HEAD BOUNDARY CONDITIONS
AND SINK/
C  SOURCE TERMS IN FORWARD MODEL.
THE DATA OF HYDRAULIC HEAD OF
BOUNDARY
C  CONDITION AND RECHARGE OR
DISCHARGE FLUX OF SINK/SOURCE TERM
MUST BE GIVEN
C  AT SAME POINTS WITH CALCULATION
POINTS IN SPACE AND TIME.
C  HOWEVER, IF NOT SO, THE DATA FOR
BOUNDARY CONDITIONS MUST BE SHIFTED
IN SPACE
```

```
C  AND TIME TO MATCH WITH
CALCULATION POINTS. IN CASE OF
SPACE, LOCATIONS GIVEN
C  IN INPUT FOR BOUNDARY CONDITIONS
CAN BE BELIEVED TO BE SAME WITH
CALCULATION
C  POINTS. THUS, I USED TIMELINE
LINEAR INTERPOLATION TO MATCH
BOUNDARY CONDITIONS
C  WITH CALCULATION POINTS IN TIME.
THIS METHOD CAN PROVIDE SOLUTIONS
SIMPLY. TO
C  GET MORE EXACT SOLUTIONS, CUBIC
SPLINE INTERPOLATION CAN BE APPLIED
LATER.
C
C   NEST: ESTIMATED TOTAL NUMBER OF
SIMULATION TIME STEP

      IMPLICIT REAL*8(A-H,O-Z)
      INCLUDE 'PARAMS.F'
      DIMENSION
TSBC1(NTIMES),TSBC2(NTIMES)
      DIMENSION NXHB(NNODES*NNODES)
      DIMENSION NYHB(NNODES*NNODES)
      DIMENSION NZHB(NNODES*NNODES)
      DIMENSION NXSS(NNODES*NNODES)
      DIMENSION NYSS(NNODES*NNODES)
      DIMENSION NZSS(NNODES*NNODES)
      DIMENSION
HEAD(NXNODE,NYNODE,NZNODE)
      DIMENSION
W(NXNODE,NYNODE,NZNODE)
      DIMENSION
BHEAD(NTIMES,NNODES*NNODES)
      DIMENSION
BFTI(NTIMES,NNODES*NNODES)
      DIMENSION
HM(NTIMES,NNODES*NNODES)
      DIMENSION
QM(NTIMES,NNODES*NNODES)
      DIMENSION STIME(NTIMES)

      DO I=1,NTB1
         DO J=1,NCHB
           HM(I,J)=BHEAD(I,J)
         ENDDO
      ENDDO
      DO I=1,NTB2
         DO J=1,NSST
           QM(I,J)=BFTI(I,J)
         ENDDO
      ENDDO

      NEST=DINT(ET/DT)+1+2*NLC

      IF(NEST .GT. NTIMES)THEN
        WRITE(*,'(A,I8)') ' THE
PARAMETER NTIMES=',NTIMES
        WRITE(*,'(A,I8)')
     >    '  IN PROGRAM, NTIMES
SHOULD BE INCREASED AT LEAST TO:'
        WRITE(*,'(A,I8)') '
NTIMES=',NEST

WRITE(*,'(A,F8.3)')'CALCULATED TIME
STEP =', DT
C        WRITE(*,'(A)')' NOTE:
NUMBER OF MEASUREMENT DATA
DETERMINED THAT
C     >   IT IS DIVIDED WITH
CALCULATED TIME STEP'
        STOP
      ENDIF

      TC=STIME(NIT)

C  START REVISING OLD VERSION
C   CALCULATE LINEAR INTERPOLATION
C        DO K=1,NCHB
C          DO I=1,(NTB1-1)
C              IF
((TC.GT.TSBC1(I)).AND.(TC.LT.TSBC1(I
+1))) THEN
C
HEAD(NXHB(K),NYHB(K),NZHB(K))=((HM(I
+1,K)-HM(I,K))*
C     >          (TC-
TSBC1(I)))/(TSBC1(I+1)-
TSBC1(I))+HM(I,K)
C              ELSEIF
(TC.EQ.TSBC1(I)) THEN
C
HEAD(NXHB(K),NYHB(K),NZHB(K))=HM(I,K
)
C              ELSEIF
(TC.EQ.TSBC1(NTB1)) THEN
C
HEAD(NXHB(K),NYHB(K),NZHB(K))=HM(NTB
1,K)
C              ENDIF
C          ENDDO
C        ENDDO
C  END OF REVISING OLD VERSION
C  V.2.5
        DO K=1,NCHB

HEAD(NXHB(K),NYHB(K),NZHB(K))=HM(NTB
1,K)
        ENDDO
```

```
C V.2.5

        DO K=1,NSST
          DO I=1,(NTB2-1)
              IF
((TC.GT.TSBC2(I)).AND.(TC.LT.TSBC2(I
+1))) THEN

W(NXSS(K),NYSS(K),NZSS(K))=((QM(I+1,
K)-QM(I,K))*
     >          (TC-
TSBC2(I)))/(TSBC2(I+1)-
TSBC2(I))+QM(I,K)
              ELSEIF
(TC.EQ.TSBC2(I)) THEN

W(NXSS(K),NYSS(K),NZSS(K))=QM(I,K)
              ELSEIF
(TC.EQ.TSBC2(NTB2)) THEN

W(NXSS(K),NYSS(K),NZSS(K))=QM(NTB2,K
)
              ENDIF
          ENDDO
         ENDDO
C       ENDIF

       RETURN
       END

C
####################################
####################################
#
```

APPENDIX I

UPSCALING SIMULATOR FOR FORWARD MODELING

```
program upscale_forward_statistics

c  Program to automatically make
input of tough2 from King's
equation. Addionally, calculate
statistics of renormalized
permeability after twice
applications.
c  This program needs two input
files. First, a mesh file generated
from tough2
c  using mesh.dat. Second, another
tough2 input file with from rocks,
multi,
c  selec, solvr, start, param, and
times blocks. Here permeability
values of rocks block
c  are revised to average
(arithmetic) permeability values for
each group with the same
c  permeability value. So this
program generates an input file for
each group with the same
c  average (arithmetic) permeability
values for each group and for one
leakage pathway.
c  However number of grids are
103*103*11, that is, this program
arranges average permeability
c  values for sections in whole
grids (103*103*11 with grid size
100m*100m*100m).
c  This simulation is to arrange the
same permeability for 24 sections
(arbitrarily divided)
c  in overlying and storage
formations respectively with
103*103*11 grids. so rock block
c  consists of 12 overlying
sections, 12 storage sections, 1
shale section, and 1 leakage
pathway.
c  One section of overlying and
storage formations has the same
permeability regardless of z-
direction (depth).
c
c  Gener and foft blocks should be
filled out manually in output file
from this
c  program.

      IMPLICIT REAL*8(A-H,O-Z)

      character*80 title
      character*80 title1,title2
      character*5 cord(500000)
C      character*5 cord2(500000)
      CHARACTER*80 CONN(500000),
POINP(500000)
      dimension ni(500000)
      dimension
xc(500000),yc(500000),zc(500000)
C      real mop(500000) ! modifier
of permeability
      DIMENSION
XDD(200000),YDD(200000),ZDD(200000)
      dimension
ni2(500000),area(500000),vol(500000)
      dimension
xc2(500000),yc2(500000),zc2(500000)
      dimension
xperm(110,110,15),yperm(110,110,15),
zperm(110,110,15)
      dimension
fxperm(110,110,15),fyperm(110,110,15
),fzperm(110,110,15)
      dimension
sxperm(110,110,15),syperm(110,110,15
),szperm(110,110,15)
      dimension
gsxperm(110,110,15),gsyperm(110,110,
15)
      dimension gszperm(110,110,15)
      DIMENSION
PORO(200000),TEMP(200000),SALT(20000
0),CO2(200000)
      DIMENSION
HEAD(200,200,100),PR(200000)
      dimension
ng(20),nni(110,110,15)
      dimension
GXP(500000),GYP(500000),GZP(500000)
      dimension
AGXP(500000),AGYP(500000),AGZP(50000
0)
c      dimension zm(100,10000)
      DIMENSION DX(26),DY(26),DZ(11)
! DX,DY,DZ: final scaled blocks
c   DX,DY,DZ: number of each
directional blocks with the same
permeability
c   That is, the interval of
upscaled blocks with the same
permeability
c   (number of red and blue lines in
my worksheet (fw_9_4).
c   IMPORTANT!! This is not interval
of each cell in upscaled simulation
domain !!
```

```
      DATA DX
/400.,400.,400.,400.,400.,400.,400.,
400.,400.,400.,

>400.,400.,400.,200.,400.,400.,400.,
400.,400.,400.,
     >400.,400.,400.,400.,400.,300./
      DATA DY
/400.,400.,400.,400.,400.,400.,400.,
400.,400.,400.,

>400.,400.,400.,400.,400.,200.,400.,
400.,400.,400.,
     >400.,400.,400.,400.,400.,300./
      DATA DZ
/20.0,20.0,20.0,20.0,20.0,20.0,20.0,
20.0,20.0,20.0,20.0/

      NX=103  ! NUMBER OF X-
COORDINATE of original domain
      NY=103  ! NUMBER OF Y-
COORDINATE of original domain
      NZ=11  ! NUMBER OF Z-
COORDINATE of original domain
      nx1=26 ! # of x-directional
just upcale blocks
      ny1=26 ! # of y-directional
just upcale blocks
      nz1=11  ! # of z-directional
just upcale blocks
      nx2=33 ! number of x-
coordinate of cells in upscaled
domain
      ny2=42 ! number of y-
coordinate of cells in upscaled
domain
      nz2=11 ! number of z-
coordinate of cells in upscaled
domain
      NLCON=58776  !TOTAL NUMBER OF
LINE IN MESH FILE of upscaled domain
      NLINP=74  ! TOTAL NUMBER OF
LINE IN INPUT_PROPERTY FILE
      ELV=-1000. ! ELEVATION OF TOP
FROM ORIGIN
      UTEMP=50.0 ! TEMPERATURE AT
TOP
      NS=8  ! NUMBER OF GROUPS WITH
SAME AVERAGE PERMEABILITY

      NC=NX2*NY2*NZ2 ! NUMBER OF
CELLS
      open(5,
file='mesh_fw_9_3.dat',
status='unknown')
      OPEN(7,
FILE='input_property_fw_9_3.dat',
STATUS='UNKNOWN')
      OPEN(8,
FILE='perm_distribution_9_1.dat',
STATUS='UNKNOWN')
      open(6,
file='input_fw_9_3.out',
status='unknown')

open(9,file='no_medium_upscale_fw_9_
3.out',status='unknown')
      open(10,
file='medium_upscale_fw_9_3.out',
status='unknown')
      open(11,
file='statistics_fw_9_3.out',
status='unknown')
c  read original perm values from
file: perm_distribution_9_1.dat
      read(8,'(a)') title1
      read(8,'(a)') title2
      nog=0
      sumx1=0.
      sumy1=0.
      sumz1=0.
      DO I=1,nx
        DO J=1,ny
          DO K=1,nz
          nog=nog+1
      read(8,*)
xc(i),yc(j),zc(k),xperm(i,j,k),yperm
(i,j,k),zperm(i,j,k)
      sumx1=sumx1+xperm(i,j,k)
      sumy1=sumy1+yperm(i,j,k)
      sumz1=sumz1+zperm(i,j,k)
          enddo
        enddo
      enddo
      x1mean=sumx1/float(nx*ny*nz)
      y1mean=sumy1/float(nx*ny*nz)
      z1mean=sumz1/float(nx*ny*nz)

      ssumx1=0.
      ssumy1=0.
      ssumz1=0.
      do i=1,nx
        do j=1,ny
          do k=1,nz
        ssumx1=ssumx1+(xperm(i,j,k)-
x1mean)**2.
        ssumy1=ssumy1+(yperm(i,j,k)-
y1mean)**2.
        ssumz1=ssumz1+(zperm(i,j,k)-
z1mean)**2.
```

```
          enddo
        enddo
      enddo

stdx1=sqrt(ssumx1/float(nx*ny*nz))

stdy1=sqrt(ssumy1/float(nx*ny*nz))

stdz1=sqrt(ssumz1/float(nx*ny*nz))
      write(11,'(a)') ' 1st step
Arithmetic mean          std. dev. '
      write(11,'(a)') ' xperm
yperm     zperm       xperm
yperm
     >  zperm'
      write(11,*)
x1mean,y1mean,z1mean,stdx1,stdy1,std
z1
      write(11,*)


C  READ MESH FILE of upscaled domain
      read(5,'(a)') title
      DO I=1,NLCON-1
       IF (I.LE.NC) THEN

read(5,'(a,13x,i2,2e10.4,10x,3f10.3)
') cord(i),ni2(i),vol(i),
     >area(i),xc2(i),yc2(i),zc2(i)
        ELSE
          READ(5,'(A)') CONN(I)
        ENDIF
       ENDDO

c  Calculate x,y,z-perm of upcaled
domain by renormalization.
c  This procedure needs two times
because original domain has
100m*100m*20m grid.
c  I want 400m*400m*20m for normal
grids, 400m*400m*120m for caprock
grids.
c  So renormalization performs two
times,
c  that is, 100m*100m*20m ->
200m*200m*20m -> 400m*400m*20m.
c  Start first upscale !!
      nox=0
      noy=0
      noz=0
      in=int(nx/2)
      jn=int(ny/2)
c      kn=int(nz/2)
      DO I=1,nx,2
        nox=nox+1
        DO J=1,ny,2
          noy=noy+1
          DO K=1,nz

       if (k.le.2) then
         noz=noz+1
          if
((nox.le.in).and.(noy.le.jn)) then
            goto 10

           elseif
((nox.gt.in).and.(noy.gt.jn)) then

fxperm(nox,noy,noz)=xperm(i,j,k)

fyperm(nox,noy,noz)=yperm(i,j,k)

fzperm(nox,noy,noz)=zperm(i,j,k)
            goto 20

           elseif (nox.gt.in) then

xperm(i+1,j,k)=xperm(i,j,k)

xperm(i+1,j+1,k)=xperm(i,j+1,k)

yperm(i+1,j,k)=yperm(i,j,k)

yperm(i+1,j+1,k)=yperm(i,j+1,k)

zperm(i+1,j,k)=zperm(i,j,k)

zperm(i+1,j+1,k)=zperm(i,j+1,k)
            goto 10

           elseif (noy.gt.jn) then

xperm(i+1,j+1,k)=xperm(i+1,j,k)

xperm(i,j+1,k)=xperm(i,j,k)

yperm(i+1,j+1,k)=yperm(i+1,j,k)

yperm(i,j+1,k)=yperm(i,j,k)

zperm(i+1,j+1,k)=zperm(i+1,j,k)

zperm(i,j+1,k)=zperm(i,j,k)
            goto 10
          endif

       elseif
((k.gt.2).and.(k.lt.9)) then
          noz=noz+1
```

```
           fxperm(nox,noy,noz)=1.0e-
20
           fyperm(nox,noy,noz)=1.0e-
20
           fzperm(nox,noy,noz)=1.0e-
20
            goto 20

       elseif (k.ge.9) then
         noz=noz+1
          if
((nox.le.in).and.(noy.le.jn)) then
            goto 10

           elseif
((nox.gt.in).and.(noy.gt.jn)) then

fxperm(nox,noy,noz)=xperm(i,j,k)

fyperm(nox,noy,noz)=yperm(i,j,k)

fzperm(nox,noy,noz)=zperm(i,j,k)
            goto 20

           elseif (nox.gt.in) then

xperm(i+1,j,k)=xperm(i,j,k)

xperm(i+1,j+1,k)=xperm(i,j+1,k)

yperm(i+1,j,k)=yperm(i,j,k)

yperm(i+1,j+1,k)=yperm(i,j+1,k)

zperm(i+1,j,k)=zperm(i,j,k)

zperm(i+1,j+1,k)=zperm(i,j+1,k)
            goto 10

           elseif (noy.gt.jn) then

xperm(i+1,j+1,k)=xperm(i+1,j,k)

xperm(i,j+1,k)=xperm(i,j,k)

yperm(i+1,j+1,k)=yperm(i+1,j,k)

yperm(i,j+1,k)=yperm(i,j,k)

zperm(i+1,j+1,k)=zperm(i+1,j,k)

zperm(i,j+1,k)=zperm(i,j,k)
            goto 10
          endif
       endif
```

```
  10
fxa=xperm(i,j,k)*xperm(i+1,j,k)*(xpe
rm(i,j+1,k)+xperm(i+1,j+1,k))

>+xperm(i,j+1,k)*xperm(i+1,j+1,k)*(x
perm(i,j,k)+xperm(i+1,j,k))
       fxb=3*(xperm(i+1,j,k)+

>xperm(i+1,j+1,k))*(xperm(i,j,k)+xpe
rm(i,j+1,k))*(xperm(i+1,j,k)+

>xperm(i,j,k))*(xperm(i+1,j+1,k)+xpe
rm(i,j+1,k))

fya=yperm(i,j,k)*yperm(i+1,j,k)*(ype
rm(i,j+1,k)+yperm(i+1,j+1,k))

>+yperm(i,j+1,k)*yperm(i+1,j+1,k)*(y
perm(i,j,k)+yperm(i+1,j,k))
       fyb=3*(yperm(i+1,j,k)+

>yperm(i+1,j+1,k))*(yperm(i,j,k)+ype
rm(i,j+1,k))*(yperm(i+1,j,k)+

>yperm(i,j,k))*(yperm(i+1,j+1,k)+ype
rm(i,j+1,k))

fza=zperm(i,j,k)*zperm(i+1,j,k)*(zpe
rm(i,j+1,k)+zperm(i+1,j+1,k))

>+zperm(i,j+1,k)*zperm(i+1,j+1,k)*(z
perm(i,j,k)+zperm(i+1,j,k))
       fzb=3*(zperm(i+1,j,k)+

>zperm(i+1,j+1,k))*(zperm(i,j,k)+zpe
rm(i,j+1,k))*(zperm(i+1,j,k)+

>zperm(i,j,k))*(zperm(i+1,j+1,k)+zpe
rm(i,j+1,k))


fxperm(nox,noy,noz)=4*(xperm(i+1,j,k
)+xperm(i,j,k))*

>(xperm(i+1,j+1,k)+xperm(i,j+1,k))*f
xa/(fxa*(xperm(i+1,j,k)+

>xperm(i+1,j+1,k)+xperm(i,j,k)+xperm
(i,j+1,k))+fxb)

fyperm(nox,noy,noz)=4*(yperm(i+1,j,k
)+yperm(i,j,k))*
```

```
>(yperm(i+1,j+1,k)+yperm(i,j+1,k))*f
ya/(fya*(yperm(i+1,j,k)+

>yperm(i+1,j+1,k)+yperm(i,j,k)+yperm
(i,j+1,k))+fyb)

fzperm(nox,noy,noz)=4*(zperm(i+1,j,k
)+zperm(i,j,k))*

>(zperm(i+1,j+1,k)+zperm(i,j+1,k))*f
za/(fza*(zperm(i+1,j,k)+

>zperm(i+1,j+1,k)+zperm(i,j,k)+zperm
(i,j+1,k))+fzb)

c      write(*,*) nox,noy,noz,
xperm(i,j,k),xperm(i,j+1,k),xperm(i+
1,j,k)
c
>,xperm(i+1,j+1,k),fxperm(nox,noy,no
z)
c      write(*,*) nox,noy,noz,
yperm(i,j,k),yperm(i,j+1,k),yperm(i+
1,j,k)
c
>,yperm(i+1,j+1,k),fyperm(nox,noy,no
z)
       if
(fxperm(nox,noy,noz).ne.fyperm(nox,n
oy,noz)) then
       write(*,*) 'xperm is not
equal to yperm, please check input
or
     >logic in first upscale !!'
       stop
       endif
 20         enddo
          nozz=noz
          noz=0
         enddo
         noyy=noy
         noy=0
       enddo
       noy=noyy
       noz=nozz

       write(*,'(a,3i5)') '# of
(x,y,z) grids from first upscale =',
     >nox, noy, noz

       sumx2=0.
       sumy2=0.
       sumz2=0.
       DO I=1,nox
         DO J=1,noy
           DO K=1,noz
          sumx2=sumx2+fxperm(i,j,k)
          sumy2=sumy2+fyperm(i,j,k)
          sumz2=sumz2+fzperm(i,j,k)
           enddo
          enddo
        enddo

x2mean=sumx2/float(nox*noy*noz)

y2mean=sumy2/float(nox*noy*noz)

z2mean=sumz2/float(nox*noy*noz)

      ssumx2=0.
      ssumy2=0.
      ssumz2=0.
      do i=1,nox
        do j=1,noy
          do k=1,noz

ssumx2=ssumx2+(fxperm(i,j,k)-
x2mean)**2.

ssumy2=ssumy2+(fyperm(i,j,k)-
y2mean)**2.

ssumz2=ssumz2+(fzperm(i,j,k)-
z2mean)**2.
          enddo
        enddo
      enddo

stdx2=sqrt(ssumx2/float(nox*noy*noz)
)

stdy2=sqrt(ssumy2/float(nox*noy*noz)
)

stdz2=sqrt(ssumz2/float(nox*noy*noz)
)
      write(11,'(a)') ' 2nd step
Arithmetic mean          std. dev. '
      write(11,'(a)') ' fxperm
fyperm    fzperm      fxperm
fyperm
     >   fzperm'
      write(11,*)
x2mean,y2mean,z2mean,stdx2,stdy2,std
z2
      write(11,*)
```

```
c  End of first upscale !!

c  Start second upscale !!
      nox1=0
      noy1=0
      noz1=0
      in=int(nox/2)
      jn=int(noy/2)
c      kn=int(nz/2)
      DO I=1,nox,2
        nox1=nox1+1
        DO J=1,noy,2
          noy1=noy1+1
          DO K=1,noz

       if (k.le.2) then
         noz1=noz1+1
          if
((nox1.le.in).and.(noy1.le.jn)) then
            goto 30

           elseif
((nox1.gt.in).and.(noy1.gt.jn)) then

sxperm(nox1,noy1,noz1)=fxperm(i,j,k)

syperm(nox1,noy1,noz1)=fyperm(i,j,k)

szperm(nox1,noy1,noz1)=fzperm(i,j,k)
            goto 40

           elseif (nox1.gt.in) then

fxperm(i+1,j,k)=fxperm(i,j,k)

fxperm(i+1,j+1,k)=fxperm(i,j+1,k)

fyperm(i+1,j,k)=fyperm(i,j,k)

fyperm(i+1,j+1,k)=fyperm(i,j+1,k)

fzperm(i+1,j,k)=fzperm(i,j,k)

fzperm(i+1,j+1,k)=fzperm(i,j+1,k)
            goto 30

           elseif (noy1.gt.jn) then

fxperm(i+1,j+1,k)=fxperm(i+1,j,k)

fxperm(i,j+1,k)=fxperm(i,j,k)

fyperm(i+1,j+1,k)=fyperm(i+1,j,k)

fyperm(i,j+1,k)=fyperm(i,j,k)

fzperm(i+1,j+1,k)=fzperm(i+1,j,k)

fzperm(i,j+1,k)=fzperm(i,j,k)
            goto 30
          endif

       elseif
((k.gt.2).and.(k.lt.9)) then
          noz1=noz1+1

sxperm(nox1,noy1,noz1)=1.0e-20

syperm(nox1,noy1,noz1)=1.0e-20

szperm(nox1,noy1,noz1)=1.0e-20
            goto 40

       elseif (k.ge.9) then
         noz1=noz1+1
          if
((nox1.le.in).and.(noy1.le.jn)) then
            goto 30

           elseif
((nox1.gt.in).and.(noy1.gt.jn)) then

sxperm(nox1,noy1,noz1)=fxperm(i,j,k)

syperm(nox1,noy1,noz1)=fyperm(i,j,k)

szperm(nox1,noy1,noz1)=fzperm(i,j,k)
            goto 40

           elseif (nox1.gt.in) then

fxperm(i+1,j,k)=fxperm(i,j,k)

fxperm(i+1,j+1,k)=fxperm(i,j+1,k)

fyperm(i+1,j,k)=fyperm(i,j,k)

fyperm(i+1,j+1,k)=fyperm(i,j+1,k)

fzperm(i+1,j,k)=fzperm(i,j,k)

fzperm(i+1,j+1,k)=fzperm(i,j+1,k)
            goto 30

           elseif (noy1.gt.jn) then

fxperm(i+1,j+1,k)=fxperm(i+1,j,k)

fxperm(i,j+1,k)=fxperm(i,j,k)
```

```
fyperm(i+1,j+1,k)=fyperm(i+1,j,k)

fyperm(i,j+1,k)=fyperm(i,j,k)

fzperm(i+1,j+1,k)=fzperm(i+1,j,k)

fzperm(i,j+1,k)=fzperm(i,j,k)
            goto 30
          endif
       endif

  30
sxa=fxperm(i,j,k)*fxperm(i+1,j,k)*(f
xperm(i,j+1,k)+
     >fxperm(i+1,j+1,k))

>+fxperm(i,j+1,k)*fxperm(i+1,j+1,k)*
(fxperm(i,j,k)+fxperm(i+1,j,k))

sxb=3*(fxperm(i+1,j,k)+fxperm(i+1,j+
1,k))*(fxperm(i,j,k)+

>fxperm(i,j+1,k))*(fxperm(i+1,j,k)+

>fxperm(i,j,k))*(fxperm(i+1,j+1,k)+f
xperm(i,j+1,k))

sya=fyperm(i,j,k)*fyperm(i+1,j,k)*(f
yperm(i,j+1,k)+
     >fyperm(i+1,j+1,k))

>+fyperm(i,j+1,k)*fyperm(i+1,j+1,k)*
(fyperm(i,j,k)+fyperm(i+1,j,k))

syb=3*(fyperm(i+1,j,k)+fyperm(i+1,j+
1,k))*(fyperm(i,j,k)+

>fyperm(i,j+1,k))*(fyperm(i+1,j,k)+

>fyperm(i,j,k))*(fyperm(i+1,j+1,k)+f
yperm(i,j+1,k))

sza=fzperm(i,j,k)*fzperm(i+1,j,k)*(f
zperm(i,j+1,k)+
     >fzperm(i+1,j+1,k))

>+fzperm(i,j+1,k)*fzperm(i+1,j+1,k)*
(fzperm(i,j,k)+fzperm(i+1,j,k))

szb=3*(fzperm(i+1,j,k)+fzperm(i+1,j+
1,k))*(fzperm(i,j,k)+

>fzperm(i,j+1,k))*(fzperm(i+1,j,k)+
>fzperm(i,j,k))*(fzperm(i+1,j+1,k)+f
zperm(i,j+1,k))


sxperm(nox1,noy1,noz1)=4*(fxperm(i+1
,j,k)+fxperm(i,j,k))*

>(fxperm(i+1,j+1,k)+fxperm(i,j+1,k))
*sxa/(sxa*
     >(fxperm(i+1,j,k)+

>fxperm(i+1,j+1,k)+fxperm(i,j,k)+fxp
erm(i,j+1,k))+sxb)

syperm(nox1,noy1,noz1)=4*(fyperm(i+1
,j,k)+fyperm(i,j,k))*

>(fyperm(i+1,j+1,k)+fyperm(i,j+1,k))
*sya/(sya*
     >(fyperm(i+1,j,k)+

>fyperm(i+1,j+1,k)+fyperm(i,j,k)+fyp
erm(i,j+1,k))+syb)

szperm(nox1,noy1,noz1)=4*(fzperm(i+1
,j,k)+fzperm(i,j,k))*

>(fzperm(i+1,j+1,k)+fzperm(i,j+1,k))
*sza/(sza*
     >(fzperm(i+1,j,k)+

>fzperm(i+1,j+1,k)+fzperm(i,j,k)+fzp
erm(i,j+1,k))+szb)

c      write(*,*)
nox1,noy1,noz1,fxperm(i,j,k),fxperm(
i,j+1,k),
c
>fxperm(i+1,j,k),fxperm(i+1,j+1,k),s
xperm(nox1,noy1,noz1)
       if
(sxperm(nox1,noy1,noz1).ne.syperm(no
x1,noy1,noz1)) then
       write(*,*) 'xperm is not
equal to yperm, please check input
or
     >logic in second upscale !!'
       stop
       endif
 40        enddo
          nozz1=noz1
          noz1=0
         enddo
         noyy1=noy1
```

```
        noy1=0
       enddo
        noy1=noyy1
        noz1=nozz1

        write(*,'(a,3i5)')
     >'# of (x,y,z) grids from
second upscale =', nox1, noy1, noz1

        if
((nox1.ne.nx1).or.(noy1.ne.ny1).or.(
noz1.ne.nz1))then
        write(*,*) 'Error !! Please
check program logic and input'
        stop
        endif

        sumx3=0.
        sumy3=0.
        sumz3=0.
        DO I=1,nox1
          DO J=1,noy1
            DO K=1,noz1
           sumx3=sumx3+sxperm(i,j,k)
           sumy3=sumy3+syperm(i,j,k)
           sumz3=sumz3+szperm(i,j,k)
            enddo
           enddo
         enddo

x3mean=sumx3/float(nox1*noy1*noz1)

y3mean=sumy3/float(nox1*noy1*noz1)

z3mean=sumz3/float(nox1*noy1*noz1)

       ssumx3=0.
       ssumy3=0.
       ssumz3=0.
       do i=1,nox1
         do j=1,noy1
           do k=1,noz1

ssumx3=ssumx3+(sxperm(i,j,k)-
x3mean)**2.

ssumy3=ssumy3+(syperm(i,j,k)-
y3mean)**2.

ssumz3=ssumz3+(szperm(i,j,k)-
z3mean)**2.
           enddo
         enddo
       enddo

stdx3=sqrt(ssumx3/float(nox1*noy1*no
z1))

stdy3=sqrt(ssumy3/float(nox1*noy1*no
z1))

stdz3=sqrt(ssumz3/float(nox1*noy1*no
z1))
       write(11,'(a)') ' 3rd step
Arithmetic mean          std. dev. '
       write(11,'(a)') ' sxperm
syperm    szperm      sxperm
syperm
     >   szperm'
       write(11,*)
x3mean,y3mean,z3mean,stdx3,stdy3,std
z3
       write(11,*)

c  Calculate normalized permeability
of blocks within the ranges and
group (arrange)
c  the blocks included in the given
ranges.
       do i=1,NS   ! Divide the
blocks to 8 groups with same x,y-
perm
         ng(i)=0
       enddo

       Do i=1,nx1
         do j=1,ny1
           do k=1,nz1
       if ((k.le.2).or.(k.ge.9)) then
c          if
((sxperm(i,j,k).le.1.0e-
12).and.(sxperm(i,j,k).gt.
c     >5.0e-13))
c     >then
c          ng(1)=ng(1)+1 ! Counting
the block
c          nni(i,j,k)=1 ! numbering
block for ni(nk) below
c          gsxperm(i,j,k)=7.5e-13
c          gsyperm(i,j,k)=7.5e-13
c          gszperm(i,j,k)=4.99e-14

          if ((sxperm(i,j,k).le.5.0e-
13).and.(sxperm(i,j,k).gt.
     >1.0e-13))then
           ng(1)=ng(1)+1
           nni(i,j,k)=1
           gsxperm(i,j,k)=2.5e-13
           gsyperm(i,j,k)=2.5e-13
```

```
          gszperm(i,j,k)=1.84e-14

         elseif
((sxperm(i,j,k).le.1.0e-
13).and.(sxperm(i,j,k).gt.
     >5.0e-14))
     >then
          ng(2)=ng(2)+1 ! Counting
the block
          nni(i,j,k)=2 ! numbering
block for ni(nk) below
          gsxperm(i,j,k)=7.5e-14
          gsyperm(i,j,k)=7.5e-14
          gszperm(i,j,k)=4.99e-15

         elseif
((sxperm(i,j,k).le.5.0e-
14).and.(sxperm(i,j,k).gt.
     >1.0e-14))then
          ng(3)=ng(3)+1
          nni(i,j,k)=3
          gsxperm(i,j,k)=2.5e-14
          gsyperm(i,j,k)=2.5e-14
          gszperm(i,j,k)=1.84e-15

         elseif
((sxperm(i,j,k).le.1.0e-
14).and.(sxperm(i,j,k).gt.
     >5.0e-15))then
          ng(4)=ng(4)+1
          nni(i,j,k)=4
          gsxperm(i,j,k)=7.5e-15
          gsyperm(i,j,k)=7.5e-15
          gszperm(i,j,k)=4.99e-16

         elseif
((sxperm(i,j,k).le.5.0e-
15).and.(sxperm(i,j,k).gt.
     >1.0e-15))then
          ng(5)=ng(5)+1
          nni(i,j,k)=5
          gsxperm(i,j,k)=2.5e-15
          gsyperm(i,j,k)=2.5e-15
          gszperm(i,j,k)=1.84e-16

         elseif
((sxperm(i,j,k).le.1.0e-
15).and.(sxperm(i,j,k).gt.
     >5.0e-16))then
          ng(6)=ng(6)+1
          nni(i,j,k)=6
          gsxperm(i,j,k)=7.5e-16
          gsyperm(i,j,k)=7.5e-16
          gszperm(i,j,k)=4.99e-17

         elseif
((sxperm(i,j,k).le.5.0e-
16).and.(sxperm(i,j,k).gt.
     >1.0e-16))then
          ng(7)=ng(7)+1
          nni(i,j,k)=7
          gsxperm(i,j,k)=2.5e-16
          gsyperm(i,j,k)=2.5e-16
          gszperm(i,j,k)=1.84e-17

         elseif
((sxperm(i,j,k).le.1.0e-
16).and.(sxperm(i,j,k).gt.
     >5.0e-17))then
          ng(8)=ng(8)+1
          nni(i,j,k)=8
          gsxperm(i,j,k)=7.5e-17
          gsyperm(i,j,k)=7.5e-17
          gszperm(i,j,k)=4.99e-18

c         elseif
((sxperm(i,j,k).le.5.0e-
16).and.(sxperm(i,j,k).gt.
c     >1.0e-16))then
c          ng(10)=ng(10)+1
c          nni(i,j,k)=10
c          gsxperm(i,j,k)=2.5e-16
c          gsyperm(i,j,k)=2.5e-16

c         elseif
((sxperm(i,j,k).le.1.0e-
16).and.(sxperm(i,j,k).gt.
c     >5.0e-17))then
c          ng(11)=ng(11)+1
c          nni(i,j,k)=11
c          gsxperm(i,j,k)=7.5e-17
c          gsyperm(i,j,k)=7.5e-17

c         elseif
((sxperm(i,j,k).le.5.0e-
17).and.(sxperm(i,j,k).gt.
c     >1.0e-17))then
c          ng(12)=ng(12)+1
c          nni(i,j,k)=12
c          gsxperm(i,j,k)=2.5e-17
c          gsyperm(i,j,k)=2.5e-17

c         elseif
((sxperm(i,j,k).le.1.0e-
17).and.(sxperm(i,j,k).gt.
c     >5.0e-18))then
c          ng(13)=ng(13)+1
c          nni(i,j,k)=13
c          gsxperm(i,j,k)=7.5e-18
c          gsyperm(i,j,k)=7.5e-18
```

```
c         elseif
((sxperm(i,j,k).le.5.0e-
18).and.(sxperm(i,j,k).gt.
c     >1.0e-18))then
c          ng(14)=ng(14)+1
c          nni(i,j,k)=14
c          gsxperm(i,j,k)=2.5e-18
c          gsyperm(i,j,k)=2.5e-18
         endif


elseif((k.gt.2).and.(k.lt.9))then
          gsxperm(i,j,k)=1.0e-20
          gsyperm(i,j,k)=1.0e-20
          gszperm(i,j,k)=1.0e-20
      endif
          enddo
        enddo
      enddo

c        do i=1,ns
c          do j=1,ng(i)
c          ncc(i)=1
c          zzm(i,ncc(i))=zm(i,1)
c          zzm(i)=0.
c           enddo
c        enddo
c          do i=1,nx1
c             do j=1,ny1
c               do k=1,nz1
c                  do ii=1,NS
c             zzm(ncc(ii))=zm(ii,1)
c                   do jj=1,ng(ii)
c                      do kk=1,ng(ii)
c
if(zzm(ii,ncc(ii)).eq.zm(ii,kk))then
c           ncc(ii)=ncc(ii)+1
c           zzm(kk)=zm(ii,jj)
c          write(*,*) zzm

c        endif
c       enddo

      nsum=0
      do i=1,NS  ! Calculate number
of divided group
        nsum=nsum+ng(i)
      enddo
       write(*,'(A)')'  Group #
# of blocks'
        do i=1,ns
           write(*,'(i6,5x,i6)') i,
ng(i)
        enddo
       write(*,'(A,i6)')'  Total #
of blocks =  ',nsum

C SPECIFY X,Y-DIRECTIONAL DISTANCE
and numbering OF BLOCKS FROM ORIGIN
c This is not based on block
centers, but this is based on line.
c A line on origin is 0, the first
line (x-axis) is 0m-400m, 2nd line
is 400m-800m....
c As the same way, 1st line (y-axis)
is 0-400m, 2nd line is 400-800m...
      XDD(1)=0. ! X-POINT OF FIRST
BLOCK
      YDD(1)=0.
      DO I=2,NX1+1
        XDD(I)=XDD(I-1)+DX(I-1)
      ENDDO
      DO J=2,NY1+1
        YDD(J)=YDD(J-1)+DY(J-1)
      ENDDO

c Assign renormalized each block for
each grid
      NK=0
      NCH=0
      do i=1,nx2
       do j=1,ny2
        do k=1,nz2
           nk=nk+1
           PORO(NK)=0.2 ! SANDSTONE
PERMEABILITY
           TEMP(NK)=50.0 ! INITIAL
TEMPERATURE
           SALT(NK)=0.05 ! INITIAL
SALT MASS FRACTION
           CO2(NK)=0.0   ! INITIAL
CO2 MASS FRACTION
         IF ((k.gt.2).and.(k.lt.9))
THEN ! Define shale layer
           ni(nk)=9  ! impermeable
layer
           PORO(NK)=0.02 ! SHALE
PERMEABILITY
           GXP(nk)=1.0e-20
           GYP(nk)=1.0e-20
           GZP(nk)=1.0e-20
           AGXP(nk)=1.0e-20
           AGYP(nk)=1.0e-20
           AGZP(nk)=1.0e-20
c            if
((i.eq.11).and.(j.eq.14)) then ! IG
of leakage pathway
```

```
c               ni(nk)=10 ! rock
property number of expected leakage
path
c               poro(nk)=0.2 !
porosity of leakage pathway
c             elseif
((i.eq.11).and.(j.eq.22)) then ! IG
of leakagepathway
c               ni(nk)=11   ! rock
property number of leakage pathway
c             elseif
((i.eq.11).and.(j.eq.29)) then ! IG
of leakagepathway
c               ni(nk)=12   ! rock
property number of leakage pathway
c             elseif
((i.eq.11).and.(j.eq.34)) then ! IG
of leakagepathway
c               ni(nk)=13   ! rock
property number of leakage pathway
c             endif
          elseif (K.le.2) then !
DEFINE NI IN OVERLYING FORM
           DO IG=1,NX1
            DO JG=1,NY1
             IF
(((XC2(nk).GE.XDD(IG)).AND.(XC2(nk).
LE.XDD(IG+1))).AND.

>((YC2(nk).GE.YDD(JG)).AND.(YC2(nk).
LE.YDD(JG+1)))) THEN
             GXP(nk)=sxperm(IG,JG,K)
             GYP(nk)=syperm(IG,JG,K)
             GZP(NK)=SZPERM(IG,JG,K)
             ni(nk)=nni(IG,JG,K)

AGXP(NK)=gsxperm(IG,JG,K)

AGYP(NK)=GSYPERM(IG,JG,K)

AGZP(NK)=GSZPERM(IG,JG,K)
             ENDIF
            ENDDO
           ENDDO
          elseif (k.ge.9) then !
DEFINE NI IN STORAGE FORM
           DO IG=1,NX1
            DO JG=1,NY1
             IF
(((XC2(nk).GE.XDD(IG)).AND.(XC2(nk).
LE.XDD(IG+1))).AND.

>((YC2(nk).GE.YDD(JG)).AND.(YC2(nk).
LE.YDD(JG+1)))) THEN
             GXP(nk)=sxperm(IG,JG,K)
             GYP(nk)=syperm(IG,JG,K)
             GZP(NK)=SZPERM(IG,JG,K)
             ni(nk)=nni(IG,JG,K)

AGXP(NK)=gsxperm(IG,JG,K)

AGYP(NK)=GSYPERM(IG,JG,K)

AGZP(NK)=GSZPERM(IG,JG,K)
             ENDIF
            ENDDO
           ENDDO
          ENDIF

          IF (J.EQ.1 .OR. J.EQ.NY2)
THEN ! DEFINE CONSTANT HEAD BOUNDARY
            VOL(NK)=0.2E66
            NCH=NCH+1
          ENDIF
         ENDDO
        ENDDO
       ENDDO

C SPECIFY ELEVATION OF CELLS (z-axis
is based on block centered scheme to
calculate
c initial condition
       ZDD(1)=-DZ(1)/2.+ELV
C       WRITE(*,*) ZDD(1)
       DO K=2,NZ2
        ZDD(K)=ZDD(K-1)-(DZ(K-
1)+DZ(K))/2.
C        WRITE(*,*) ZDD(K)
       ENDDO
C CALCULATE INITIAL HYDRAULIC HEAD
VALUES AT ALL CELLS
       DO I=1,NX2
         DO J=1,NY2
           DO K=1,NZ2

              HEAD(I,J,K)=10.0   !
Initial head of Brine

           ENDDO
         ENDDO
       ENDDO

C  CALCULATE PRESSURE AT ALL CELLS
FOR INITIAL CONDITION
       NK1=0
C       NCH=0
c       NOFLOW=0
       DO I=1,NX2
        DO J=1,NY2
         DO K=1,NZ2
```

```
          NK1=NK1+1

PR(NK1)=1025.*9.80665*(HEAD(I,J,K)-
ZDD(K))
c
TEMP(NK)=UTEMP+25.0/1000.0*(ZDD(1)-
ZDD(K))
        ENDDO
       ENDDO
      ENDDO

C  READ INPUT PROPERTY
      DO I=1,NLINP
       READ(7,'(A)') POINP(I) !
POINP : PROPERTY OF INPUT
      ENDDO

C  WRITE INPUT PROPERTY IN OUTPUT
FILE (input_fw_  .out)
      DO I=1,NLINP-1
       WRITE(6,'(A)') POINP(I)
      ENDDO

C  WRITE INITIAL CONDITION IN OUTPUT
FILE (input_fw_  .out)
C      WRITE(6,*)
      WRITE(6,'(A)') 'INCON'
      DO I=1,NC
       WRITE(6,'(A,10X,E15.8)')
CORD(I),PORO(I)
       WRITE(6,'(4E20.13)')
PR(I),SALT(I),CO2(I),TEMP(I)
C       WRITE(6,'(3E20.13)')
PRESS(I),VAP(I),TEMP(I)
      ENDDO

C  WRITE ELEME & CONNE BLOCK
       WRITE(6,*)
       write(6,'(a)') title
        do I=1,NC

WRITE(6,'(a,13x,i2,2e10.4,10x,3f10.3
)') cord(i),ni(i),vol(i),
     >area(i),xc2(i),yc2(i),zc2(i)
        ENDDO
        DO J=NC+1,NLCON-1
         WRITE(6,'(A)') CONN(J)
        ENDDO

C  WRITE INPUT PROPERTY
       WRITE(6,*)
       WRITE(6,'(A)') POINP(NLINP)
       WRITE(6,*)

c  write permability values
depanding on coordinate for techplot
form
       write(9,'(a)') 'Variables = X
Y Z X-PERM Y-PERM Z-PERM'
       write(9,'(a,a,i7)') 'Zone
F=POINT ','I=',nc
       write(10,'(a)') 'Variables =
X Y Z X-PERM Y-PERM Z-PERM'
       write(10,'(a,a,i7)') 'Zone
F=POINT ','I=',nc
       do i=1,nc !PRINTOUT REAL
UPSCALED PERM ON EACH GRIDS
       write(9,'(3f10.3,x,3E13.4)')
xc2(i),yc2(i),zc2(i),GXP(i),
     >GYP(i),GZP(i)
       enddo
       do i=1,nc !PRINTOUT MEDIUM
UPSCALED PERM ON EACH GRIDS
       write(10,'(3f10.3,x,3E13.4)')
xc2(i),yc2(i),zc2(i),AGXP(i),
     >AGYP(i),AGZP(i)
       enddo

      write(*,*) ' The number of
constant head boundary nodes= ', nch
      write(*,*)

      IF (NK.NE.NC) THEN
      WRITE(*,*)' PLEASE CHECK THE
NUMBER OF CELLS'
      STOP
      ENDIF

      STOP
      END
```

APPENDIX J

UPSCALING SIMULATOR FOR INVERSE MODELING

```
program upscaling_inverse_input

c  Program to automatically make
input of tough2 with eleme,
c  conne and incon blocks from
applications to King's equation for
twice.
c  This program needs two input
files. First, a mesh file generated
from tough2
c  using mesh.dat. Second, another
tough2 input file with from rocks,
multi,
c  selec, solvr, start, param, and
times blocks. Here permeability
values of rocks block
c  are revised to average
(arithmetic) permeability values for
each group with the same
c  permeability value. So this
program generates an input file for
each group with the same
c  average (arithmetic) permeability
values for each group and for one
leakage pathway.
c  However number of grids are
103*103*11, that is, this program
arranges average permeability
c  values for sections in whole
grids (103*103*11 with grid size
100m*100m*100m).
c  This simulation is to arrange the
same permeability for 24 sections
(arbitrarily divided)
c  in overlying and storage
formations respectively with
103*103*11 grids. so rock block
c  consists of 12 overlying
sections, 12 storage sections, 1
shale section, and 1 leakage
pathway.
c  One section of overlying and
storage formations has the same
permeability regardless of z-
direction (depth).
c
c  Gener and foft blocks should be
filled out manually in output file
from this
c  program.

      IMPLICIT REAL*8(A-H,O-Z)

      character*80 title
      character*80 title1,title2
      character*5 cord(500000)
C      character*5 cord2(500000)
      CHARACTER*80 CONN(500000),
POINP(500000)
      dimension ni(500000)
      dimension
xc(500000),yc(500000),zc(500000)
C      real mop(500000) ! modifier
of permeability
      DIMENSION
XDD(200000),YDD(200000),ZDD(200000)
      dimension
ni2(500000),area(500000),vol(500000)
      dimension
xc2(500000),yc2(500000),zc2(500000)
      dimension
xperm(105,105,12),yperm(105,105,12),
zperm(105,105,12)
      dimension
fxperm(105,105,12),fyperm(105,105,12
),fzperm(105,105,12)
      dimension
sxperm(105,105,12),syperm(105,105,12
),szperm(105,105,12)
      dimension
gsxperm(105,105,12),gsyperm(105,105,
12)
      dimension gszperm(105,105,12)
      DIMENSION
PORO(200000),TEMP(200000),SALT(20000
0),CO2(200000)
      DIMENSION
HEAD(200,200,100),PR(200000)
      dimension
ng(20),nni(110,110,15)
      dimension
GXP(500000),GYP(500000),GZP(500000)
      dimension
AGXP(500000),AGYP(500000),AGZP(50000
0)
c      dimension zm(100,10000)
      DIMENSION DX(26),DY(26),DZ(11)
! DX,DY,DZ: final scaled blocks
c   DX,DY,DZ: number of each
directional blocks with the same
permeability
c   That is, the interval of
upscaled blocks with the same
permeability
c   (number of red and blue lines in
my worksheet (fw_9_4).
c   IMPORTANT!! This is not interval
of each cell in upscaled simulation
domain !!
```

```
      DATA DX
/400.,400.,400.,400.,400.,400.,400.,
400.,400.,400.,

>400.,400.,400.,200.,400.,400.,400.,
400.,400.,400.,
     >400.,400.,400.,400.,400.,300./
      DATA DY
/400.,400.,400.,400.,400.,400.,400.,
400.,400.,400.,

>400.,400.,400.,400.,400.,200.,400.,
400.,400.,400.,
     >400.,400.,400.,400.,400.,300./
      DATA DZ
/20.0,20.0,20.0,20.0,20.0,20.0,20.0,
20.0,20.0,20.0,20.0/

      NX=103  ! NUMBER OF X-
COORDINATE of original domain
      NY=103  ! NUMBER OF Y-
COORDINATE of original domain
      NZ=11  ! NUMBER OF Z-
COORDINATE of original domain
      nx1=26 ! # of x-directional
just upcale blocks
      ny1=26 ! # of y-directional
just upcale blocks
      nz1=11  ! # of z-directional
just upcale blocks
      nx2=35 ! number of x-
coordinate of cells in upscaled
domain
      ny2=34 ! number of y-
coordinate of cells in upscaled
domain
      nz2=11 ! number of z-
coordinate of cells in upscaled
domain
      NLCON=50414  !TOTAL NUMBER OF
LINE IN MESH FILE of upscaled domain
      NLINP=78  ! TOTAL NUMBER OF
LINE IN INPUT_PROPERTY FILE
      ELV=-1000. ! ELEVATION OF TOP
FROM ORIGIN
      UTEMP=50.0 ! TEMPERATURE AT
TOP
      NS=8  ! NUMBER OF GROUPS WITH
SAME AVERAGE PERMEABILITY

      NC=NX2*NY2*NZ2 ! NUMBER OF
CELLS
      open(5,
file='mesh_inv_94_9.dat',
status='unknown')
      OPEN(7,
FILE='input_property_inv_94.dat',
STATUS='UNKNOWN')
      OPEN(8,
FILE='perm_distribution_9_1.dat',
STATUS='UNKNOWN')
      open(6,
file='input_inv_94_9.out',
status='unknown')

open(9,file='no_medium_upscale_inv_9
4.out',status='unknown')
      open(10,
file='medium_upscale_inv_94.out',
status='unknown')
c  read original perm values from
file: perm_distribution_9_1.dat
      read(8,'(a)') title1
      read(8,'(a)') title2
      nog=0
      DO I=1,nx
        DO J=1,ny
          DO K=1,nz
          nog=nog+1
      read(8,*)
xc(i),yc(j),zc(k),xperm(i,j,k),yperm
(i,j,k),zperm(i,j,k)
          enddo
        enddo
      enddo

C  READ MESH FILE of upscaled domain
      read(5,'(a)') title
      DO I=1,NLCON-1
       IF (I.LE.NC) THEN

read(5,'(a,13x,i2,2e10.4,10x,3f10.3)
') cord(i),ni2(i),vol(i),
     >area(i),xc2(i),yc2(i),zc2(i)
        ELSE
          READ(5,'(A)') CONN(I)
        ENDIF
      ENDDO

c  Calculate x,y,z-perm of upcaled
domain by renormalization.
c  This procedure needs two times
because original domain has
100m*100m*20m grid.
c  I want 400m*400m*20m for normal
grids, 400m*400m*120m for caprock
grids.
c  So renormalization performs two
times,
```

```
c  that is, 100m*100m*20m ->
200m*200m*20m -> 400m*400m*20m.
c  Start first upscale !!
      nox=0
      noy=0
      noz=0
      in=int(nx/2)
      jn=int(ny/2)
c      kn=int(nz/2)
      DO I=1,nx,2
        nox=nox+1
        DO J=1,ny,2
          noy=noy+1
          DO K=1,nz

       if (k.le.2) then
         noz=noz+1
          if
((nox.le.in).and.(noy.le.jn)) then
            goto 10

           elseif
((nox.gt.in).and.(noy.gt.jn)) then

fxperm(nox,noy,noz)=xperm(i,j,k)

fyperm(nox,noy,noz)=yperm(i,j,k)

fzperm(nox,noy,noz)=zperm(i,j,k)
            goto 20

           elseif (nox.gt.in) then

xperm(i+1,j,k)=xperm(i,j,k)

xperm(i+1,j+1,k)=xperm(i,j+1,k)

yperm(i+1,j,k)=yperm(i,j,k)

yperm(i+1,j+1,k)=yperm(i,j+1,k)

zperm(i+1,j,k)=zperm(i,j,k)

zperm(i+1,j+1,k)=zperm(i,j+1,k)
            goto 10

           elseif (noy.gt.jn) then

xperm(i+1,j+1,k)=xperm(i+1,j,k)

xperm(i,j+1,k)=xperm(i,j,k)

yperm(i+1,j+1,k)=yperm(i+1,j,k)

yperm(i,j+1,k)=yperm(i,j,k)

zperm(i+1,j+1,k)=zperm(i+1,j,k)

zperm(i,j+1,k)=zperm(i,j,k)
            goto 10
          endif

       elseif
((k.gt.2).and.(k.lt.9)) then
          noz=noz+1
           fxperm(nox,noy,noz)=1.0e-
20
           fyperm(nox,noy,noz)=1.0e-
20
           fzperm(nox,noy,noz)=1.0e-
20
            goto 20

       elseif (k.ge.9) then
         noz=noz+1
          if
((nox.le.in).and.(noy.le.jn)) then
            goto 10

           elseif
((nox.gt.in).and.(noy.gt.jn)) then

fxperm(nox,noy,noz)=xperm(i,j,k)

fyperm(nox,noy,noz)=yperm(i,j,k)

fzperm(nox,noy,noz)=zperm(i,j,k)
            goto 20

           elseif (nox.gt.in) then

xperm(i+1,j,k)=xperm(i,j,k)

xperm(i+1,j+1,k)=xperm(i,j+1,k)

yperm(i+1,j,k)=yperm(i,j,k)

yperm(i+1,j+1,k)=yperm(i,j+1,k)

zperm(i+1,j,k)=zperm(i,j,k)

zperm(i+1,j+1,k)=zperm(i,j+1,k)
            goto 10

           elseif (noy.gt.jn) then

xperm(i+1,j+1,k)=xperm(i+1,j,k)

xperm(i,j+1,k)=xperm(i,j,k)
```

```
yperm(i+1,j+1,k)=yperm(i+1,j,k)

yperm(i,j+1,k)=yperm(i,j,k)

zperm(i+1,j+1,k)=zperm(i+1,j,k)

zperm(i,j+1,k)=zperm(i,j,k)
            goto 10
          endif
       endif

  10
fxa=xperm(i,j,k)*xperm(i+1,j,k)*(xpe
rm(i,j+1,k)+xperm(i+1,j+1,k))

>+xperm(i,j+1,k)*xperm(i+1,j+1,k)*(x
perm(i,j,k)+xperm(i+1,j,k))
       fxb=3*(xperm(i+1,j,k)+

>xperm(i+1,j+1,k))*(xperm(i,j,k)+xpe
rm(i,j+1,k))*(xperm(i+1,j,k)+

>xperm(i,j,k))*(xperm(i+1,j+1,k)+xpe
rm(i,j+1,k))

fya=yperm(i,j,k)*yperm(i+1,j,k)*(ype
rm(i,j+1,k)+yperm(i+1,j+1,k))

>+yperm(i,j+1,k)*yperm(i+1,j+1,k)*(y
perm(i,j,k)+yperm(i+1,j,k))
       fyb=3*(yperm(i+1,j,k)+

>yperm(i+1,j+1,k))*(yperm(i,j,k)+ype
rm(i,j+1,k))*(yperm(i+1,j,k)+

>yperm(i,j,k))*(yperm(i+1,j+1,k)+ype
rm(i,j+1,k))

fza=zperm(i,j,k)*zperm(i+1,j,k)*(zpe
rm(i,j+1,k)+zperm(i+1,j+1,k))

>+zperm(i,j+1,k)*zperm(i+1,j+1,k)*(z
perm(i,j,k)+zperm(i+1,j,k))
       fzb=3*(zperm(i+1,j,k)+

>zperm(i+1,j+1,k))*(zperm(i,j,k)+zpe
rm(i,j+1,k))*(zperm(i+1,j,k)+

>zperm(i,j,k))*(zperm(i+1,j+1,k)+zpe
rm(i,j+1,k))


fxperm(nox,noy,noz)=4*(xperm(i+1,j,k
)+xperm(i,j,k))*

>(xperm(i+1,j+1,k)+xperm(i,j+1,k))*f
xa/(fxa*(xperm(i+1,j,k)+

>xperm(i+1,j+1,k)+xperm(i,j,k)+xperm
(i,j+1,k))+fxb)

fyperm(nox,noy,noz)=4*(yperm(i+1,j,k
)+yperm(i,j,k))*

>(yperm(i+1,j+1,k)+yperm(i,j+1,k))*f
ya/(fya*(yperm(i+1,j,k)+

>yperm(i+1,j+1,k)+yperm(i,j,k)+yperm
(i,j+1,k))+fyb)

fzperm(nox,noy,noz)=4*(zperm(i+1,j,k
)+zperm(i,j,k))*

>(zperm(i+1,j+1,k)+zperm(i,j+1,k))*f
za/(fza*(zperm(i+1,j,k)+

>zperm(i+1,j+1,k)+zperm(i,j,k)+zperm
(i,j+1,k))+fzb)

c      write(*,*) nox,noy,noz,
xperm(i,j,k),xperm(i,j+1,k),xperm(i+
1,j,k)
c
>,xperm(i+1,j+1,k),fxperm(nox,noy,no
z)
       if
(fxperm(nox,noy,noz).ne.fyperm(nox,n
oy,noz)) then
       write(*,*) 'xperm is not
equal to yperm, please check input
or
     >logic in first upscale !!'
       stop
       endif
 20         enddo
          nozz=noz
          noz=0
         enddo
         noyy=noy
         noy=0
       enddo
       noy=noyy
       noz=nozz

       write(*,'(a,3i5)') '# of
(x,y,z) grids from first upscale =',
     >nox, noy, noz

c  End of first upscale !!
```

```
c  Start second upscale !!
      nox1=0
      noy1=0
      noz1=0
      in=int(nox/2)
      jn=int(noy/2)
c      kn=int(nz/2)
      DO I=1,nox,2
        nox1=nox1+1
        DO J=1,noy,2
          noy1=noy1+1
          DO K=1,noz

       if (k.le.2) then
         noz1=noz1+1
          if
((nox1.le.in).and.(noy1.le.jn)) then
            goto 30

           elseif
((nox1.gt.in).and.(noy1.gt.jn)) then

sxperm(nox1,noy1,noz1)=fxperm(i,j,k)

syperm(nox1,noy1,noz1)=fyperm(i,j,k)

szperm(nox1,noy1,noz1)=fzperm(i,j,k)
            goto 40

           elseif (nox1.gt.in) then

fxperm(i+1,j,k)=fxperm(i,j,k)

fxperm(i+1,j+1,k)=fxperm(i,j+1,k)

fyperm(i+1,j,k)=fyperm(i,j,k)

fyperm(i+1,j+1,k)=fyperm(i,j+1,k)

fzperm(i+1,j,k)=fzperm(i,j,k)

fzperm(i+1,j+1,k)=fzperm(i,j+1,k)
            goto 30

           elseif (noy1.gt.jn) then

fxperm(i+1,j+1,k)=fxperm(i+1,j,k)

fxperm(i,j+1,k)=fxperm(i,j,k)

fyperm(i+1,j+1,k)=fyperm(i+1,j,k)

fyperm(i,j+1,k)=fyperm(i,j,k)

fzperm(i+1,j+1,k)=fzperm(i+1,j,k)

fzperm(i,j+1,k)=fzperm(i,j,k)
            goto 30
          endif

       elseif
((k.gt.2).and.(k.lt.9)) then
          noz1=noz1+1

sxperm(nox1,noy1,noz1)=1.0e-20

syperm(nox1,noy1,noz1)=1.0e-20

szperm(nox1,noy1,noz1)=1.0e-20
            goto 40

       elseif (k.ge.9) then
         noz1=noz1+1
          if
((nox1.le.in).and.(noy1.le.jn)) then
            goto 30

           elseif
((nox1.gt.in).and.(noy1.gt.jn)) then

sxperm(nox1,noy1,noz1)=fxperm(i,j,k)

syperm(nox1,noy1,noz1)=fyperm(i,j,k)

szperm(nox1,noy1,noz1)=fzperm(i,j,k)
            goto 40

           elseif (nox1.gt.in) then

fxperm(i+1,j,k)=fxperm(i,j,k)

fxperm(i+1,j+1,k)=fxperm(i,j+1,k)

fyperm(i+1,j,k)=fyperm(i,j,k)

fyperm(i+1,j+1,k)=fyperm(i,j+1,k)

fzperm(i+1,j,k)=fzperm(i,j,k)

fzperm(i+1,j+1,k)=fzperm(i,j+1,k)
            goto 30

           elseif (noy1.gt.jn) then

fxperm(i+1,j+1,k)=fxperm(i+1,j,k)

fxperm(i,j+1,k)=fxperm(i,j,k)
```

```
fyperm(i+1,j+1,k)=fyperm(i+1,j,k)

fyperm(i,j+1,k)=fyperm(i,j,k)

fzperm(i+1,j+1,k)=fzperm(i+1,j,k)

fzperm(i,j+1,k)=fzperm(i,j,k)
            goto 30
          endif
       endif

  30
sxa=fxperm(i,j,k)*fxperm(i+1,j,k)*(f
xperm(i,j+1,k)+
     >fxperm(i+1,j+1,k))

>+fxperm(i,j+1,k)*fxperm(i+1,j+1,k)*
(fxperm(i,j,k)+fxperm(i+1,j,k))

sxb=3*(fxperm(i+1,j,k)+fxperm(i+1,j+
1,k))*(fxperm(i,j,k)+

>fxperm(i,j+1,k))*(fxperm(i+1,j,k)+

>fxperm(i,j,k))*(fxperm(i+1,j+1,k)+f
xperm(i,j+1,k))

sya=fyperm(i,j,k)*fyperm(i+1,j,k)*(f
yperm(i,j+1,k)+
     >fyperm(i+1,j+1,k))

>+fyperm(i,j+1,k)*fyperm(i+1,j+1,k)*
(fyperm(i,j,k)+fyperm(i+1,j,k))

syb=3*(fyperm(i+1,j,k)+fyperm(i+1,j+
1,k))*(fyperm(i,j,k)+

>fyperm(i,j+1,k))*(fyperm(i+1,j,k)+

>fyperm(i,j,k))*(fyperm(i+1,j+1,k)+f
yperm(i,j+1,k))

sza=fzperm(i,j,k)*fzperm(i+1,j,k)*(f
zperm(i,j+1,k)+
     >fzperm(i+1,j+1,k))

>+fzperm(i,j+1,k)*fzperm(i+1,j+1,k)*
(fzperm(i,j,k)+fzperm(i+1,j,k))

szb=3*(fzperm(i+1,j,k)+fzperm(i+1,j+
1,k))*(fzperm(i,j,k)+

>fzperm(i,j+1,k))*(fzperm(i+1,j,k)+
>fzperm(i,j,k))*(fzperm(i+1,j+1,k)+f
zperm(i,j+1,k))


sxperm(nox1,noy1,noz1)=4*(fxperm(i+1
,j,k)+fxperm(i,j,k))*

>(fxperm(i+1,j+1,k)+fxperm(i,j+1,k))
*sxa/(sxa*
     >(fxperm(i+1,j,k)+

>fxperm(i+1,j+1,k)+fxperm(i,j,k)+fxp
erm(i,j+1,k))+sxb)

syperm(nox1,noy1,noz1)=4*(fyperm(i+1
,j,k)+fyperm(i,j,k))*

>(fyperm(i+1,j+1,k)+fyperm(i,j+1,k))
*sya/(sya*
     >(fyperm(i+1,j,k)+

>fyperm(i+1,j+1,k)+fyperm(i,j,k)+fyp
erm(i,j+1,k))+syb)

szperm(nox1,noy1,noz1)=4*(fzperm(i+1
,j,k)+fzperm(i,j,k))*

>(fzperm(i+1,j+1,k)+fzperm(i,j+1,k))
*sza/(sza*
     >(fzperm(i+1,j,k)+

>fzperm(i+1,j+1,k)+fzperm(i,j,k)+fzp
erm(i,j+1,k))+szb)

c      write(*,*)
nox1,noy1,noz1,fxperm(i,j,k),fxperm(
i,j+1,k),
c
>fxperm(i+1,j,k),fxperm(i+1,j+1,k),s
xperm(nox1,noy1,noz1)
       if
(sxperm(nox1,noy1,noz1).ne.syperm(no
x1,noy1,noz1)) then
       write(*,*) 'xperm is not
equal to yperm, please check input
or
     >logic in second upscale !!'
       stop
       endif
 40        enddo
          nozz1=noz1
          noz1=0
         enddo
         noyy1=noy1
```

```
        noy1=0
       enddo
        noy1=noyy1
        noz1=nozz1

        write(*,'(a,3i5)')
     >'# of (x,y,z) grids from
second upscale =', nox1, noy1, noz1

        if
((nox1.ne.nx1).or.(noy1.ne.ny1).or.(
noz1.ne.nz1))then
        write(*,*) 'Error !! Please
check program logic and input'
        stop
        endif

c  Calculate normalized permeability
of blocks within the ranges and
group (arrange)
c  the blocks included in the given
ranges.
      do i=1,NS   ! Divide the
blocks to 8 groups with same x,y-
perm
        ng(i)=0
       enddo

       Do i=1,nx1
         do j=1,ny1
           do k=1,nz1
       if ((k.le.2).or.(k.ge.9)) then
c          if
((sxperm(i,j,k).le.1.0e-
12).and.(sxperm(i,j,k).gt.
c     >5.0e-13))
c     >then
c           ng(1)=ng(1)+1 ! Counting
the block
c           nni(i,j,k)=1 ! numbering
block for ni(nk) below
c           gsxperm(i,j,k)=7.5e-13
c           gsyperm(i,j,k)=7.5e-13
c           gszperm(i,j,k)=4.99e-14

          if ((sxperm(i,j,k).le.5.0e-
13).and.(sxperm(i,j,k).gt.
     >1.0e-13))then
           ng(1)=ng(1)+1
           nni(i,j,k)=1
           gsxperm(i,j,k)=2.5e-13
           gsyperm(i,j,k)=2.5e-13
           gszperm(i,j,k)=1.84e-14

          elseif
((sxperm(i,j,k).le.1.0e-
13).and.(sxperm(i,j,k).gt.
     >5.0e-14))
     >then
           ng(2)=ng(2)+1 ! Counting
the block
           nni(i,j,k)=2 ! numbering
block for ni(nk) below
           gsxperm(i,j,k)=7.5e-14
           gsyperm(i,j,k)=7.5e-14
           gszperm(i,j,k)=4.99e-15

          elseif
((sxperm(i,j,k).le.5.0e-
14).and.(sxperm(i,j,k).gt.
     >1.0e-14))then
           ng(3)=ng(3)+1
           nni(i,j,k)=3
           gsxperm(i,j,k)=2.5e-14
           gsyperm(i,j,k)=2.5e-14
           gszperm(i,j,k)=1.84e-15

          elseif
((sxperm(i,j,k).le.1.0e-
14).and.(sxperm(i,j,k).gt.
     >5.0e-15))then
           ng(4)=ng(4)+1
           nni(i,j,k)=4
           gsxperm(i,j,k)=7.5e-15
           gsyperm(i,j,k)=7.5e-15
           gszperm(i,j,k)=4.99e-16

          elseif
((sxperm(i,j,k).le.5.0e-
15).and.(sxperm(i,j,k).gt.
     >1.0e-15))then
           ng(5)=ng(5)+1
           nni(i,j,k)=5
           gsxperm(i,j,k)=2.5e-15
           gsyperm(i,j,k)=2.5e-15
           gszperm(i,j,k)=1.84e-16

          elseif
((sxperm(i,j,k).le.1.0e-
15).and.(sxperm(i,j,k).gt.
     >5.0e-16))then
           ng(6)=ng(6)+1
           nni(i,j,k)=6
           gsxperm(i,j,k)=7.5e-16
           gsyperm(i,j,k)=7.5e-16
           gszperm(i,j,k)=4.99e-17
```

```
         elseif
((sxperm(i,j,k).le.5.0e-
16).and.(sxperm(i,j,k).gt.
     >1.0e-16))then
              ng(7)=ng(7)+1
              nni(i,j,k)=7
              gsxperm(i,j,k)=2.5e-16
              gsyperm(i,j,k)=2.5e-16
              gszperm(i,j,k)=1.84e-17

         elseif
((sxperm(i,j,k).le.1.0e-
16).and.(sxperm(i,j,k).gt.
     >5.0e-17))then
              ng(8)=ng(8)+1
              nni(i,j,k)=8
              gsxperm(i,j,k)=7.5e-17
              gsyperm(i,j,k)=7.5e-17
              gszperm(i,j,k)=4.99e-18

c         elseif
((sxperm(i,j,k).le.5.0e-
16).and.(sxperm(i,j,k).gt.
c     >1.0e-16))then
c              ng(10)=ng(10)+1
c              nni(i,j,k)=10
c              gsxperm(i,j,k)=2.5e-16
c              gsyperm(i,j,k)=2.5e-16

c         elseif
((sxperm(i,j,k).le.1.0e-
16).and.(sxperm(i,j,k).gt.
c     >5.0e-17))then
c              ng(11)=ng(11)+1
c              nni(i,j,k)=11
c              gsxperm(i,j,k)=7.5e-17
c              gsyperm(i,j,k)=7.5e-17

c         elseif
((sxperm(i,j,k).le.5.0e-
17).and.(sxperm(i,j,k).gt.
c     >1.0e-17))then
c              ng(12)=ng(12)+1
c              nni(i,j,k)=12
c              gsxperm(i,j,k)=2.5e-17
c              gsyperm(i,j,k)=2.5e-17

c         elseif
((sxperm(i,j,k).le.1.0e-
17).and.(sxperm(i,j,k).gt.
c     >5.0e-18))then
c              ng(13)=ng(13)+1
c              nni(i,j,k)=13
c              gsxperm(i,j,k)=7.5e-18
c              gsyperm(i,j,k)=7.5e-18

c         elseif
((sxperm(i,j,k).le.5.0e-
18).and.(sxperm(i,j,k).gt.
c     >1.0e-18))then
c              ng(14)=ng(14)+1
c              nni(i,j,k)=14
c              gsxperm(i,j,k)=2.5e-18
c              gsyperm(i,j,k)=2.5e-18
            endif


elseif((k.gt.2).and.(k.lt.9))then
              gsxperm(i,j,k)=1.0e-20
              gsyperm(i,j,k)=1.0e-20
              gszperm(i,j,k)=1.0e-20
      endif
          enddo
        enddo
      enddo

c        do i=1,ns
c          do j=1,ng(i)
c         ncc(i)=1
c         zzm(i,ncc(i))=zm(i,1)
c         zzm(i)=0.
c          enddo
c        enddo
c         do i=1,nx1
c            do j=1,ny1
c              do k=1,nz1
c                do ii=1,NS
c              zzm(ncc(ii))=zm(ii,1)
c                    do jj=1,ng(ii)
c                       do kk=1,ng(ii)
c
if(zzm(ii,ncc(ii)).eq.zm(ii,kk))then
c           ncc(ii)=ncc(ii)+1
c           zzm(kk)=zm(ii,jj)
c         write(*,*) zzm

c        endif
c       enddo

      nsum=0
      do i=1,NS  ! Calculate number
of divided group
        nsum=nsum+ng(i)
      enddo
       write(*,'(A)')'  Group #
# of blocks'
        do i=1,ns
           write(*,'(i6,5x,i6)') i,
ng(i)
        enddo
```

```
       write(*,'(A,i6)')'  Total #
of blocks =  ',nsum

C SPECIFY X,Y-DIRECTIONAL DISTANCE
and numbering OF BLOCKS FROM ORIGIN
c This is not based on block
centers, but this is based on line.
c A line on origin is 0, the first
line (x-axis) is 0m-400m, 2nd line
is 400m-800m....
c As the same way, 1st line (y-axis)
is 0-400m, 2nd line is 400-800m...
      XDD(1)=0. ! X-POINT OF FIRST
BLOCK
      YDD(1)=0.
      DO I=2,NX1+1
        XDD(I)=XDD(I-1)+DX(I-1)
      ENDDO
      DO J=2,NY1+1
        YDD(J)=YDD(J-1)+DY(J-1)
      ENDDO

c Assign renormalized each block for
each grid
      NK=0
      NCH=0
      do i=1,nx2
       do j=1,ny2
        do k=1,nz2
           nk=nk+1
           PORO(NK)=0.2 ! SANDSTONE
PERMEABILITY
           TEMP(NK)=50.0 ! INITIAL
TEMPERATURE
           SALT(NK)=0.05 ! INITIAL
SALT MASS FRACTION
           CO2(NK)=0.0   ! INITIAL
CO2 MASS FRACTION
         IF ((k.gt.2).and.(k.lt.9))
THEN ! Define shale layer
           ni(nk)=9  ! impermeable
layer
           PORO(NK)=0.02 ! SHALE
PERMEABILITY
           GXP(nk)=1.0e-20
           GYP(nk)=1.0e-20
           GZP(nk)=1.0e-20
           AGXP(nk)=1.0e-20
           AGYP(nk)=1.0e-20
           AGZP(nk)=1.0e-20
            if
((i.eq.17).and.(j.eq.19)) then ! IG
of leakage pathway
             ni(nk)=10 ! rock
property number of expected leakage
path
c              poro(nk)=0.2 !
porosity of leakage pathway
           elseif
((i.eq.17).and.(j.eq.21)) then ! IG
of leakagepathway
             ni(nk)=11   ! rock
property number of leakage pathway
           elseif
((i.eq.20).and.(j.eq.19)) then ! IG
of leakagepathway
             ni(nk)=12   ! rock
property number of leakage pathway
           elseif
((i.eq.20).and.(j.eq.21)) then ! IG
of leakagepathway
             ni(nk)=13   ! rock
property number of leakage pathway
           endif
         elseif (K.le.2) then !
DEFINE NI IN OVERLYING FORM
          DO IG=1,NX1
           DO JG=1,NY1
            IF
(((XC2(nk).GE.XDD(IG)).AND.(XC2(nk).
LE.XDD(IG+1))).AND.

>((YC2(nk).GE.YDD(JG)).AND.(YC2(nk).
LE.YDD(JG+1)))) THEN
           GXP(nk)=sxperm(IG,JG,K)
           GYP(nk)=syperm(IG,JG,K)
           GZP(NK)=SZPERM(IG,JG,K)
           ni(nk)=nni(IG,JG,K)

AGXP(NK)=gsxperm(IG,JG,K)

AGYP(NK)=GSYPERM(IG,JG,K)

AGZP(NK)=GSZPERM(IG,JG,K)
           ENDIF
          ENDDO
         ENDDO
        elseif (k.ge.9) then !
DEFINE NI IN STORAGE FORM
          DO IG=1,NX1
           DO JG=1,NY1
            IF
(((XC2(nk).GE.XDD(IG)).AND.(XC2(nk).
LE.XDD(IG+1))).AND.

>((YC2(nk).GE.YDD(JG)).AND.(YC2(nk).
LE.YDD(JG+1)))) THEN
           GXP(nk)=sxperm(IG,JG,K)
```

```
            GYP(nk)=syperm(IG,JG,K)
            GZP(NK)=SZPERM(IG,JG,K)
            ni(nk)=nni(IG,JG,K)

AGXP(NK)=gsxperm(IG,JG,K)

AGYP(NK)=GSYPERM(IG,JG,K)

AGZP(NK)=GSZPERM(IG,JG,K)
             ENDIF
           ENDDO
          ENDDO
         ENDIF

         IF (J.EQ.1 .OR. J.EQ.NY2)
THEN ! DEFINE CONSTANT HEAD BOUNDARY
           VOL(NK)=0.2E66
           NCH=NCH+1
         ENDIF
        ENDDO
       ENDDO
      ENDDO

C SPECIFY ELEVATION OF CELLS (z-axis
is based on block centered scheme to
calculate
c initial condition
      ZDD(1)=-DZ(1)/2.+ELV
C      WRITE(*,*) ZDD(1)
      DO K=2,NZ2
       ZDD(K)=ZDD(K-1)-(DZ(K-
1)+DZ(K))/2.
C       WRITE(*,*) ZDD(K)
      ENDDO
C CALCULATE INITIAL HYDRAULIC HEAD
VALUES AT ALL CELLS
      DO I=1,NX2
        DO J=1,NY2
          DO K=1,NZ2

             HEAD(I,J,K)=10.0   !
Initial head of Brine

          ENDDO
        ENDDO
      ENDDO

C  CALCULATE PRESSURE AT ALL CELLS
FOR INITIAL CONDITION
      NK1=0
C      NCH=0
c      NOFLOW=0
      DO I=1,NX2
       DO J=1,NY2
        DO K=1,NZ2
          NK1=NK1+1

PR(NK1)=1025.*9.80665*(HEAD(I,J,K)-
ZDD(K))
c
TEMP(NK)=UTEMP+25.0/1000.0*(ZDD(1)-
ZDD(K))
        ENDDO
       ENDDO
      ENDDO

C  READ INPUT PROPERTY
      DO I=1,NLINP
       READ(7,'(A)') POINP(I) !
POINP : PROPERTY OF INPUT
      ENDDO

C  WRITE INPUT PROPERTY IN OUTPUT
FILE (input_fw_  .out)
      DO I=1,NLINP-1
       WRITE(6,'(A)') POINP(I)
      ENDDO

C  WRITE INITIAL CONDITION IN OUTPUT
FILE (input_fw_  .out)
C      WRITE(6,*)
      WRITE(6,'(A)') 'INCON'
      DO I=1,NC
       WRITE(6,'(A,10X,E15.8)')
CORD(I),PORO(I)
       WRITE(6,'(4E20.13)')
PR(I),SALT(I),CO2(I),TEMP(I)
C       WRITE(6,'(3E20.13)')
PRESS(I),VAP(I),TEMP(I)
      ENDDO

C  WRITE ELEME & CONNE BLOCK
       WRITE(6,*)
       write(6,'(a)') title
        do I=1,NC

WRITE(6,'(a,13x,i2,2e10.4,10x,3f10.3
)') cord(i),ni(i),vol(i),
     >area(i),xc2(i),yc2(i),zc2(i)
        ENDDO
        DO J=NC+1,NLCON-1
         WRITE(6,'(A)') CONN(J)
        ENDDO

C  WRITE INPUT PROPERTY
       WRITE(6,*)
       WRITE(6,'(A)') POINP(NLINP)
       WRITE(6,*)
```

```
c  write permability values
depanding on coordinate for techplot
form
       write(9,'(a)') 'Variables = X
Y Z X-PERM Y-PERM Z-PERM'
       write(9,'(a,a,i7)') 'Zone
F=POINT ','I=',nc
       write(10,'(a)') 'Variables =
X Y Z X-PERM Y-PERM Z-PERM'
       write(10,'(a,a,i7)') 'Zone
F=POINT ','I=',nc
       do i=1,nc !PRINTOUT REAL
UPSCALED PERM ON EACH GRIDS
       write(9,'(3f10.3,x,3E13.4)')
xc2(i),yc2(i),zc2(i),GXP(i),
     >GYP(i),GZP(i)
       enddo
       do i=1,nc !PRINTOUT MEDIUM
UPSCALED PERM ON EACH GRIDS
       write(10,'(3f10.3,x,3E13.4)')
xc2(i),yc2(i),zc2(i),AGXP(i),
     >AGYP(i),AGZP(i)
       enddo

      write(*,*) ' The number of
constant head boundary nodes= ', nch
      write(*,*)

      IF (NK.NE.NC) THEN
      WRITE(*,*)' PLEASE CHECK THE
NUMBER OF CELLS'
      STOP
      ENDIF

      STOP
      END
```

APPENDIX K

TOUGH2 INPUT GENERATOR

```
program tough2_input_generator

c  Program to automatically make
input of tough2 with eleme,
c  conne and incon blocks.
c  This program needs two input
files. First, a mesh file generated
from tough2
c  using mesh.dat. Second, another
tough2 input file with from rocks,
multi,
c  selec, solvr, start, param, and
times blocks.
c  Gener and foft blocks should be
filled out manually in output file
from this
c  program.

c  This simulation is to generate
domain (10100*10100*220m) with
grids(25*35*6).
c  For Contour map of risidual

      IMPLICIT REAL*8(A-H,O-Z)

      character*40 title
      character*5 cord(500000)
      CHARACTER*80 CONN(500000),
POINP(500000)
      dimension
ni(500000),area(500000),vol(500000)
      dimension
xc(500000),yc(500000),zc(500000)
      real mop(500000) ! modifier of
permeability
      DIMENSION
XDD(200000),YDD(200000),ZDD(200000)
      DIMENSION
PORO(200000),TEMP(200000),SALT(20000
0),CO2(200000)
      DIMENSION
HEAD(200,200,100),PR(200000)
C      DIMENSION
DTEMP(7000),DSALT(7000),DCO2(7000),D
PORO(7000)
C      DIMENSION
PORO(5000),PRESS(5000),TEMP(5000),VA
P(5000)

      DIMENSION DX(21),DY(31),DZ(11)
      DATA DX
/3949.85,0.3,49.85,100.,849.85,0.3,4
9.85,100.,

>49.85,0.3,99.7,0.3,99.7,0.3,199.7,0
.3,

>449.85,100.,449.85,0.3,3549.85/
      DATA DY
/100.,3900.,100.,449.85,0.3,449.85,1
00.,449.85,

>0.3,199.7,0.3,199.7,0.3,99.7,0.3,99
.7,

>0.3,199.7,0.3,199.7,0.3,499.7,0.3,4
99.7,

>0.3,499.7,0.3,999.7,0.3,949.85,100.
/
      DATA DZ
/20.0,20.0,20.0,20.0,20.0,20.0,20.0,
20.0,20.0,20.0,20.0/

      NX=21  ! NUMBER OF X-
COORDINATE
      NY=31  ! NUMBER OF Y-
COORDINATE
      NZ=11  ! NUMBER OF Z-
COORDINATE
      NLCON=27424  ! TOTAL NUMBER OF
LINE IN MESH FILE
      NLINP=62  ! TOTAL NUMBER OF
LINE IN INPUT_PROPERTY FILE
      ELV=-1000. ! ELEVATION OF TOP
FROM ORIGIN
      UTEMP=50.0 ! TEMPERATURE AT
TOP

      NC=NX*NY*NZ ! NUMBER OF CELLS

      open(5,
file='mesh_inv_8_5_leaksize0.3m.dat'
, status='unknown')
      OPEN(7,
FILE='input_property_inv.dat',
STATUS='UNKNOWN')
C      OPEN(6,
FILE='SAVE_FORWARD.OUT',
STATUS='UNKNOWN')
      open(6,
file='input_inv_8_5_leaksize0.3m.out
', status='unknown')

C  READ MESH FILE AND SPECIFY ROCK &
INITIAL PROPERTY
      read(5,'(a)') title
      DO I=1,NLCON-1
```

```
       IF (I.LE.NC) THEN

read(5,'(a,14x,i1,2e10.4,10x,3f10.3)
') cord(i),ni(i),vol(i),
     >area(i),xc(i),yc(i),zc(i)
         ELSE
           READ(5,'(A)') CONN(I)
         ENDIF
       ENDDO

       nk=0
       nch=0
       DO I=1,NX
        DO J=1,NY
         DO K=1,NZ
           nk=nk+1
            PORO(NK)=0.2 ! SANDSTONE
PERMEABILITY
            TEMP(NK)=50.0 ! INITIAL
TEMPERATURE
            SALT(NK)=0.05 ! INITIAL
SALT MASS FRACTION
            CO2(NK)=0.0   ! INITIAL
CO2 MASS FRACTION
          IF (K.GE.3 .AND. K.LE.8)
THEN ! Define shale layer
            ni(nk)=3  ! impermeable
layer
            PORO(NK)=0.02 ! SHALE
PERMEABILITY
c             if
((i.eq.54).and.(j.eq.62)) then !
define leakage pathway
c               ni(nk)=4   ! rock
property number of leakage pathway
c               poro(nk)=0.2 !
porosity of leakage pathway
c             endif
          elseif (K.le.2) then
            ni(nk)=1
          elseif (k.ge.9) then
            ni(nk)=2
          ENDIF
          IF (J.EQ.1 .OR. J.EQ.NY)
THEN ! DEFINE CONSTANT HEAD BOUNDARY
            VOL(NK)=0.2E66
            NCH=NCH+1
          ENDIF
         ENDDO
        ENDDO
       ENDDO

C SPECIFY X,Y-DIRECTIONAL DISTANCE
OF CELLS FROM ORIGIN
      XDD(1)=DX(1)/2.
      YDD(1)=DY(1)/2.
      DO I=2,NX
        XDD(I)=XDD(I-1)+(DX(I-
1)+DX(I))/2.
      ENDDO
      DO J=2,NY
        YDD(J)=YDD(J-1)+(DY(J-
1)+DY(J))/2.
      ENDDO
C SPECIFY ELEVATION OF CELLS
      ZDD(1)=-DZ(1)/2.+ELV
      WRITE(*,*) ZDD(1)
      DO K=2,NZ
       ZDD(K)=ZDD(K-1)-(DZ(K-
1)+DZ(K))/2.
       WRITE(*,*) ZDD(K)
      ENDDO
C CALCULATE INITIAL HYDRAULIC HEAD
VALUES AT ALL CELLS
      DO I=1,NX
        DO J=1,NY
          DO K=1,NZ

             HEAD(I,J,K)=10.0   !
Initial head of Brine

          ENDDO
        ENDDO
      ENDDO

C  CALCULATE PRESSURE AT ALL CELLS
FOR INITIAL CONDITION
      NK=0
C      NCH=0
      NOFLOW=0
      DO I=1,NX
       DO J=1,NY
        DO K=1,NZ
          NK=NK+1

PR(NK)=1025.*9.80665*(HEAD(I,J,K)-
ZDD(K))
c
TEMP(NK)=UTEMP+25.0/1000.0*(ZDD(1)-
ZDD(K))
        ENDDO
       ENDDO
      ENDDO

C  READ INPUT PROPERTY
      DO I=1,NLINP
       READ(7,'(A)') POINP(I) !
POINP : PROPERTY OF INPUT
      ENDDO
```

```
C  WRITE INPUT PROPERTY IN OUTPUT
FILE (input_fw_  .out)
      DO I=1,NLINP-1
       WRITE(6,'(A)') POINP(I)
      ENDDO

C  WRITE INITIAL CONDITION IN OUTPUT
FILE (input_fw_  .out)
C      WRITE(6,*)
      WRITE(6,'(A)') 'INCON'
      DO I=1,NC
       WRITE(6,'(A,10X,E15.8)')
CORD(I),PORO(I)
       WRITE(6,'(4E20.13)')
PR(I),SALT(I),CO2(I),TEMP(I)
C       WRITE(6,'(3E20.13)')
PRESS(I),VAP(I),TEMP(I)
      ENDDO

C  WRITE ELEME & CONNE BLOCK
       WRITE(6,*)
       write(6,'(a)') title
        do I=1,NC

WRITE(6,'(a,14x,i1,2e10.4,10x,3f10.3
)') cord(i),ni(i),vol(i),
     >area(i),xc(i),yc(i),zc(i)
        ENDDO
        DO J=NC+1,NLCON-1
         WRITE(6,'(A)') CONN(J)
        ENDDO

C  WRITE INPUT PROPERTY
       WRITE(6,*)
       WRITE(6,'(A)') POINP(NLINP)
       WRITE(6,*)

      write(*,*) ' The number of
constant head boundary nodes= ', nch
      if (nk.ne.nc) then
      write(*,*)' Please check the
number of cells'
      stop
      endif

C      WRITE(*,*) ' THE NUMBER OF
CONSTANT HEAD BOUNDARY NODES= ', NCH
      IF (NK.NE.NC) THEN
      WRITE(*,*)' PLEASE CHECK THE
NUMBER OF CELLS'
      STOP
      ENDIF

      STOP
      END
```

APPENDIX L

PROGRAM FOR DIFFENCE PRESSURE AT EACH TIME STEP

```
PROGRAM dPressure

C THIS PROGRAM IS FOR 2-D SLICE
CUTTING Z-AXIS (XY PLANE) OF TECPLOT
FOR
C SENSITIVITY ANALYSIS OF PRESSURE
BETWEEN LEAKY AND NO LEAKY
CONDITIONS.
C THIS SIMULATION IS TO PRINT OUT
TIME-DEPENDENT DIFFERENCE PRESSURE
IN OVERLYING
C FORMATION BETWEEN LEKY AND NO
LEAKY CONDITIONS TO TECPLOT FORMAT.
C THE PROGRAM READS TIME DEPENDENT
PRESSURE FILES FROM TWO SIMULATION
CONDITIONS,
C CALCULATES TIME DEPENDENT
DIFFERENCE PRESSURES OF NODES IN
OVERLYING FORMATION
C AND THEN GENERATES THE FILES
SEPARATED TO TIME SERIES FOR
TECPLOT.
C THERE ARE TWO KIND OF INPUT DATA.
ONE IS PRESSURE FILES WITH LEAK
CONDITION.
C THE OTHER IS PRESSURE FILES
WITHOUT LEAK CONDITION. THE FILES
HAVE TECPLOT INPUT
C FORM FOR TIME SERIES. THE OUTPUT
FILES PROVIDE OUTPUT WITH 2-D
TECPLOT FORM
C SEPERATED DEPENDING ON TIME.

      IMPLICIT REAL*8(A-H,O-Z)

      CHARACTER*80 TITLE11,TITLE21
      CHARACTER*80 TITLE12,TITLE22
      CHARACTER*80
FNAME1(500),FNAME2(500),FNAME3(500)
      CHARACTER*20 NFILE(500)
      DIMENSION
XC1(200),YC1(200),ZC1(200)
      DIMENSION
XC2(200),YC2(200),ZC2(200)
      DIMENSION
P1(200,200,20),T1(200000),SG1(200000
),SS1(200000)
      DIMENSION
XNACL1(200000),YH2OG1(200000),XCO2A1
(200000)
      DIMENSION
PCAP1(200000),KRED1(200000),DG1(2000
00),DL1(200000)
      DIMENSION
P2(200,200,20),T2(200000),SG2(200000
),SS2(200000)
      DIMENSION
XNACL2(200000),YH2OG2(200000),XCO2A2
(200000)
      DIMENSION
PCAP2(200000),KRED2(200000),DG2(2000
00),DL2(200000)
      DIMENSION DP(200000)
C      REAL KX, KY,
KZ,KZBLEAK,LSTIME
      INTEGER XCN

      NX=103 ! THE NUMBER OF X-
DIRECTIONAL CELLS
      NY=103 ! THE NUMBER OF Y-
DIRECTIONAL CELLS
      NZ=11 ! THE NUMBER OF Z-
DIRECTIONAL CELLS
      NT=38 ! THE NUMBER OF TIME
STEP (number of each input file)
      XCN=2 ! THE Z-COORDINATE
NUMBER TO PRINT OUT

      NTN=NX*NY*NZ
      NOT=NX*NY

C   CONVERT REAL NUMBER TO CHARACTER
      DO I=1,NT
       IF (I.LE.9) THEN
        WRITE(7,'(A1,I1)') '0',I
       ELSEIF (I.LE.99) THEN
        WRITE(7,'(I2)') I
C       WRITE(7,'(A1,I2)') '0',I
       ELSE
        WRITE(*,*) 'PLEASE CHECK
NUMBER OF INPUT FILES AND REVISE IT'
        STOP
C       WRITE(7,'(I3)') I
       ENDIF
      ENDDO
      REWIND(7)
      DO I=1,NT
       READ(7,'(A)') NFILE(I)
      ENDDO
      CLOSE(7)

C  REVISE INPUT FILE NAMES
      DO 100 JJ=1,NT

FNAME1='input_fw_8_leaksize1m.out.'/
/NFILE(JJ)
       FNAME2='input_fw_8_leakperm-
20_120seal.out.'//NFILE(JJ)
```

```
FNAME3='dp_fw_9_inj6.34.out.'//NFILE
(JJ)
C     // ADDES A CHARACTER VALUE OF
NFILE(J) AT END OF FILE NAME
      OPEN(5, FILE=FNAME1,
STATUS='UNKNOWN') ! READ INPUT FILE
      OPEN(6, FILE=FNAME2,
STATUS='UNKNOWN') ! READ INPUT FILE
      OPEN(8, FILE=FNAME3,
STATUS='UNKNOWN') ! WRITE OUTPUT
FILE

       READ (5,'(A)') TITLE11 ! 1st
line in the 1st input file
       READ (5,'(A)') TITLE12 ! 2nd
line in the 1st input file
       READ (6,'(A)') TITLE21 ! 1st
line in the 2nd input file
       READ (6,'(A)') TITLE22 ! 2nd
line in the 2nd input file

      NC=0
      DO I=1,NX
       DO J=1,NY
        DO K=1,NZ
        NC=NC+1
       READ(5,*)
XC1(I),YC1(J),ZC1(K),P1(I,J,K),T1(NC
),SG1(NC),

>SS1(NC),XNACL1(NC),YH2OG1(NC),XCO2A
1(NC),PCAP1(NC),KRED1(NC),
     >DG1(NC),DL1(NC) ! Read 1st
input file
       READ(6,*)
XC2(I),YC2(J),ZC2(K),P2(I,J,K),T2(NC
),SG2(NC),

>SS2(NC),XNACL2(NC),YH2OG2(NC),XCO2A
2(NC),PCAP2(NC),KRED2(NC),
     >DG2(NC),DL2(NC) ! Read 2nd
input file
        ENDDO
       ENDDO
      ENDDO

      IF (NC.NE.NTN) THEN
       WRITE(*,*) ' CHECK TOTAL
NUMBER OF CELLS !! '
      STOP
      ENDIF

      WRITE(8,'(A)') 'Variables = X
Y dP '
      WRITE(8,'(A,A,I6,X,A)')
TITLE12(1:13),'I=',NOT,TITLE12(23:43
)
c    (1:13) is to print out
character from 1 digit to 14 digit

      NC=0
      NP=0
      DO I=1,NX
       DO J=1,NY
        DO K=1,NZ
        NC=NC+1
        DP(NC)=ABS(P1(I,J,K)-
P2(I,J,K)) ! Calculate pressure
difference
          IF(K.EQ.XCN) THEN
           NP=NP+1
         WRITE(8,'(2E14.5,x,E17.7)')
XC1(I),YC1(J),DP(NC) ! PRINTOUT
          ENDIF
        ENDDO
       ENDDO
      ENDDO
      CLOSE(5)
      CLOSE(6)
      CLOSE(8)

      IF (NP.NE.NOT) THEN
       WRITE(*,*) 'CHECK NUMBER OF
CELLS TO PRINT OUT !!'
       STOP
      ENDIF

      WRITE(*,'(A,I3)')' TERMINATED
FILE IS ',JJ
       REWIND(5)
       REWIND(6)
       REWIND(8)

 100  CONTINUE
      WRITE(*,'(A,I10)')'THE NUMBER
OF NODE TO PRINT OUT IN EACH OUTPUT
     >FILE = ',NP
      WRITE(*,*)

      STOP
      END
Program for random noise generation
in measurements

      Program Noise
c      implicit real*8(a-h,o-z)
      dimension
time(1000),press(30,1000)
```

```
      dimension
rdnum1(30,1000),rdnum2(30,1000)
      real nran(30,1000),
noispres(30,1000),noise(30,1000)
      dimension summ(30), ssum(30),
std(30)
      real mean(30)

      open(4, file='mea85.dat',
status='unknown')
      open(3, file='noise.out',
status='unknown')
      open(5,
file='noise_mea85.out',
status='unknown')
      open(6, file='ran_number.out',
status='unknown')
C     Starting part for input data
      nm=9  ! Number of measurement
points
      nmd=776 ! Number of
measurement time dependent
      idum=1000 ! coefficient to
generate random number
C     end of input data

      do j=1,nmd
      read(4,*) time(j),(press(i,j),
i=1,nm)
      enddo

C     Random number
      do i=1,nm
        do j=1,nmd
       rdnum1(i,j)=ran(idum)
       rdnum2(i,j)=ran(idum)
        enddo
      enddo

c This random generator makes 0<
random number <0.999
c so negative sign is given randomly
from rdnum2
c and for 1% noise 0.01 is
multiplied.
c      New random number
      do i=1,nm
        do j=1,nmd
        if
((rdnum2(i,j).gt.0.2).and.(rdnum2(i,
j).lt.0.7))then
      sign=-1.

nran(i,j)=rdnum1(i,j)*sign*0.001
        else
      nran(i,j)=rdnum1(i,j)*0.001
        endif
        enddo
       enddo

      do j=1,nmd
       write(6,'(10f10.5)')
(nran(i,j), i=1,nm)
      enddo

c  Generate new pressure with noise


      do i=1,nm
        summ(i)=0.
        do j=1,nmd

noise(i,j)=press(i,j)*nran(i,j) !
calculate pure noise
        summ(i)=summ(i)+noise(i,j)

noispres(i,j)=press(i,j)+noise(i,j)
! calcu. pressure with noise
        enddo
        mean(i)=summ(i)/float(nmd) !
mean value of noise for each MW
      enddo
      do i=1,nm
        ssum(i)=0.
        do j=1,nmd
        ssum(i)=ssum(i)+(noise(i,j)-
mean(i))**2.
        enddo

std(i)=sqrt(ssum(i)/float(nmd))
      enddo

      do j=1,nmd
        write(3,'(e13.6,10e17.8)')
time(j), (noise(i,j),i=1,nm)
        write(5,'(e13.6,10e17.8)')
time(j), (noispres(i,j), i=1,nm)
      enddo
        write(3,*)
        write(3,'(a,10e17.8)')' Mean
', (mean(i),i=1,nm)

write(3,'(a,10e17.8)')'Std.deviation
', (std(i),i=1,nm)
      close(3)
      close(4)
      close(5)
      stop
      end
```

```
C
####################################
##############################
      Function ran(idum)
c      implicit real*8(a-h,o-z)
      PARAMETER (IA=16807,
IM=18000000, AM=1./IM, IQ=127773,
     >           IR=2836,
MASK=123459876)
      idum=ieor(idum,MASK)
      k=idum/IQ
      idum=IA*(idum-k*IQ)-IR*k
      if (idum.lt.0) idum=idum+IM
      ran=abs(AM*idum)
      idum=ieor(idum,MASK)

      return
      end
C
####################################
##############################
```

APPENDIX M

FORWARD INPUT GENERATOR OF DEVELOPED SINGLE-PHASE MODE

```
PROGRAM FORWARD_INPUT_GENERATOR

C      DIMENSION
X(200),Y(200),Z(200)
      DIMENSION
XDD(5000),YDD(5000),ZDD(5000)
      REAL
KX(105,105,30),KY(105,105,30),KZ(105
,105,30),HEAD(105,105,30)
      REAL KZBLEAK,LSTIME
      OPEN(5, FILE='fw_case2_4.out',
STATUS='UNKNOWN')
      NX=103 ! THE NUMBER OF X-
DIRECTIONAL CELLS
      NY=101 ! THE NUMBER OF Y-
DIRECTIONAL CELLS
      NZ=13 ! THE NUMBER OF Z-
DIRECTIONAL CELLS
      DX=100. ! INTERVAL OF X-
DIRECTIONAL CELLS
      DY=100.
      DZ=20. ! INTERVAL OF Z-
DIRECTIONAL CELLS
      ELV=-1000.

      SS=1.0e-6
C      NBC=0
      KZBLEAK=0.
      ALEAK=0.
      LSTIME=0.

c      BH1=20.0 ! HYDRAULIC HEAD AT
UP AQUIFER OF FACE 2
c      BH2=25.0 ! HYDRAULIC HEAD AT
INJ AQUIFER OF FACE 2
c      BH3=15.0 ! HYDRAULIC HEAD AT
UP AQUIFER OF FACE 4
c      BH4=20.0 ! HYDRAULIC HEAD AT
INJ AQUIFER OF FACE 4
      BH1=10.0 ! HYDRAULIC HEAD AT
UP AQUIFER OF FACE 1
      BH2=10.0 ! HYDRAULIC HEAD AT
INJ AQUIFER OF FACE 5
      ST=0.0
      ET=100000000.0

C SPECIFY X,Y-DIRECTIONAL DISTANCE
OF CELLS FROM ORIGIN
      XDD(1)=DX/2.
      YDD(1)=DY/2.
      DO I=2,NX
        XDD(I)=XDD(I-1)+DX
      ENDDO
      DO J=2,NY
        YDD(J)=YDD(J-1)+DY
      ENDDO
C SPECIFY ELEVATION OF CELLS
      ZDD(1)=-DZ/2.+ELV
      DO K=2,NZ
       ZDD(K)=ZDD(K-1)-DZ
      ENDDO


      NCH=0
      NCHB=0
      IY=1
      JY=NY
      DO I=1,NX
       DO J=1,NY
        DO K=1,NZ

           HEAD(I,J,K)=BH1-(BH1-
BH2)/(YDD(NY)-YDD(1))*
     >       (YDD(J)-YDD(1))
         NBC=0
        IF ((K.GE.1).AND.(K.LE.3))
THEN
            KX(i,j,k)=0.18285E-07
            KY(i,j,k)=0.18285E-07
            KZ(i,j,k)=0.18285E-07
            PORO=0.2
        elseif
((K.GE.4).AND.(K.LE.9)) then
            KX(i,j,k)=0.18285E-12
            KY(i,j,k)=0.18285E-12
            KZ(i,j,k)=0.18285E-12
            PORO=0.02
            NBC=2
        elseif
((K.GE.10).AND.(K.LE.13)) then
            KX(i,j,k)=0.18285E-05
            KY(i,j,k)=0.18285E-05
            KZ(i,j,k)=0.18285E-05
            PORO=0.2
        ENDIF


IF((I.EQ.1).OR.(I.EQ.NX).OR.(k.EQ.1)
.OR.(k.EQ.NZ))THEN
           NBC=2
           NCH=NCH+1

ELSEIF(j.EQ.1.or.j.eq.ny)THEN
           NBC=1
           NCHB=NCHB+1
c           IF
((K.GE.8).AND.(K.LE.12)) THEN
c             KX(i,j,k)=1.0e-13
c             KY(i,j,k)=1.0e-13
c             KZ(i,j,k)=1.0e-13
```

```
c            ENDIF
c          ELSE
c            NBC=0
          ENDIF



           WRITE(5,11)
I,J,K,DX,DY,DZ,head(i,j,k),KX(i,j,k)
,KY(i,j,k),

>KZ(i,j,k),PORO,SS,NBC,KZBLEAK,ALEAK
,LSTIME
         ENDDO
        ENDDO
       ENDDO
c       NCHB=NCH
        write(*,*) ' number of
constant boundary = ', nchb
        write(*,*) ' number of no
flow boundary = ', nch
C       WRITE(5,41) NCHB
c       WRITE(5,21) ((I,IY,K,
K=2,4),(I,IY,K, K=6,NZ-1),I=2,NX-1),
c      >((II,JY,KK,
KK=2,4),(II,JY,KK, KK=6,NZ-
1),II=2,NX-1)
c       WRITE(5,31) ST,((BH1,
I=2,4),(BH2, I=6,NZ-1),J=2,NX-1),
c      >((BH3, II=2,4),(BH4, II=6,NZ-
1),JJ=2,NX-1)
c       WRITE(5,31) ET,((BH1,
I=2,4),(BH2, I=6,NZ-1),J=2,NX-1),
c      >((BH3, II=2,4),(BH4, II=6,NZ-
1),JJ=2,NX-1)
c       WRITE(5,21) ((I,iy,K, K=2,nz-
1),I=2,2),
c      >((ii,jy,kk, kk=2,nz-
1),ii=2,2)
c       WRITE(5,31) ST, (BH1,i=2,nz-
1),(bh2,j=2,nz-1)
c       WRITE(5,31) ET, (BH1,i=2,nz-
1),(bh2,j=2,nz-1)
 11   FORMAT
(3I5,X,3F7.1,2X,f14.10,x,3E13.5,2X,F
5.2,2X,f8.6,2X,I2,
     >3F7.1)
 21   FORMAT (8X,12000(3I3,x))
 31   FORMAT (8X,F12.2,12000F8.2)
 41   FORMAT (I5)
      STOP
      END
```

APPENDIX N

HYDRAULIC CONDUCTIVITY GROUP GENERATOR FOR MEASUREMENT INPUT IN DEVELOPED SINGLE-PHASE MODEL

```
PROGRAM GROUPING

C  THIS IS A PROGRAM TO GRPOUP CELLS
WITH THE SAME EXPECTED HYDRAULIC
CONDUCTIVITY
C  FOR INVERSE ANALYSIS
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*120 RT, RTX
      DIMENSION RT(20),RTX(20)
      dimension head(50,50,20)
      REAL
KX(50,50,20),KY(50,50,20),KZ(50,50,2
0)
      REAL
KZBLEAK(50,50,20),LSTIME(50,50,20)
      REAL HC(50,50)
      DIMENSION
IX(50),IY(50),IZ(50),DX(50),DY(50),D
Z(50)
      DIMENSION
SS(50,50,20),NBC(50,50,20),ALEAK(50,
50,20)
      DIMENSION PORO(50,50,20)
      INTEGER
N(50),XG(5,10000),YG(5,10000),ZG(5,1
0000)

      OPEN(4, FILE='inv_case2.dat',
STATUS='UNKNOWN')
      OPEN(5,
FILE='group_case2.out',
STATUS='UNKNOWN')
      NOG=1 ! TOTAL NUMBER OF GROUP
      NX=14 ! THE NUMBER OF X-
DIRECTIONAL CELLS
      NY=14 ! THE NUMBER OF Y-
DIRECTIONAL CELLS
      NZ=13 ! THE NUMBER OF Z-
DIRECTIONAL CELLS
      NPRINT=0 ! IF 1 MEANS PRINTING
NEW FORWARD INPUT DATA
               ! IF 0 MEANS NO
PRINTING FORWARD INPUT DATA

      DO I=1,11
       READ(4,'(A)') RT(I)
      ENDDO

      DO I=1,NX
        DO J=1,NY
          DO K=1,NZ
           READ(4,*)
IX(I),IY(J),IZ(K),DX(I),DY(J),DZ(K),
head(i,j,k),
>KX(I,J,K),KY(I,J,K),KZ(I,J,K),PORO(
I,J,K),SS(I,J,K),NBC(I,J,K),

>KZBLEAK(I,J,K),ALEAK(I,J,K),LSTIME(
I,J,K)
          ENDDO
        ENDDO
      ENDDO

      DO I=1,7
       READ(4,'(A)') RTX(I)
      ENDDO
       CLOSE(4)
      NTG=0 ! TOTAL NUMBER OF NODES
FOR GROUPING (EXCEPT NBC=2)
      N(1)=0
c      N(2)=0
c      N(3)=0
c      N(4)=0
      DO I=1,NX
        DO J=1,NY
          DO K=1,NZ
          IF (NBC(I,J,K).NE.2)THEN
c          NTG=NTG+1

IF(K.GE.2.AND.K.LE.3)THEN
              NTG=NTG+1
              N(1)=N(1)+1
              XG(1,N(1))=I
              YG(1,N(1))=J
              ZG(1,N(1))=K
c            ELSEIF(J.LE.2)THEN
c              N(2)=N(2)+1
c              XG(2,N(2))=I
c              YG(2,N(2))=J
c              ZG(2,N(2))=K
c
ELSEIF(J.GE.3.AND.J.LE.5)THEN
c              N(3)=N(3)+1
c              XG(3,N(3))=I
c              YG(3,N(3))=J
c              ZG(3,N(3))=K
c            ELSEIF(J.GE.6)THEN
c              N(4)=N(4)+1
c              XG(4,N(4))=I
c              YG(4,N(4))=J
c              ZG(4,N(4))=K
            ENDIF
          ENDIF
          ENDDO
        ENDDO
      ENDDO
```

```
       WRITE(*,'(A,I5)') 'TOTAL
NUMBER OF ELEMENTS FOR GROUPING=',
NTG

       WRITE(5,'(I5)') NOG
      DO I=1,NOG

WRITE(5,'(I2,3X,I4,3X,100(3I3,2X))')
I,N(I),(XG(I,J),YG(I,J),
     >ZG(I,J), J=1,N(I))
       ENDDO
C      REWIND(5)
C  PRINT OUT INPUT DATA TO VERIFY
       IF (NPRINT.EQ.1) THEN

       KZBLEAK(3,3,3)=0.02482893
       ALEAK(3,3,3)=1.0
       NBC(3,3,3)=4
       LSTIME(3,3,3)=1000000.0
       KZBLEAK(3,3,5)=0.02482893
       ALEAK(3,3,5)=1.0
       NBC(3,3,5)=5
       LSTIME(3,3,5)=1000000.0
       KZBLEAK(3,5,3)=0.03851417
       ALEAK(3,5,3)=1.0
       NBC(3,5,3)=4
       LSTIME(3,5,3)=1000000.0
       KZBLEAK(3,5,5)=0.03851417
       ALEAK(3,5,5)=1.0
       NBC(3,5,5)=5
       LSTIME(3,5,5)=1000000.0
       KZBLEAK(5,3,3)=0.14017595
       ALEAK(5,3,3)=1.0
       NBC(5,3,3)=4
       LSTIME(5,3,3)=1000000.0
       KZBLEAK(5,3,5)=0.14017595
       ALEAK(5,3,5)=1.0
       NBC(5,3,5)=5
       LSTIME(5,3,5)=1000000.0
       KZBLEAK(5,5,3)=0.01564027
       ALEAK(5,5,3)=1.0
       NBC(5,5,3)=4
       LSTIME(5,5,3)=1000000.0
       KZBLEAK(5,5,5)=0.01564027
       ALEAK(5,5,5)=1.0
       NBC(5,5,5)=5
       LSTIME(5,5,5)=1000000.0

       HC(1,1)=0.00062353
       HC(1,2)=0.00066275
       HC(2,1)=0.00079216
       HC(2,2)=0.00066667
       HC(3,1)=0.00021177
       HC(3,2)=0.00083922
       HC(4,1)=0.00070589
       HC(4,2)=0.00043530

       DO I=1,NOG
         DO J=1,N(I)

KX(XG(I,J),YG(I,J),ZG(I,J))=HC(I,1)

KY(XG(I,J),YG(I,J),ZG(I,J))=HC(I,1)

KZ(XG(I,J),YG(I,J),ZG(I,J))=HC(I,2)
          ENDDO
        ENDDO

       OPEN(6,
FILE='FORWARD_INPUT.OUT',
STATUS='UNKNOWN')
      DO I=1,11
       WRITE(6,'(A)') RT(I)
      ENDDO

      DO I=1,NX
        DO J=1,NY
          DO K=1,NZ

WRITE(6,'(3I4,2X,3F7.2,3F12.9,2F8.5,
I3,X,F12.9,X,F6.2,F12.2)')
     >
IX(I),IY(J),IZ(K),DX(I),DY(J),DZ(K),
KX(I,J,K),KY(I,J,K),

>KZ(I,J,K),PORO(I,J,K),SS(I,J,K),NBC
(I,J,K),KZBLEAK(I,J,K),
     >ALEAK(I,J,K),LSTIME(I,J,K)
          ENDDO
         ENDDO
       ENDDO

      DO I=1,9
       WRITE(6,'(A)') RTX(I)
      ENDDO

      ENDIF

      STOP
      END
```