KYOTO UNIVERSITY

Development and applications of computational methods for saturated-unsaturated subsurface flow

by

Hyunuk AN

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Hydrology & Water Resources Engineering
Dept. of Urban & Environmental Engineering

Declaration of Authorship

I declare that this thesis titled 'Development and applications of computational methods for saturated-unsaturated subsurface flow' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Hyunuk An

KYOTO UNIVERSITY

Abstract

Graduate School of Engineering
Urban & Environmental Engineering

Doctor of Philosophy

by Hyunuk AN

The modeling of saturated–unsaturated flow through porous media is an important research topic in various branches of water resources engineering, agricultural engineering, chemical contaminant tracing, and rainfall runoff modeling. Although several analytical solutions of the governing equations of saturated–unsaturated flow through porous media have been reported, these solutions are generally obtained under simple initial and boundary conditions. Hence, numerical models are usually used to investigate saturated–unsaturated flow in porous media, where analytical solutions are not appropriate. Particularly, in hillslope hydrology, saturated–unsaturated subsurface flow models are used as tools to analyze the runoff processes obtained by observations or to better understand the hydrological processes of a hillslope. Further, saturated–unsaturated subsurface flow models have been widely used for simulating the nonequilibrium and preferential flow in many recent researches.

Over the last three decades, many numerical models including finite-difference models (FDMs) and finite-element models (FEMs) have been developed for simulating saturated–unsaturated flow. FDMs have certain advantages with respect to the ease of coding and understanding, owing to their simplicity of discretization as compared to FEMs. However, a disadvantage of FDMs has been often pointed out: FDMs do not accurately represent all geometrically complex flow domains with low resolution, especially in multidimensional simulations. This thesis focuses on the modeling of FDMs and tries to overcome their disadvantages. Computational cost is also an important issue in the modeling of saturated–unsaturated flow in porous media. In this thesis, the iterative alternating direction implicit (IADI) approach is considered in an attempt to improve the computational cost problem.

The main objectives of this thesis are as follows:

1. To reduce computational costs of the saturated–unsaturated flow model by using IADI method.

2. To extend FDM for simulating the saturated–unsaturated flow by using the coordinate transformation method.

3. To compare the iteration methods of the saturated–unsaturated flow model.

4. To present the applications of the saturated–unsaturated flow model.

Because the conventional IADI algorithm has problems of numerical stability and applicability to three-dimensional cases, a new IADI algorithm is derived in this thesis. The performance of the new IADI scheme is compared with the fully implicit scheme through several test problems. The coordinate-transformed FDM is applied to the curvilinear flow domain and its performance is compared with that of the conventional FDM or commercial FEM through test cases. To search for the most efficient and robust iteration method for the saturated–unsaturated flow model, the behavior of the Picard, Newton, and Newton-Krylov methods is investigated. By varying the boundary conditions, domain shape, heterogeneity, and anisotropy, several test simulations are conducted. We conclude that the numerical model proposed in this thesis shows good performance.

The main purpose of the research on the saturated–unsaturated flow model in this thesis is to use the model as a tool to understand the rainfall runoff processes on a hillslope. The saturated–unsaturated flow model has a better physical basis than conventional hydrological models such as the reservoir, tank, and integrated kinematic wave models and can provide more detailed information about rainfall runoff processes on a hillslope. As examples of the applications of the saturated–unsaturated flow model, two applications are presented in this thesis. The first case is a pipe-matrix subsurface flow combining the saturated–unsaturated flow model and the slot model in an iterative manner. The results obtained by the proposed model are compared with the experimental observation data. The second case is an assessment of the validity of the integrated kinematic wave model for a hillslope. Mainly, the magnitude of the infiltration effect is investigated using a two-dimensional (2D) physical model, which combines the saturated–unsaturated flow model for subsurface flow and the kinematic wave model for overland flow. By controlling several conditions such as soil depth, rainfall intensity, slope angle, and initial condition, simulations were conducted and the performance of the integrated kinematic wave model was assessed.

Acknowledgements

Acknowledgements was the most difficult part for me in this thesis due to my poor English writing. It was not easy to find the right word to describe my mind but I tried to express gratitude to people have supported me through this page.

First, I would like to express profound gratitude to my supervisor, Prof. Michiharu Shiiba. He is a great roll model as a researcher to me. His pure curiosity not only about research area but also about every unusual things was always inspiring me with respect to the researcher's attitude towards the problems. His insightful and appropriate advises especially on the mathematics were very helpful when I overcome the problems, which often seem not to be solved to me. Further, his confidence encouraged me to proceed to PhD course in Kyoto University. I think a lot of students have applied to our lab because his smile looks lovely; I was also one of them.

I came to Japan as a government scholarship foreigner student in undergraduate course of Kyoto University. But before joining *Shiiba lab*, Hydrology and Water Resources Research Laboratory in Kyoto University, I did not notice that a study or research could be pleasant. I remember the moment when my first assignment was given after joining the lab. That time was filled with happiness and enthusiasm. Associated Prof. Yutaka Ichikawa always gave me a motivation and taught me what I have to do to proceed to next step. I soon came to be interested in my research and could enjoy the time at the lab. For first several months, I disturbed him almost every day bringing questions, which might be troublesome to him but he always kindly tried to answer my questions. A lot of parts in this thesis started from his ideas. Without him, there would not be this dissertation. I am sincerely grateful to you, Prof. Ichikawa.

Associated Prof. Yasuto Tachikawa was a perfect roll model in another aspect with Prof. Shiiba. I am very impressed by his managing skills and activity in teamwork. He always arised me and the other lab members for our work, which encouraged us to do our best and gave us a motivation to proceed to the next step. Moreover, I also touched by his presentation ability at the seminar and the conference. I have learned many things from his presentations and his advises on extensive area were very helpful to proceed the research. Further, he took care of private difficulties from financial aspect to troubles of living in Japan. Thanks to his effort, I could concentrate on my research.

I also would like to give my gratitude to Lecturer Sunmin Kim, Mr. Sungjin Noh and Assistant Prof. Kazuaki Yorozu, They broadened my horizons and change my thoughts about the research. They showed me fresh point of view and approaches differ from those I learned at the lab. They were also good advisors and friends. Especially Dr.

Kim has been a person who can go to drink with me and talk private stories. I wish he would meet perfect woman for him and could get married soon.

I have been always thinking that it was lucky that I can participate in *Shiiba lab.* In the period about six years, I really enjoyed the time at the lab. Of course, all things taken place at the lab were not enjoyable. A lot of Hard-work, dedication, creativity, and courage were required of me in PhD degree course. When I did my first year of PhD course, I recognized that the work I dedicated to about two years went the wrong way. It was precious experience as I look back now but it was very tough time for me. I had to face a failure of my work, which seems failure of my PhD course to me at the time. Positive support and warm encouragement of lab members gave me the strength and were of help in overcoming the problem. That work eventually became the base of the chapter 2 in this thesis. I sincerely would like to express my gratitude to all lab members. Especially I am grateful to my contemporaries, Oshima, Nakayama, Numa, Aoki, Takubo, Mizukoshi, Nishizawa, Fujita, Yamaguchi, and my senior, Dr. Teramoto and Dr. Hunu. Thanks to all of you, I was nice time at the lab. Further, I would like to thank to secretary of the lab, Mrs. Mayumi Iwasa, for her assistance.

I also should be indebted to my father, mother, and sister for supporting and encouraging me since childhood with a lot of affection and for sharing their love and experiences with me. They always pray for my success and happiness. I can not imagine where I would be without them. Finally, I express my deepest gratitude to K. and Y. who encouraged me to come to study in Japan. They have always been a real inspiration to me and have remained as most reliable persons. The constant support from them led to this thesis.

Contents

Declaration of Authorship ii

Abstract iv

Acknowledgements vi

List of Figures xii

List of Tables xvi

Abbreviations xvii

Symbols xviii

1 Introduction 1
- 1.1 Background . . . 1
  - 1.1.1 IADI method . . . 2
  - 1.1.2 Coordinate transformation method . . . 3
  - 1.1.3 Iteration method for nonlinear system . . . 5
- 1.2 Objectives . . . 6
- 1.3 Outline of the thesis . . . 7

2 Development of a new iterative alternating direction implicit (IADI) algorithm 9
- 2.1 Indroduction . . . 9
- 2.2 Theory . . . 10
  - 2.2.1 Picard iterative linearization . . . 10
  - 2.2.2 Coventional IADI scheme . . . 12
  - 2.2.3 Advanced IADI (AIADI) scheme . . . 13
- 2.3 Numerical simulations . . . 16
  - 2.3.1 Test 1: Two-dimensional infiltration into dry soil . . . 18
  - 2.3.2 Test 2: Two-dimensional transient variably saturated flow . . . 22
  - 2.3.3 Test 3: Two-dimensional simulation for rainfall-runoff on a simple slope . . . 24
  - 2.3.4 Test 4: Three-dimensional infiltration into layered soil . . . 25
- 2.4 Summary . . . 30

3 Three-dimensional saturated–unsaturated flow modeling with non-orthogonal grids 33
3.1 Introduction . . . 33
3.2 Coodinate Transformation . . . 34
3.3 Numerical modeling . . . 36
3.3.1 Temporal discretization . . . 36
3.3.2 Finite-difference discretization . . . 37
3.3.3 Metrics evaluation . . . 41
3.3.4 Boundary condition . . . 42
3.3.5 Grid generation . . . 44
3.4 Numerical simulation . . . 44
3.4.1 Test 1: steady-state simulation with a highly skewed mesh . . . 45
3.4.2 Test 2: unsteady-state simulation to investigate the non-orthogonality effect . . . 46
3.4.3 Test 3: transient variably saturated flow in two dimensions . . . 52
3.4.4 Test 4: rainfall-runoff simulation for a slope in three dimensions . 56
3.5 Summary . . . 60

4 Comparison of iteration methods for saturated–unsaturated flow model 63
4.1 Introduction . . . 63
4.2 Iteration schemes . . . 64
4.2.1 Newton method . . . 65
4.2.2 Picard method . . . 65
4.2.3 Newton-Krylov method . . . 66
4.3 Finite-difference discretization . . . 67
4.4 Numerical simulations . . . 73
4.4.1 Test 1: pumping well, steady state . . . 74
4.4.2 Test 2: infiltration problem with different skewnesses and anisotropies . . . 77
4.4.3 Test 3: rainfall-runoff simulation for a curvilinear slope . . . 84
4.5 Summary . . . 88

5 Development of a coupled model of pipe-matrix subsurface flow 91
5.1 Indroduction . . . 91
5.2 Model description . . . 92
5.2.1 Basic concept of the model . . . 92
5.2.2 saturated–unsaturated flow model . . . 93
5.2.3 Pipe flow model . . . 94
5.3 Numerical simulation . . . 95
5.3.1 Simulation Condition . . . 95
5.3.2 Results and discussions . . . 98
5.4 Summary . . . 100

6 Assessment of integrated kinematic wave equations for a hillslope runoff modeling 103
6.1 Introduction . . . 103
6.2 Model description . . . 104
6.2.1 Integrated kinematic wave model . . . 104

6.2.2 2D model . . . 106
6.3 Numerical experiment . . . 107
6.3.1 Simulation condition . . . 107
6.3.2 Model parameter . . . 108
6.4 Result and discussion . . . 109
6.4.1 Runoff discharge . . . 109
6.4.2 Nash-Sutcliffe coefficient . . . 114
6.4.3 Discharge components . . . 117
6.5 Summary . . . 117

7 Conclusions **121**

Bibliography **127**

List of Figures

2.1 Test simulation 1, ψ at the end of simulation. 19
2.2 Test simulation 1, time step 20
2.3 Test simulation 1, water table positions in the cases of sand and sandy loam 20
2.4 Test simulation 1, residual norm at each calculation time step. 21
2.5 Water-table mounding data collected by *Vauclin et al.* [92] and simulation. 23
2.6 Test simulation 2, time step 23
2.7 Test simulation 3, (a) slope and (b) rainfall intensity 24
2.8 Test simulation 3, discharge at the end of the slope. 26
2.9 Test simulation 3, normalized CPU time per iteration. 26
2.10 Layered soil domain for test simulation 4. 26
2.11 Test simulation 4, ψ of vertical cross section for $y = 0$ calculated by the AIADI scheme with 20×20×160 grid 29
2.12 Test simulation 4, normalized CPU time per iteration. 29
2.13 Test simulation 4, residual norm at each calculation time step. 30

3.1 Concept of coordinate transformation: an arbitrarily shaped mesh in physical space is transformed into an orthogonal mesh in computational space. 35
3.2 Evaluated (a) 19-point stencil of ψ and (b) 7-point stencil of ψ. (c) Utilized 7-point stencil of K and G in an iterative procedure. 40
3.3 Additional row of ghost nodes (unfilled circle) with zero thickness for evaluating the metrics at boundary nodes. 42
3.4 Flux directions under various boundary conditions. 43
3.5 (a) The 10 × 10 and (b) 20 × 20 Kershaw meshes used in Test 1. 45
3.6 Isolines of pressure head with (a) 10 × 10 mesh and (b) 20 × 20 mesh and the root mean square error of pressure head for the (c) 10 × 10 mesh and (d) 20 × 20 mesh results in Test 1. 46
3.7 Four types of grids used in Test 2 (20 × 40). 47
3.8 Pressure head profiles on $x = 0.1$ m obtained at the end of simulations carried out by FDM, FI-FDM, and HYDRUS in Test 2. 49
3.9 Mass balances of FDM, FI-FDM, and HYDRUS in Test 2. 51
3.10 Flow domain in Test 3. 53
3.11 Grids used in Test 3. 53
3.12 Contours of pressure head simulated in Test 3; the left and right sides show the results obtained by FDM and HYDRUS, respectively. 54
3.13 Time-step durations of FDM and HYDRUS in Test 3. 55

3.14 An inclined domain used in Test 4. (a) Perspective view; (b) plan view; (c) vertical cross section from x,y coordinate (0,0); (d) vertical section from x,y coordinate (9,-10). 56
3.15 Grids used in Test 4; left side describes vertical sections from x,y coordinate (9,-10) and right side gives plan views. 57
3.16 Rainfall intensity for Test 4. 58
3.17 Pressure head results of Test 4; the left, middle, and right side show the results obtained by FDM, HYDRUS, and conventional FDM, respectively. 58
3.18 Discharge flow rate (thin line) and cumulative water volume (thick line) at the lower end of the slope in Test 4. 59
3.19 The time-step durations of FDM, HYDRUS, and conventional FDM for Test 4. 60

4.1 Flow domain and grid used in Test 1. (a) perspective view; (b) plane view and circle represents the place of pumping well. 74
4.2 Pressure head results of Test 2 performed by the Newton method. A cross sectional distribution for $x = 626$, $y = 626$ and $z = 0$ m. 75
4.3 Convergence behavior of three iteration methods with line search method; L_1 residual norm are plotted. 76
4.4 Different grids used in Test 2. 78
4.5 Frequency of a when σ is 0.1, 0.2, 0.3, 0.4 and 0.5 (10000 samples). 79
4.6 Test simulation 2, distribution of RCT. 80
4.7 Test simulation 2, pressure head results of sand performed by the Newton method. 83
4.8 Curvilinear slope and grid used in Test 3. (a) perspective view; (b) plane view. 85
4.9 Pressure head and flux results of Test 3 at the end of simulation performed by the Newton method. (a,b) pressure head at surface; (c,d) pressure head for saturated zone; (e,f) flux vector. Surface pressure head was also plotted transparently in (c,d,e,f). 86
4.10 Seepage flux at the lower and surface boundaries using the Newton method. 87
4.11 Test simulation 3, distribution of RCT. 87

5.1 Flowchart of the coupling between the slot model and the saturated-unsatirated model 93
5.2 Schematic representation of slot model 94
5.3 Cross-sectional shape of the pipe in the slot model 96
5.4 Schematic diagram of the experimental setup, (a) open pipe; (b) closed pipe 98
5.5 Comparison of the water surface profile between the simulation results and the expreriment observations. The dimensions are cm in figures. (a) no pipe; (b) open pipe; (c) closed pipe condition 99
5.6 Percentage of the pipe flow 99
5.7 Water flux vector and water surface profile on $y - z$ plane at the upper end of open pipe; (b) at the lower end of closed pipe 101

6.1 Concept of integrated kinematic wave model [83]. 105
6.2 2D model, the subsurface flow domain. 106
6.3 Rainfall intensity, total rainfall is 10 mm. 108

6.4 Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 10 mm. 109
6.5 Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 20 mm. 110
6.6 Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 40 mm. 111
6.7 Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 70 mm. 112
6.8 Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 100 mm. 113
6.9 Nash-Sutcliffe coefficient with different soil depths: (a) wet, (b) dry initial condition. 115
6.10 Nash-Sutcliffe coefficient with different slope angles: (a) wet, (b) dry initial condition. 116
6.11 Matrix and overland flows simulated by the 2D model and the integrated kinematic wave model proposed by *Tachikawa et al.* [83] and *TaKasao & Shiiba* [84] (100 mm, $D = 0.25$ m, 5 degree, wet). 118

List of Tables

2.1 Soil properties of major soil textures referring to *Carsel & Parrish* [9] . . 17
2.2 Test simulation 1, performances of the schemes. 21
2.3 Test simulation 2, performances of the schemes. 24
2.4 Test simulation 3, performances of the schemes. 25
2.5 Test simulation 4, performances of the schemes. 28

3.1 Test simulation 2, Relative error. 48
3.2 Test simulation 2, Mass balance error. 50
3.3 Test simulation 2, model performance. 51
3.4 Test simulation 3, numerical accuracy and performance. 55
3.5 Test simulation 4, mass balance error and model performance. 60

4.1 Test simulation 1 with line search method, total iteration number and CPU time. 74
4.2 Test simulation 2, MRCT. 78
4.3 Test simulation 2, total iteration number when sand is used. 81
4.4 Test simulation 2, total iteration number when loam is used. 81
4.5 Test simulation 2, CPU time when sand is used. 82
4.6 Test simulation 2, CPU time when loam is used. 82
4.7 Test simulation 3, total iteration number. 87
4.8 Test simulation 3, CPU time. 87

5.1 Soil properties reffering to *Kosugi et al.* [49]. 97

6.1 Simulation condition 107

Abbreviations

2D	**2**-**D**imensional
3D	**3**-**D**imensional
ADI	**A**lternating **D**irection **I**mplicit
AIADI	**A**dvanced **I**terative **A**lternating **D**irection **I**mplicit
BICG	**BIC**onjugate **G**radient
BICGSTAB	**BIC**onjugate **G**radient **STAB**ilized
CFD	**C**omputational **F**luid **D**ynamic
FDM	**F**inite-**D**ifference **M**odel
FEM	**F**inite-**E**lement **M**odel
FI-FDM	**F**ully **i**mplicit **F**inite-**D**ifference **M**odel
GMRES	**G**neralized **M**inimal **RES**idual
IADI	**I**terative **A**lternating **D**irection **I**mplicit
ILU	**I**ncomplete **L**ower-**U**pper
LIS	**L**ibrary of **I**terative **S**olvers for linear system
MBE	**M**ass **B**alance **E**rror
NSE	**N**ash-**S**utcliffe co**E**fficient
PDE	**P**artial **D**ifferential **E**quation
RE	**R**elative **E**rror
RNT	**R**atio of the **N**ormal-derivative **T**erms
SSOR	**S**ymmetric **S**uccessive **O**ver-**R**elaxation

Symbols

ψ	pressure head	m
θ	volumetric moisture content	
θ_s	saturated water content	
θ_r	residual water content	
α	Van Genuchten parameter	m^{-1}
δ_ψ	convergence tolerance for ψ	m
δ_θ	convergence tolerance for θ	
C	specific moisture capacity function	m^{-1}
G	mesh skeness and anisotropy tensor	m^{-2}
H	mesh skeness and anisotropy tensor	m^{-1}
I_m	iteration parameter used in IADI method	
J	Jacobian determinant of the coordinate transformation	m^{-3}
K	hydraulic conductivity	m/s
K_s	saturated hydraulic conductivity	m/s
n	Van Genuchten parameter	m^{-1}
S_e	effective saturation	
w	gradient of the slope	rads

Chapter 1

Introduction

1.1 Background

The modeling of saturated–unsaturated flow through porous media is an important research topic in various branches of water resources engineering, agricultural engineering, chemical contaminant tracing, and rainfall runoff modeling. Although several analytical solutions of the governing equations of saturated–unsaturated flow through porous media have been reported [7, 28, 59, 67, 72], these solutions are generally obtained under simple initial and boundary conditions. Hence, numerical models are usually used to investigate saturated–unsaturated flow in porous media, where analytical solutions are not appropriate. Particularly, in hillslope hydrology, saturated–unsaturated subsurface flow models are used as tools to analyze the runoff processes obtained by observations or to better understand the hydrological processes of a hillslope. For example, *Hopp & McDonnell* [29] performed numerical experiments by controlling the storm size, slope angle, soil depth, and bedrock permeability to study their effect on hillslope runoff using a physical model that is based on the saturated–unsaturated flow model. *Keim et al.* [38] performed virtual experiments to investigate the process of evaporation and canopy interception in a hillslope using the saturated–unsaturated flow model. *Liang et al.* [54] tried to simulate stemflow in soil water dynamics around a tree on a hillslope using the saturated–unsaturated flow model and compared the results obtained by the model with observation data. Further, the saturated–unsaturated subsurface flow model has been widely used for simulating the nonequilibrium and preferential flow in many recent researches [25, 26, 35, 73, 82].

Over the last three decades, many numerical models including finite-difference models (FDMs) and finite-element models (FEMs) have been developed for simulating saturated–unsaturated flow [10, 15, 24, 36, 81, 88]. Other approaches such as a finite-volume approach, a mixed finite-element approach, and an Eulerian-Lagrangian approach have also been developed for simulating saturated–unsaturated flow [4, 21, 31, 57]. In particular, FDMs have certain advantages with respect to the ease of coding and understanding, owing to their simplicity of discretization as compared to the other models. However, a disadvantage of FDMs has been often pointed out: FDMs do not accurately represent all geometrically complex flow domains with low resolution, especially in multidimensional simulations. This thesis focuses on the modeling of FDMs and tries to overcome their disadvantages.

Computational cost is also an important issue in the modeling of saturated–unsaturated flow in porous media. Rapid developments in computer technology have made it possible to carry out not only one-dimensional (1D) [17, 27] but also multidimensional saturated–unsaturated flow simulations [15, 18, 81, 94] using a personal computer. However, a multidimensional subsurface flow simulation, particularly a simulation in a wide region with a relatively fine grid resolution, still requires a large amount of computer resources. In particular, in the case of iterative parameter estimation or Monte Carlo exercises, the simulation costs could overwhelm users because these simulations usually require hundreds or thousands of runs to arrive at an ideal parameter set or an objective function. In this thesis, the iterative alternating direction implicit (IADI) approach is considered in an attempt to improve the computational cost problem.

1.1.1 IADI method

The alternating direction implicit (ADI) approach and IADI were very popular in the 1970s for avoiding the solution of large, sparse, linear systems arising from implicit discretization of parabolic partial differential equations in two and three dimensions. The IADI scheme is an iterative adaptation of the ADI method: IADI discretizes the equation into a simultaneous system of difference equations that are solved iteratively. Subsequently, the method has been rarely used in favor of a preconditioned Krylov subspace iteration and even sparse direct solvers. However, the IADI approach has advantages over the Krylov solvers in terms of simplicity and cost (on a per iteration

basis) because only tridiagonal linear systems are involved in the calculation procedure. Optimal Krylov subspace solvers need preconditioners based on a multigrid or a domain decomposition, which introduce considerably more programming complexity than the IADI method. Furthermore, the computational cost for tridiagonal linear systems is comparatively cheap and proportional to the problem dimensions. This implies that the computational cost of the IADI method is expected to be scalable whereas the computational cost of preconditioned Krylov subspace solvers typically increases faster than the problem dimension does. Therefore, if the IADI algorithm can overcome shortcomings such as instability and convergence difficulties, which will be discussed in the next paragraph, it can be an attractive alternative for simulating a saturated–unsaturated flow in porous media.

The study by *Rubin* [76] is probably the earliest study on the simulation of a two-dimensional (2D) transient groundwater flow using the IADI method. Following this study, the IADI method has been used in several studies [e.g., 16, 68, 71, 94] to simulate a 2D saturated–unsaturated flow in porous media. All these models solved the pressure-head-based form of the Richards equation [75]. However, *Celia et al.* [10] stated that numerical methods using the pressure-head-based form of the Richards equation result in poor mass balance in the unsaturated zone because of the highly nonlinear constitutive relationship between the pressure head and the moisture content. These researchers showed that solutions based on a mixed form of the Richards equation satisfy the mass balance and are more accurate than those obtained by using the pressure-head-based form. Moreover, *Clement et al.* [15] claims that the IADI scheme is not robust because it results in numerical instabilities and convergence difficulties in solving 2D nonlinear equations. This is one of the reasons why the IADI technique is rarely used at present. Another reason might be that the conventional IADI scheme cannot be used for simulating three-dimensional (3D) problems.

1.1.2 Coordinate transformation method

There are certain advantages of the FDMs, as previously mentioned. However, a disadvantage of FDMs has been also often pointed out: FDMs do not accurately represent all geometrically complex flow domains with low resolution, especially in multidimensional simulations. High-resolution orthogonal grids are required for domains with inherently

curvilinear features such as a foundation fit [34], an embarkment dam [6], and shallow groundwater flow with a curvilinear boundary [12, 54]. For these problems, the FDM is computationally inefficient in comparison to the FEMs and finite-volume models (FVMs) that can treat nonorthogonal grids. These models can accommodate a curvilinear domain with comparatively low-resolution grids because they can discretize the governing equation of saturated–unsaturated flow without using an orthogonal coordinate system [57, 81]. If the domain shape is extremely complex, this inefficiency can be getting worse. Furthermore, the principal axes of anisotropy are typically aligned with the orthogonal axes in FDMs, which rely on the orthogonal grids. In the case of these models, it is mandatory for the principal axes of anisotropy to be uniformly oriented in the same direction throughout the flow domain.

To solve this problem, a coordinate transformation method [e.g., 34, 41, 45, 46, 77] or an adaptively refined grid approach [53] could be considered. In particular, the coordinate transformation method based on tensor analysis has been commonly applied to the general Navier-Stokes equation in computational fluid dynamics (CFD) [14, 30, 55, 95]; it has also been used for describing circulation and transport in estuaries and oceans [13, 62]. Further, this method was also successfully applied to heat transport [51, 77] and groundwater modeling [34, 41, 45, 46, 77]. *Koo & Leap* [45, 46] proposed an FDM for groundwater flow, in which they used a successive over-relaxation (SOR) method to solve the system of equations. The cross-derivative terms were evaluated at a previous time iteration level for convergence stability, because the matrix of the system equation should be diagonally dominant when the SOR method is used. *Jie et al.* [34] carried out steady-state groundwater modeling to analyze the seepage flow for a foundation fit, a lock foundation, and an embarkment dam using a coordinate-transformed FDM. *Ruhaak et al.* [77] applied a coordinate transformation method to an FVM for simulating heat transport and groundwater flow. They solved the cross-derivative terms explicitly and the other terms implicitly. The above three models simulated only saturated flow. *Kinouchi et al.* [41] applied a coordinate transformation method to an FDM for simulating 2D unsaturated flow in porous media. They solved the pressure-head-based Richards equation, in which the transformed equation took a nonconservative form because the mesh skewness tensor was cast outside the differential operator. Apart from the above example, it appears that little attention has been paid to the coordinate transformation method for saturated–unsaturated flow simulation. However, we believe that this

method can relax the constraint of representation of a curvilinear shape and allow for the principal axes of anisotropy to align with a curvilinear surface, which gradually and continuously changes orientation throughout the flow domain in a saturated–unsaturated flow model.

1.1.3 Iteration method for nonlinear system

Because saturated–unsaturated flow equation systems are highly nonlinear, implicit temporal discretization and iterative procedures are generally needed for numerical stability. The Picard and Newton iteration methods might be the most common approaches for modeling the saturated–unsaturated flow. The Picard method is simple to implement and cost efficient (per iteration basis). However, it converges linearly. On the other hand, the Newton method is comparatively complex and consumes more CPU resources than the Picard method. However, it converges quadrically. *Lehmann & Ackerer* [52], *Paniconi et al.* [65], *Paniconi & Putti* [66] compared the performances of FEMs linearized by the Picard and Newton methods for 1D, 2D, and 3D problems. They showed that the Newton method is generally more robust and converges faster than the Picard method. However, the Newton method was more sensitive to the initial estimation than the Picard method and often failed to converge in particular problems, especially in steady-state problems.

Coordinate transformation represents diffusion with cross-derivative terms. Therefore, the transformed equation requires a 19-point stencil instead of a 7-point stencil, which is required by the conventional FDM with an orthogonal grid. All the terms including the cross-derivative terms are quite difficult to treat in an implicit manner in practical aspects because they make the programming extremely complex. Therefore, the usual option is that the cross-derivative terms are evaluated in an explicit manner and the other normal-derivative terms are evaluated in an implicit manner. In this approach, the 7-point stencil matrix is formed at the iteration, as in the case of the conventional FDM. Therefore, the behaviors of the Picard and Newton methods in the coordinate-transformed FDM are expected to be different from those of the FEM observed in previous studies.

Apart from the evaluation of the cross-derivative terms in an explicit manner, an alternative approach for avoiding complexity is a combination of the Newton method for

linearization of nonlinear systems and the Krylov subspace method for solving linearized simultaneous equations, called the Newton-Krylov method. The Newton-Krylov method does not require the direct formation of a 19-point stencil matrix. Instead, it requires the calculation of a matrix-vector product, which can be approximated by calculating the differences of the original nonlinear function. This feature is particularly favorable in cases where forming the matrix is laborious, for example, when the coordinated transformation is applied or when the anisotropic flow domain is to be considered. In this respect, the Newton-Krylov method is probably the most appropriate iteration method for coordinate-transformed saturated–unsaturated flow equations. However, it should be noted that in the Newton-Krylov method, there is an additional cost for calculating the matrix-vector product at every Krylov iteration step. Therefore, it is unclear whether the Newton-Krylov method is more efficient than the 7-point stencil Newton or Picard methods in the case of coordinate-transformed FDM of saturated–unsaturated flow in porous media.

1.2 Objectives

Considering the background mentioned above, the main objectives of this thesis are as follows:

1. Development of a new IADI algorithm to overcome the numerical instabilities and the limitations with respect to the applicability to three-dimensional simulations of the conventional IADI method of *Rubin* [76]. The developed algorithm is supposed to be applied to not only 2D but also 3D problems and shows improved stability.

2. Development of a saturated–unsaturated flow model with a nonorthogonal grid, using the coordinate transformation method. The coordinate transformation method has been used to overcome the limitations with respect to the applicability to complex flow domain. The coordinate-transformed equation is supposed to be in a mass conservative form. At first, the 7-point stencil Picard method is used to avoid the complexity of programming, in which the cross-derivative terms are calculated in an explicit manner.

3. Comparison of three iteration methods, the Picard, Newton, and Newton-Krylov methods, for saturated–unsaturated flow modeling. The Picard and Newton methods implement the 7-point stencil strategy, calculating the cross-derivative terms in an explicit manner, and the Newton-Krylov method considers all terms in an implicit manner without calculating the 19-point stencil matrix directly. Through 3D test problems with a curvilinear flow domain, the performances and robustness of these three iteration methods are compared.

The motivation of the research on the modeling of saturated–unsaturated subsurface flow arose from the experience of applying the conventional finite-difference saturated–unsaturated flow model to a rainfall runoff simulation on a hillslope. However, the conventional 3D model consumed huge computational resources and provided a restricted representation of a complex flow domain. Hence, our first objective was the development of an efficient and robust numerical model for simulating a 3D saturated–unsaturated subsurface flow model. In addition to the development of the numerical model, two application studies of the saturated–unsaturated subsurface flow model are presented in this thesis: (1) development of a coupled model of pipe-matrix flow and (2) assessment of integrated kinematic wave equations for hillslope modeling using the saturated–unsaturated subsurface flow model. The backgrounds of these two application studies will be presented in Chapters 5 and 6.

1.3 Outline of the thesis

The general background and objectives of this thesis are described in previous sections. The remainder of this thesis is organized as Chapter 2∼4 describing the modeling of saturated–unsaturated flow and Chapter 5∼6 describing applications of the model.

Chapter 2 develops a new IADI algorithm for saturated–unsaturated flow in porous media. The new algorithm can be applied to 2D and 3D problems and shows improved stability. To evaluate the proposed method, four test simulations were conducted, and the results were compared with those obtained by using the conventional IADI method and the fully implicit scheme linearized by the Picard iteration method.

Chapter 3 presents a 3D saturated–unsaturated flow model with nonorthogonal grids using the coordinate transformation method. The performance of the proposed

model is assessed by carrying out test simulations. We then compare the simulation results with the dense-grid solutions to evaluate the numerical accuracy of the proposed model. To examine the performance of the proposed model, we also draw a comparison between the simulations obtained by the proposed model, those obtained by a model in which all terms are considered fully implicitly, those obtained by a FEM, and those obtained by a conventional FDM with a high-resolution orthogonal grid. The contents of this chapter have been published in a journal paper [2].

Chapter 4 compares the performances of the three iteration methods, the Picard, Newton, and Newton-Krylov methods. Three test simulations are carried out by varying the magnitude of effectiveness of the cross-derivative terms, which is affected by the grid skewness and the anisotropy of the flow domains. The robustness of the three methods is also investigated.

Chapter 5 develops the coupled model of pipe-matrix subsurface flow combining the 3D saturated–unsaturated flow model and the slot model. To test the proposed model, the simulations are carried out for three different conditions (no pipe, open pipe, and closed pipe). The results obtained by the proposed model are compared with the observation data of the experiment. The contents of this chapter have been published in a journal paper [1].

Chapter 6 assesses the validity of the integrated kinematic wave model through numerical experiments. By controlling the slope angle, soil depth, total rainfall and initial conditions, numerical experiments were conducted using a physical 2D model, which is a combination of the 2D saturated–unsaturated flow model and the kinematic wave model. The integrated kinematic wave model was calibrated to reproduce the results of the 2D model and the results obtained by the two models were compared. The results obtained by the 2D model are considered as a surrogate of observations and the validity of the integrated kinematic wave model was assessed. The contents of this chapter have been published in journal paper [3].

Finally, **Chapter 7** presents the conclusions of the thesis.

Chapter 2

Development of a new iterative alternating direction implicit (IADI) algorithm

2.1 Indroduction

The IADI approach has advantages in terms of simplicity and computational cost (on a per iteration basis) because only tridiagonal linear systems are involved in the calculation procedure. However, at the same time, the saturated–unsaturated subsurface flow model using the IADI method resulted in numerical instabilities and convergence difficulties as mentioned in Chapter 1. Further, the conventional IADI method cannot apply to three-dimensional problems. Those might be the reasons that the IADI method are being rarely used for simulating saturated–unsaturated flow in porous media in recent year.

In this chapter, in order to overcome the numerical instabilities and the limitations with respect to the applicability to three-dimensional simulations of the conventional IADI method of *Rubin* [76], we derived a new equation from the ADI method of *Douglas & Rachford* [19]. The newly derived equation can be applied to two and three-dimensional problems and shows improved stability. In order to evaluate the proposed method, four test simulations were conducted, and the results were compared with those

obtained by using the conventional IADI scheme and the fully implicit scheme linearized by the Picard iteration method.

2.2 Theory

Richards' equation, which has typically been used to simulate saturated–unsaturated flow, is written as

$$\frac{\partial \theta(\psi)}{\partial t} = \nabla \cdot K(\psi) \nabla \psi + \frac{\partial K(\psi)}{\partial z}, \tag{2.1}$$

where ψ is the pressure head, θ is the volumetric moisture content, K is the hydraulic conductivity, t denotes the time, and z is the vertical dimension, assumed to be positive upwards. It is also assumed that the appropriate constitutive relationships between θ and ψ and those between ψ and K are available. The source/sink term has been ignored for simplicity. Equation (2.1) includes both θ and ψ and is thus called the mixed form of Richards' equation. This form is generally considered to have advantages over the other two forms, namely, the ψ-based and θ-based forms, because of perfect mass balance [10, 56].

2.2.1 Picard iterative linearization

The backward Euler scheme is the one of the most widely used time approximations for the Richards equation and is used in this study. Since this equation is nonlinear because of the nonlinear dependency of θ on ψ and K on ψ, iterative calculation and linearization are required. Although several iterative schemes have been proposed [e.g., 5, 22, 37, 65, 66], from a practical aspect, the Picard method is used in this chapter because it is simple and exhibits a good performance in many problems [52, 65, 66]. The backward Euler approximation and Picard linearization of the two-dimensional Eq. (2.1) is written as

$$\begin{aligned} &\frac{\theta^{n+1,m+1} - \theta^{n}}{\Delta t} \\ = \; &\frac{\partial}{\partial x}\left\{ K^{n+1,m} \left.\frac{\partial \psi}{\partial x}\right|^{n+1,m+1} \right\} + \frac{\partial}{\partial z}\left\{ K^{n+1,m} \left.\frac{\partial \psi}{\partial z}\right|^{n+1,m+1} \right\} + \left.\frac{\partial K}{\partial z}\right|^{n+1,m} , \end{aligned} \tag{2.2}$$

where the superscripts n and m denote the time level and the iteration level, respectively, and x denotes the horizontal dimension.

The moisture content at the new time step and a new iteration level ($\theta^{n+1,m+1}$) is replaced with the Taylor series expansion with respect to ψ, around the expansion point $\psi^{n+1,m}$ as follows:

$$\theta^{n+1,m+1} = \theta^{n+1,m} + \left.\frac{d\theta}{d\psi}\right|^{n+1,m} (\psi^{n+1,m+1} - \psi^{n+1,m}) + O(\delta^2). \tag{2.3}$$

By neglecting the higher-order terms in Eq. (2.3) and substituting this equation into Eq. (2.2), we obtain

$$\begin{aligned} & C^{n+1,m}\frac{\psi^{n+1,m+1} - \psi^{n+1,m}}{\Delta t} + \frac{\theta^{n+1,m} - \theta^{n}}{\Delta t} \\ = \; & \frac{\partial}{\partial x}\left\{ K^{n+1,m}\left.\frac{\partial \psi}{\partial x}\right|^{n+1,m+1}\right\} + \frac{\partial}{\partial z}\left\{ K^{n+1,m}\left.\frac{\partial \psi}{\partial z}\right|^{n+1,m+1}\right\} + \left.\frac{\partial K}{\partial z}\right|^{n+1,m}, \end{aligned} \tag{2.4}$$

where $C(= d\theta/d\psi)$ is the specific moisture capacity function. A finite-difference approximation of Eq. (2.4) can be written as

$$\begin{aligned} & C_{i,j}^{n+1,m}\frac{\psi_{i,j}^{n+1,m+1} - \psi_{i,j}^{n+1,m}}{\Delta t} + \frac{\theta_{i,j}^{n+1,m} - \theta_{i,j}^{n}}{\Delta t} \\ = \; & \Delta_x(K^{n+1,m}\Delta_x\psi^{n+1,m+1}) + \Delta_z(K^{n+1,m}\Delta_z\psi^{n+1,m+1}) + \Delta_z(K^{n+1,m}), \end{aligned} \tag{2.5}$$

where

$$\begin{aligned} \Delta_x(K^{n+1,m}\Delta_x\psi^{n+1,m+1}) &= \frac{1}{\Delta x^2}K_{i+1/2,j}^{n+1,m}(\psi_{i+1,j}^{n+1,m+1} - \psi_{i,j}^{n+1,m+1}) \\ &\quad -\frac{1}{\Delta x^2}K_{i-1/2,j}^{n+1,m}(\psi_{i,j}^{n+1,m+1} - \psi_{i-1,j}^{n+1,m+1}), \\ \Delta_z(K^{n+1,m}\Delta_z\psi^{n+1,m+1}) &= \frac{1}{\Delta z^2}K_{i,j+1/2}^{n+1,m}(\psi_{i,j+1}^{n+1,m+1} - \psi_{i,j}^{n+1,m+1}) \\ &\quad -\frac{1}{\Delta z^2}K_{i,j-1/2}^{n+1,m}(\psi_{i,j}^{n+1,m+1} - \psi_{i,j-1}^{n+1,m+1}), \\ \Delta_z(K^{n+1,m}) &= \frac{K_{i,j+1/2}^{n+1,m} - K_{i,j-1/2}^{n+1,m}}{\Delta z}, \end{aligned} \tag{2.6}$$

and subscripts i and j denote the spatial coordinates in the x and z axes, respectively. Eq. (2.5) represents the same method proposed by *Clement et al.* [15], except that this equation ignores the specific storage term. These linearized simultaneous equations are solved using matrix solvers such as the LU decomposition or preconditioned conjugated gradient methods. In this chapter, a library of iterative solvers for linear systems (LIS), developed by *Kotakemori et al.* [50], is used for solving the linear equations. LIS provides several types of preconditioners and Krylov iterative solvers for linear systems. While conducting the test simulations, we selected a pair of SSOR preconditioners and biconjugate gradient stabilized (BICGSTAB) methods, which was shown to be faster and more stable than the other pairs provided by the LIS library. The pressure head at the $(n+1)$th time level and $(m+1)$th Picard iteration level was obtained by solving Eq. (2.5). The iteration process of Eq. (2.5) continued until the difference between the calculated values of the pressure head of two successive iteration levels became less than the tolerance, i.e., until the following inequality was satisfied for all grid points:

$$|\psi^{n+1,m+1} - \psi^{n+1,m}| \leq \delta_{\psi}. \tag{2.7}$$

where δ_{ψ} is the convergence tolerance, whose value is sufficiently small to be neglected.

2.2.2 Coventional IADI scheme

When a fine mesh is used, normal implicit schemes such as Eq. (2.5) result in a considerably high CPU cost. However, the IADI method can carry out these calculations more efficiently because large simultaneous equations do not need to be solved using the IADI algorithm.

In two-dimensional IADI procedures, each full time step is achieved by an iterative correction of two forward discretization passes. The first forward pass is based on a horizontal discretization to determine the approximate values for the pressure head at the new iterative step level using the values of the current iterative level. The second forward corrective pass is based on a vertical discretization using values approximated at the first forward pass. Therefore, the scheme involves the performance of a pair of passes to complete a full iteration.

The conventional IADI scheme of *Rubin* [76] used the following ψ-based form of the Richards equation:

$$C(\psi)\frac{\partial \psi}{\partial t} = \nabla \cdot K(\psi)\nabla\psi + \frac{\partial K(\psi)}{\partial z}. \tag{2.8}$$

A time and spatial discretization of the two-dimensional Eq. (2.8) is written as

$$\begin{aligned} & C_{i,j}^{n+1,2m}\frac{\psi_{i,j}^{n+1,2m+1} - \psi_{i,j}^{n}}{\Delta t} + I_m \overline{K}_{i,j}^{n}(\psi_{i,j}^{n+1,2m+1} - \psi_{i,j}^{n+1,2m}) \\ = \; & \Delta_x(K^{n+1,2m}\Delta_x\psi^{n+1,2m+1}) + \Delta_z(K^{n+1,2m}\Delta_z\psi^{n+1,2m}) + \Delta_z(K^{n+1,2m}), \end{aligned} \tag{2.9}$$

$$\begin{aligned} & C_{i,j}^{n+1,2m}\frac{\psi_{i,j}^{n+1,2m+2} - \psi_{i,j}^{n}}{\Delta t} + I_m \overline{K}_{i,j}^{n}(\psi_{i,j}^{n+1,2m+2} - \psi_{i,j}^{n+1,2m+1}) \\ = \; & \Delta_x(K^{n+1,2m}\Delta_x\psi^{n+1,2m+1}) + \Delta_z(K^{n+1,2m}\Delta_z\psi^{n+1,2m+2}) + \Delta_z(K^{n+1,2m}), \end{aligned} \tag{2.10}$$

where

$$\overline{K}_{i,j}^{n} = K_{i-1/2,j}^{n} + K_{i+1/2,j}^{n} + K_{i,j-1/2}^{n} + K_{i,j+1/2}^{n}, \tag{2.11}$$

and I_m is an iteration parameter that has been used in various forms [68, 71]. In this study, $I_m = 0.55^m$ is used according to *Weeks et al.* [94]. The convergence criterion is as follows:

$$|\psi^{n+1,2m+2} - \psi^{n+1,2m}| \leq \delta_\psi. \tag{2.12}$$

The linearized simultaneous equations of each pass can be efficiently solved using a tridiagonal matrix solver. It is supposed that these equations are derived on the basis of the ADI method of *Peaceman & Rachford* [69]. However, under partially saturated conditions, this scheme often induces instability on the convergence and becomes inefficient.

2.2.3 Advanced IADI (AIADI) scheme

Here, a new IADI equation is derived on the basis of the ADI algorithm of *Douglas & Rachford* [19] in order to eliminate the previously mentioned numerical instability. The Douglas-Rachford ADI scheme is unconditionally stable for two- and three-dimensional parabolic partial differential equations while the Peaceman-Rachford ADI method is

unconditionally stable for only two-dimensional equations. The newly derived IADI scheme is given as follows:

$$\begin{aligned} & C_{i,j}^{n+1,2m}\frac{\psi_{i,j}^{n+1,2m+1}-\psi_{i,j}^{n+1,2m}}{\Delta t}+\frac{\theta_{i,j}^{n+1,2m}-\theta_{i,j}^{n}}{\Delta t} \\ & +I_m\overline{K}_{i,j}^{n+1,2m}(\psi_{i,j}^{n+1,2m+1}-\psi_{i,j}^{n+1,2m}) \\ = \;& \Delta_x(K^{n+1,2m}\Delta_x\psi^{n+1,2m+1})+\Delta_z(K^{n+1,2m}\Delta_z\psi^{n+1,2m})+\Delta_z(K^{n+1,2m}), \end{aligned} \tag{2.13}$$

$$\begin{aligned} & C_{i,j}^{n+1,2m}\frac{\psi_{i,j}^{n+1,2m+2}-\psi_{i,j}^{n+1,2m+1}}{\Delta t}+I_m\overline{K}_{i,j}^{n+1,2m}(\psi_{i,j}^{n+1,2m+2}-\psi_{i,j}^{n+1,2m+1}) \\ = \;& \Delta_z(K^{n+1,2m}\Delta_z\psi^{n+1,2m+2})-\Delta_z(K^{n+1,2m}\Delta_z\psi^{n+1,2m}), \end{aligned} \tag{2.14}$$

where

$$\overline{K}_{i,j}^{n,2m}=K_{i-1/2,j}^{n,2m}+K_{i+1/2,j}^{n,2m}+K_{i,j-1/2}^{n,2m}+K_{i,j+1/2}^{n,2m}, \tag{2.15}$$

and the convergence criterion is as given in Eq. (2.12).

Eqs. (2.13) and (2.14) can be rewritten in a one-pass equation form, which provides the theoretical basis of the new IADI scheme. By multiplying both sides of the equation with

$$a_m=\frac{\Delta t}{C^{n+1,2m}+\Delta t I_m\overline{K}_{i,j}^{n+1,2m}}, \tag{2.16}$$

we obtain the following equations:

$$\begin{aligned} (I+A_x)\psi^{n+1,2m+1} &= (I-A_z)\psi^{n+1,2m} \\ &\quad +a_m\left\{\Delta_z(K^{n+1,2m})-\frac{\theta_{i,j}^{n+1,2m}-\theta_{i,j}^{n}}{\Delta t}\right\}, \end{aligned} \tag{2.17}$$

$$(I+A_z)\psi^{n+1,2m+2} = \psi_{i,j}^{n+1,2m+1}+A_z\psi^{n+1,2m}, \tag{2.18}$$

where

$$
\begin{aligned}
A_x\psi^{n+1,2m+1} &= -a_m\Delta_x(K^{n+1,2m}\Delta_x\psi^{n+1,2m+1}) \\
&= -\frac{a_m K^{n+1,2m}_{i+1/2,j}}{\Delta x^2}(\psi^{n+1,2m+1}_{i+1,j}-\psi^{n+1,2m+1}_{i,j}) \\
&\quad +\frac{a_m K^{n+1,2m}_{i-1/2,j}}{\Delta x^2}(\psi^{n+1,2m+1}_{i,j}-\psi^{n+1,2m+1}_{i-1,j}), \\
A_z\psi^{n+1,2m+2} &= -a_m\Delta_z(K^{n+1,2m}\Delta_z\psi^{n+1,2m+2}) \\
&= -\frac{a_m K^{n+1,2m}_{i,j+1/2}}{\Delta z^2}(\psi^{n+1,2m+2}_{i,j+1}-\psi^{n+1,2m+2}_{i,j}) \\
&\quad +\frac{a_m K^{n+1,2m}_{i,j-1/2}}{\Delta z^2}(\psi^{n+1,2m+2}_{i,j}-\psi^{n+1,2m+2}_{i,j-1}), \\
A_z\psi^{n+1,2m} &= -a_m\Delta_z(K^{n+1,2m}\Delta_z\psi^{n+1,2m}), \\
I\psi^{n+1,2m+2} &= \psi^{n+1,2m+2}_{i,j}.
\end{aligned}
\tag{2.19}
$$

By eliminating the predictor $\psi^{n+1,2m+1}$ from Eq. (2.17) by means of Eq. (2.18), we obtain

$$
\begin{aligned}
(I+A_x)(I+A_z)\psi^{n+1,2m+2} &= \psi^{n+1,2m}_{i,j}+A_xA_z\psi^{n+1,2m} \\
&\quad +a_m\left\{\Delta_z(K^{n+1,2m})-\frac{\theta^{n+1,2m}_{i,j}-\theta^{n}_{i,j}}{\Delta t}\right\}.
\end{aligned}
\tag{2.20}
$$

Eq. (2.20) can be rewritten in the following one-pass form:

$$
\begin{aligned}
&C^{n+1,2m}_{i,j}\frac{\psi^{n+1,2m+2}_{i,j}-\psi^{n+1,2m}_{i,j}}{\Delta t}+\frac{\theta^{n+1,2m}_{i,j}-\theta^{n}_{i,j}}{\Delta t} \\
&+I_m\overline{K}^{n+1,2m}_{i,j}(\psi^{n+1,2m+2}_{i,j}-\psi^{n+1,2m}_{i,j})+a_mA_xA_z(\psi^{n+1,2m+2}_{i,j}-\psi^{n+1,2m}_{i,j}) \\
= \;& \Delta_x(K^{n+1,2m}\psi^{n+1,2m+2})+\Delta_z(K^{n+1,2m}\Delta_z\psi^{n+1,2m+2})+\Delta_z(K^{n+1,2m}).
\end{aligned}
\tag{2.21}
$$

Eq. (2.21) is equivalent to Eqs. (2.13) and (2.14), and this one-pass form is considerably similar to Eq. (2.5), which is linearized via the Picard iteration method. If the third and the fourth terms of the left-hand side of Eq. (2.21) are neglected, Eqs. (2.21) and (2.5) become the same. The third and the fourth terms are added while applying the IADI scheme and have a very small effect when the iteratively updated value ($\psi^{n+1,2m+2}_{i,j}-\psi^{n+1,2m}_{i,j}$) is sufficiently low. The AIADI scheme is a perturbation form of the Euler backward scheme linearized by the Picard method; it is expected that the behaviors of

both the schemes will be similar.

2.3 Numerical simulations

Four test simulations were performed to evaluate the performance of the proposed scheme. The performances of three different schemes were compared: the implicit scheme linearized by the Picard iteration method (Eq. (2.5)), the AIADI scheme (Eqs. (2.13) and (2.14)), and the conventional IADI scheme (Eqs. (2.9) and (2.10)).

Van Genuchten [91]'s equation for the soil water retention curve and *Mualem* [61]'s equation for the unsaturated hydraulic conductivity function were used in this study. The soil water retention curve is given by

$$S_e = \frac{\theta - \theta_r}{\theta_s - \theta_r} = \left\{ \frac{1}{1 + (\alpha|\psi|)^n} \right\}^{1-1/n}, \tag{2.22}$$

where S_e is the effective saturation; θ_r and θ_s are the residual and saturated water contents, respectively; and α and n are the Van Genuchten parameters whose values depend on the soil properties. The nonlinearity of Richards' equation is attributed to the nonlinear dependency of θ on ψ, which is determined by α and n, in the case of the Van Genuchten model. In general, it should be noted that larger α and n indicate higher nonlinearity. Following Mualem's model, the unsaturated hydraulic conductivity function is given by

$$K = K_s S_e^{1/2} \left\{ 1 - (1 - S_e^{n/(n-1)})^{1-1/n} \right\}^2, \tag{2.23}$$

where K_s is the saturated hydraulic conductivity. The hydraulic conductivity of the boundary between adjacent nodes is defined as

$$K_{i+1/2,j} = \frac{1}{2} \left(K_{i,j} + K_{i+1,j} \right). \tag{2.24}$$

In all the test simulations, we considered homogeneous and isotropic soil domains and used the soil properties listed in Table 2.1, referring to *Carsel & Parrish* [9]. These values represent the average values for the selected soil water retention and hydraulic conductivity parameters for major soil textural groups.

TABLE 2.1: Soil properties of major soil textures referring to *Carsel & Parrish* [9]

Texture	θ_s m^3/m^3	θ_r m^3/m^3	K_s m/s	α m^{-1}	n
sand	0.43	0.045	8.250×10^{-5}	14.5	2.68
loamy sand	0.41	0.057	4.053×10^{-5}	12.4	2.28
sandy loam	0.41	0.065	1.228×10^{-5}	7.5	1.89
sandy clay loam	0.39	0.1	3.639×10^{-6}	5.9	1.48
loam	0.43	0.078	2.889×10^{-6}	3.6	1.56
silt	0.46	0.034	6.944×10^{-7}	1.6	1.37

The comparison of the relative accuracy of the numerical results obtained from different schemes is not easy [52]. One of the most widely used criteria for evaluating the accuracy of a numerical scheme is the mass balance error (MBE) given by *Celia et al.* [10]:

$$\text{Mass Balance Error} = \left| 1 - \frac{\text{Total additional mass in the domain}}{\text{Total net flux into the domain}} \right| \tag{2.25}$$

where the total additional mass in the domain is the difference between the mass measured at any instant t and the initial mass in the domain, and the total net flux into the domain is the flux balance integrated in time up to t. Satisfying the mass balance is a necessary but not completely adequate prerequisite for a correct solution [10, 52, 74]. Hence, the relative error is also estimated by referring to *Manzini & Ferraris* [57] as follows:

$$\text{Relative Error} = \frac{\sqrt{\sum_i (\bar{\psi}_i - \psi_i)^2}}{\sqrt{\sum_i \bar{\psi}_i^{\,2}}}, \tag{2.26}$$

where ψ_i is the ith node solution by the models and $\bar{\psi}_i$ is the ith node exact solution for the pressure head. In this chapter, the solution of a high-resolution grid is considered a surrogate for the exact solution.

With the exception of test simulation 2, in all test simulations, we use the soil properties shown in Table 2.1; these properties have been obtained from *Carsel & Parrish* [9]. They represent the average values for selected soil-water retention and hydraulic conductivity parameters for major soil textural groups.

The time-step durations were adjusted automatically on the basis of the number of iterations required for convergence at the previous time step [66]. The time-step duration

cannot be less than a preselected minimum duration, and it cannot exceed a maximum duration. If the number of iterations required for convergence is less than N_m, the time-step duration for the next time step is multiplied by C_m, a predetermined value greater than 1. If the number of iterations is greater than N_r, the time-step duration for the next time step is multiplied by C_r, a preselected value less than 1. If the number of iterations becomes greater than a prescribed N_b, the iterative process for the time level is terminated. After that, the time-step duration is multiplied by C_b, a predetermined value less than 1, and the iterative process restarts. Depending on the difficulty and size of a problem, these time-step duration control factors normally need to be adjusted to ensure good performance of the iterative scheme. In the test simulations presented in this chapter, $C_m = 1.2$, $C_r = 0.8$, $C_b = 0.33$, $N_m = 4$, $N_r = 7$ and $N_b = 20$ were used.

2.3.1 Test 1: Two-dimensional infiltration into dry soil

The stability and the performance of the methods for different soil properties were assessed by simulating the two-dimensional infiltration problem. Infiltration into dried soil is in general a challenging problem in a saturated–unsaturated simulation. In this test, a 1-m^2 soil domain was considered, and the left-side surface ($0 < x \leq 25$ cm) was under a constant infiltration ($= K_s/2$) condition. The other boundaries had no-flux boundary conditions. The initial pressure head used was $\psi(x, z, 0) = -10.0$ m; further, four types of soil texture (sand, sandy loam, loam, and silt) were used, and $\Delta x = \Delta z = 2.5$ cm. In order to confirm the result, a simulation using the implicit scheme with a dense grid ($\Delta x = \Delta z = 0.5$ cm) was also performed. The simulation periods were 8 h (sand), 50 h (sandy loam), 100 h (loam), and 400 h (silt).

Fig. 2.1 shows the result of ψ at the end of the simulation. A saturated–unsaturated flow was observed in the cases of sand and sandy loam, whereas only an unsaturated flow was observed in the cases of loam and silt. The result calculated by each of the three schemes shows good agreement with the result of the implicit scheme using the dense grid although the IADI scheme exhibits a divergence in the cases of sand and sandy loam.

Fig. 2.2 describes the calculation time step. The AIADI and the implicit scheme had a similar time step. Oscillation was observed in the time step of the IADI scheme during the tests using sand and sandy loam. This oscillation occurred at 5 h 10 min for sand

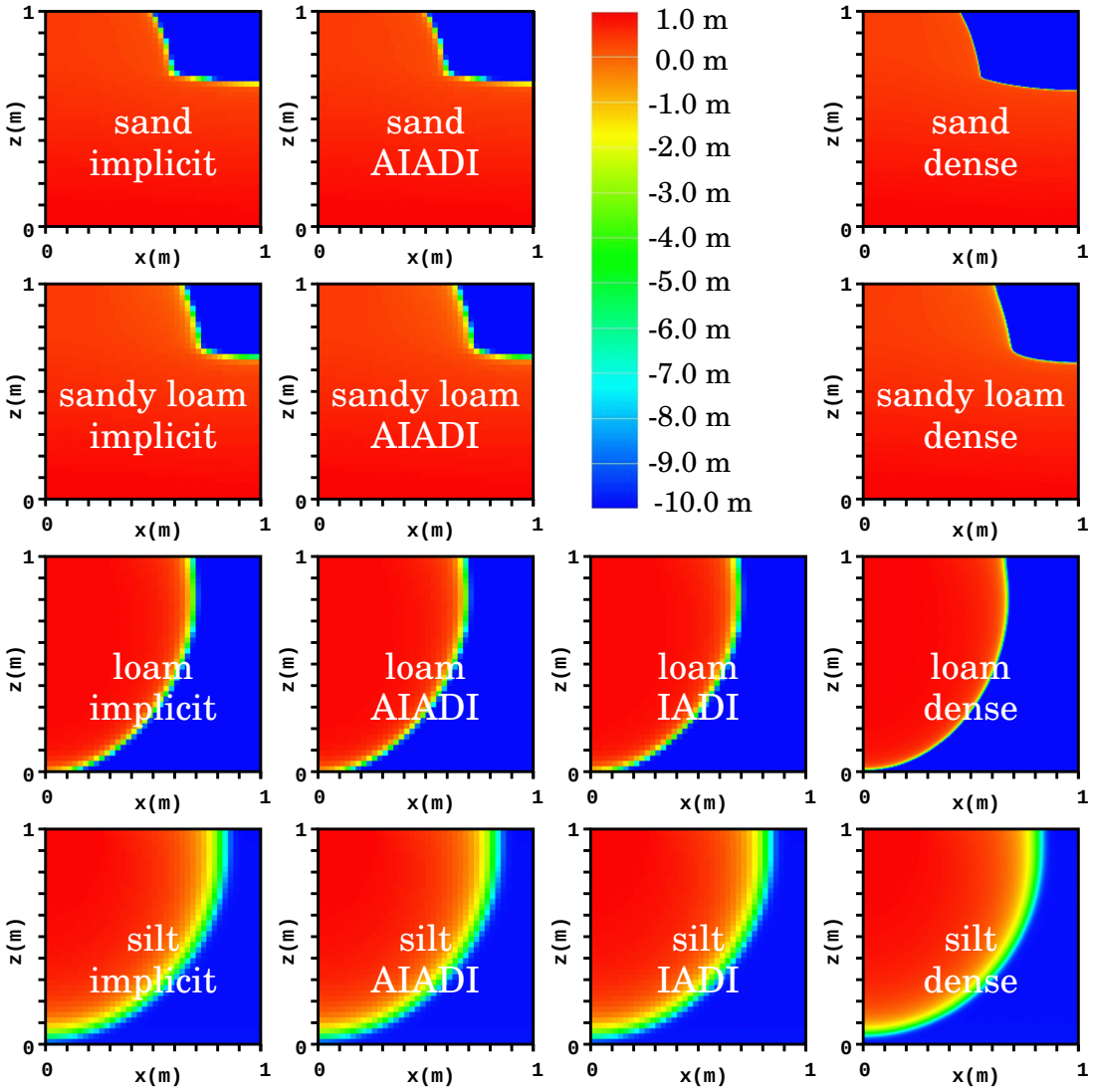

FIGURE 2.1: Test simulation 1, ψ at the end of simulation.

and at 37 h for sandy loam. Fig. 2.3 describes the water table position in both cases. It is noted that the time when the oscillation begins corresponds to the time when all bottom grids ($z = 0$) become saturated. This implies that the IADI scheme might result in a numerical instability during the simulation of a saturated flow.

Table 2.2 shows the performance of each of the schemes. The AIADI scheme was faster than the implicit scheme in all four cases. All three schemes had the same order of relative error (RE). In the case of sand and sandy loam, the values of RE were comparatively high because the sharp infiltration front led to a high relative error with low-resolution grids. The IADI scheme shows a poor mass balance conservativity as compared to the other two schemes while using the pressure-head-based Richards equation. The mass balance error (MBE) of the AIADI scheme was several orders higher than that

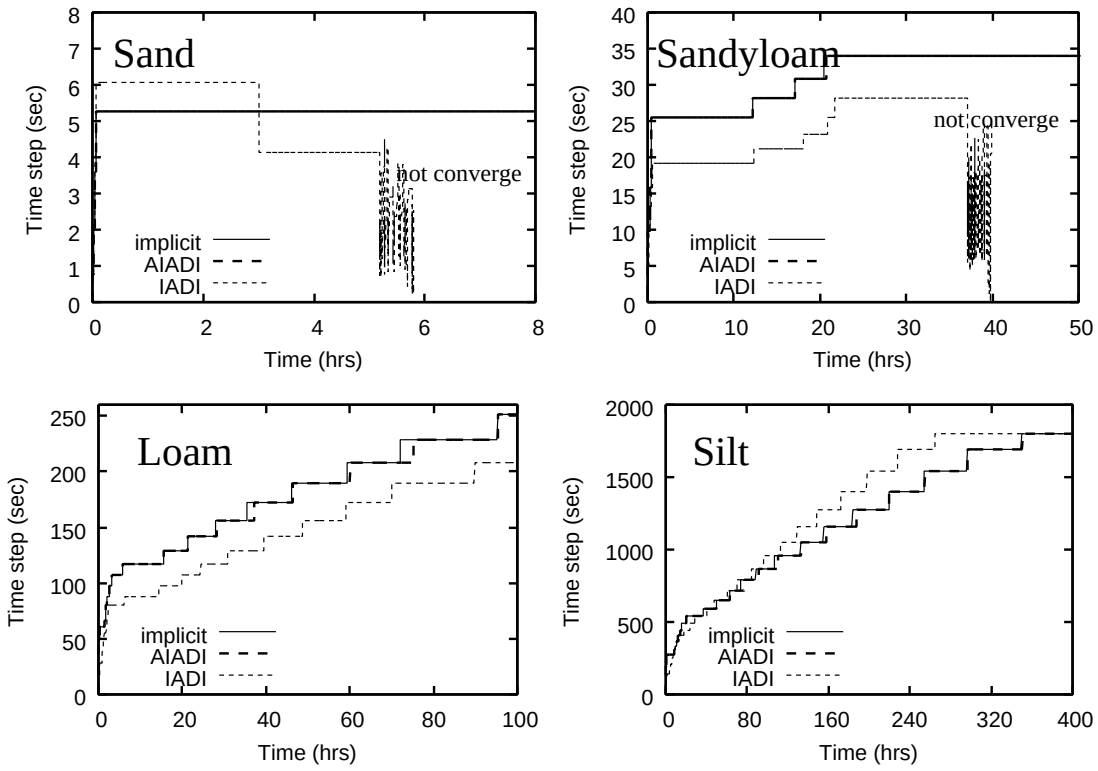

FIGURE 2.2: Test simulation 1, time step

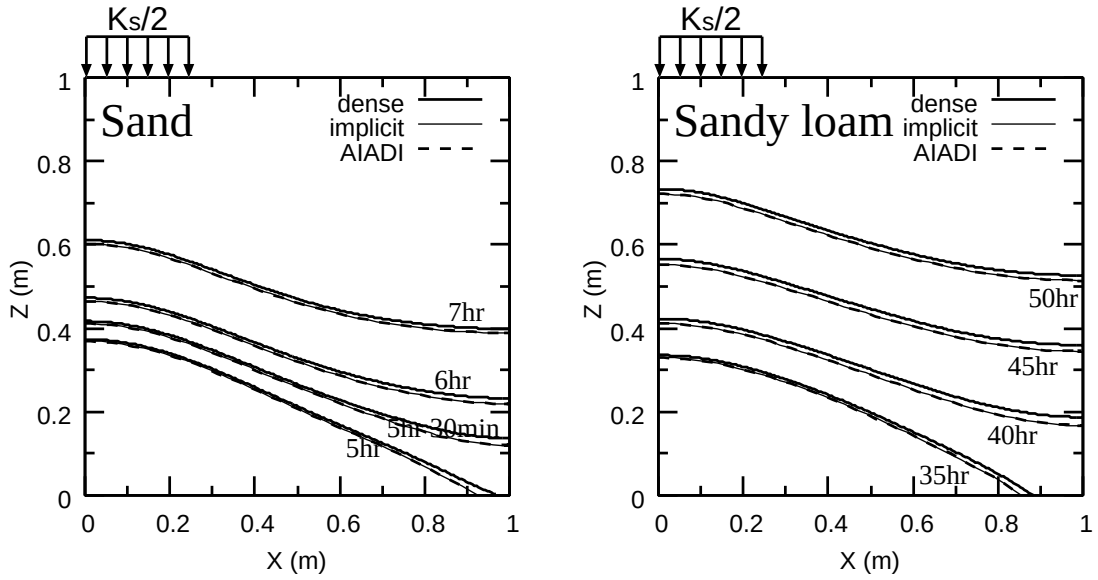

FIGURE 2.3: Test simulation 1, water table positions in the cases of sand and sandy loam

TABLE 2.2: Test simulation 1, performances of the schemes.

Soil type (period)[a]	Scheme	CPU (sec)	Nb.Iter[b]	MBE[c] (%)	RE[c] (%)
sand	implicit	532	22894	1.85e-08	38.13
(8 hrs)	AIADI	184	22948	2.17e-04	38.14
sandy loam	implicit	508	24073	6.28e-09	31.87
(50 hrs)	AIADI	178	24076	2.16e-04	31.89
loam	implicit	115	8797	2.61e-08	12.21
(100 hrs)	AIADI	52	8848	7.71e-08	12.21
	IADI	77	11218	0.68	10.68
silt	implicit	87	5820	1.02e-08	3.60
(400 hrs)	AIADI	43	5857	7.96e-07	3.60
	IADI	40	5606	0.61	2.64

[a] (period) is the simulation period.
[b] Nb.Iter is the total number of iteration.
[c] MBE and RE are evaluated at the end of simulation.

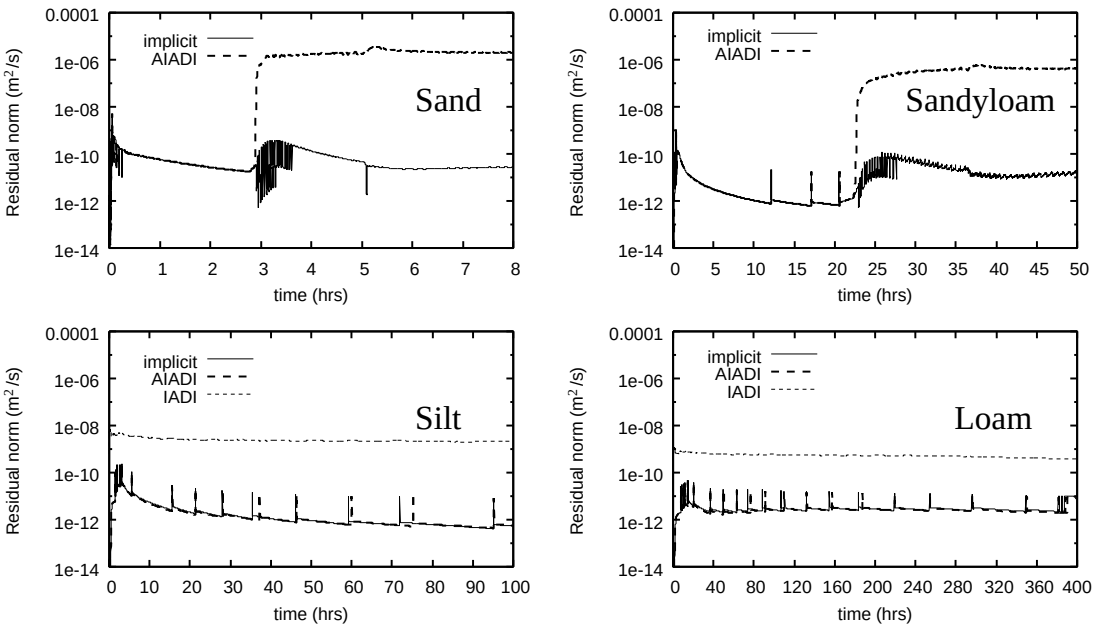

FIGURE 2.4: Test simulation 1, residual norm at each calculation time step.

of the implicit scheme for the saturated–unsaturated flow cases, sand and sandy loam, while these two schemes are based on the same mixed form of the Richards equation. In order to investigate the reason for the above observation, the residual norm at each calculation time step was evaluated, as shown in Fig. 2.4. In the case of the unsaturated flow, the implicit scheme and the AIADI scheme converged within the same order of the residual norm. However, the AIADI scheme converged under a several-order higher level of the residual norm than the implicit scheme in the case of a saturated–unsaturated flow. The appearance of the differences in the residual norm was simultaneous with the appearance of the saturated flow. It is supposed that the added terms for applying ADI technique to the Richards equation caused a comparatively larger residual norm, which resulted in a larger MBE in the saturated zone because the value of added terms was proportional to the value of hydraulic conductivity.

2.3.2 Test 2: Two-dimensional transient variably saturated flow

In order to investigate the performance for a saturated flow, the experiment conducted by *Vauclin et al.* [92] was selected for the second test simulation. The same example was also used by *Clement et al.* [15] to verify their two-dimensional variably saturated model. The model of *Clement et al.* [15] and the implicit scheme discussed in this study represent the same algorithm. The flow domain consisted of a rectangular soil domain, 6.0 m $\times$ 2.0 m, with an initial horizontal water table located at a height of 0.65 m. At the soil surface, a constant flux of 148 mm/h was applied over a width of 1.0 m at the center. The remaining soil surface was covered in order to prevent evaporation losses. Because of the symmetry, it was only necessary to model the right-hand side of the flow domain, as shown in Fig. 2.5. The modeled portion of the flow domain was 3.0 m $\times$ 2.0 m, with no flux on the bottom and the left-side boundaries.

The soil properties from *Vauclin et al.* [92] are as follows: $\theta_s = 0.3$, $\theta_r = 0.01$, and $K_s = 9.72 \times 10^{-5}$ m/s. The van Genuchten model was fitted to the water retention and the relative hydraulic conductivity data given by *Vauclin et al.* [92]. The soil properties, $\alpha = 3.3$ m^{-1}, $n = 4.1$, estimated by *Clement et al.* [15], were used along with $\Delta x = 10$ cm and $\Delta z = 5$ cm. The simulation period was 8 h.

Fig. 2.5 describes the water table observed experimentally and calculated by each scheme at 0, 2, 3, 4, and 8 h. The results simulated using each of the three schemes

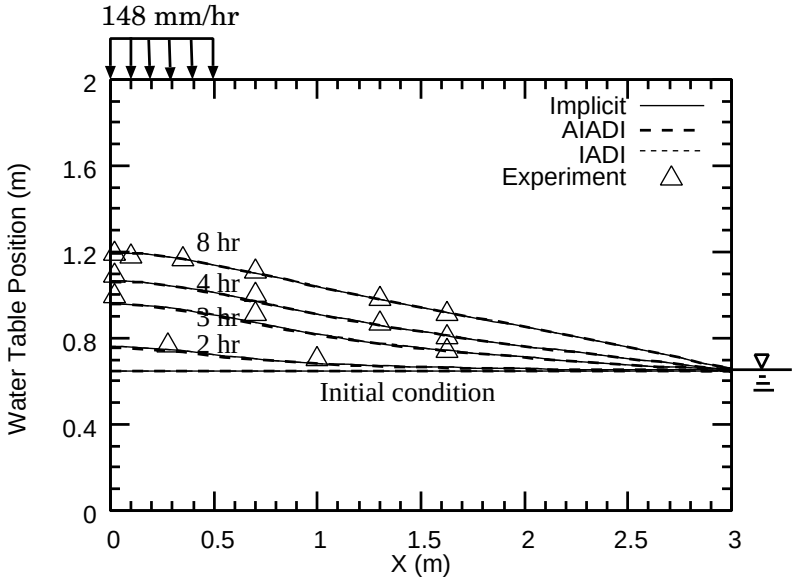

FIGURE 2.5: Water-table mounding data collected by *Vauclin et al.* [92] and simulation.

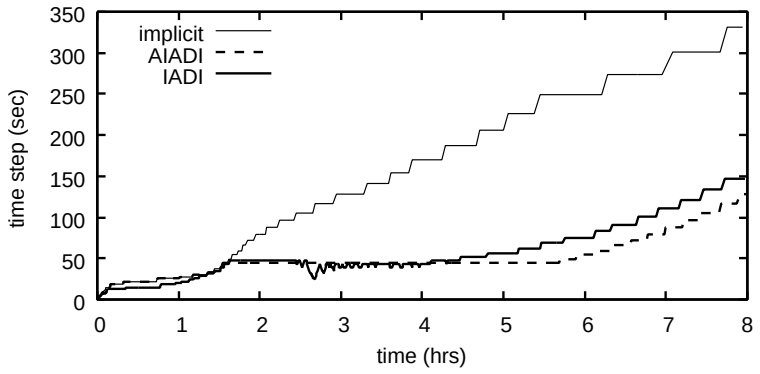

FIGURE 2.6: Test simulation 2, time step

closely agree with the experimentally observed values by *Vauclin et al.* [92] and were themselves very similar to one another. Fig. 2.6 describes the calculation time step. A small oscillation in the time step was observed in the case of the IADI scheme at around 3 h.

Table 2.3 shows the performance of the schemes. The AIADI was faster than the implicit scheme when it used a smaller time step and had more iteration steps. The AIADI generated several-order higher MBE than the implicit scheme even though this MBE was of a lower order than that of the IADI. This is coincident with the result of Test 1, in which the AIADI scheme causes a comparatively larger MBE than the implicit scheme for a saturated flow in the sand and sandy loam cases.

TABLE 2.3: Test simulation 2, performances of the schemes.

Scheme	CPU (sec)	Nb.Iter[a]	MBE[b] (%)	RE[b] (%)
implicit	21	1520	2.24e-05	14.83
AIADI	10	3043	5.23e-02	14.80
IADI	12	3246	0.53	14.82

[a] Nb.Iter is the total number of iteration.
[b] MBE and RE are evaluated at the end of simulation.

2.3.3 Test 3: Two-dimensional simulation for rainfall-runoff on a simple slope

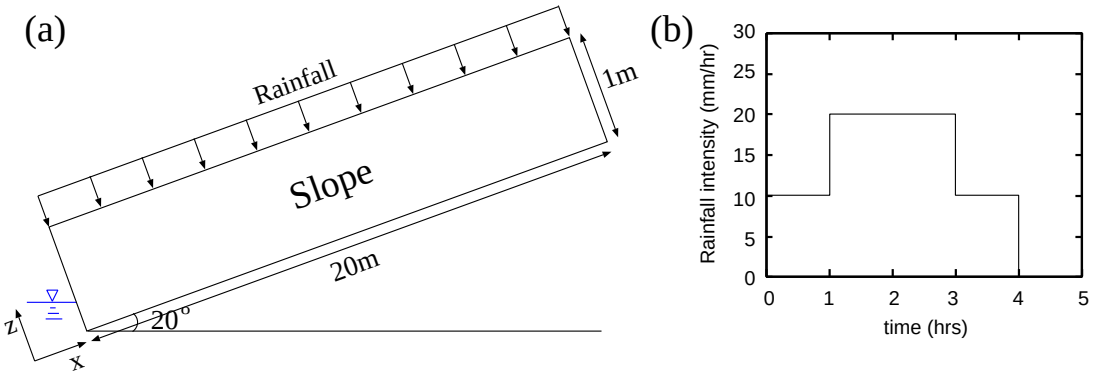

FIGURE 2.7: Test simulation 3, (a) slope and (b) rainfall intensity

The simultaneous linear system solvers have a computational complexity of CN^a, where C and a are unknown constants and N is the number of unknowns in the linear system. The constant a of the AIADI scheme is expected to be very close to 1. In order to estimate the values of a of two schemes, the implicit and AIADI schemes, a series of grid refinement experiments was conducted. The two-dimensional rainfall-runoff simulation for a simple slope is considered. A 1-m-thick, 20-m-long slope with a 20° incline and a sandy loam soil texture was considered, as shown in Fig. 2.7-(a). A constant water level (30 cm) was maintained at the lower end for the sake of simplicity. The lower and upper sides were considered to be no-flux boundaries. The surface had a rainfall-flux boundary condition, and the rainfall intensity is as shown in Fig. 2.7-(b). In the case of an inclined slope, we used the modified coordinates, as shown in Fig. 2.7-(a), and the corresponding equation can be given as

$$\frac{\partial \theta}{\partial t} = \nabla \cdot K(\psi)\nabla\psi + \sin w \frac{\partial K(\psi)}{\partial x} + \cos w \frac{\partial K(\psi)}{\partial z}, \tag{2.27}$$

where w is 20°. The initial condition was $\psi(x, z, 0) + x \sin w + z \cos w = 30$cm. The simulation period was two weeks. Six levels of grids were used: 100×20, 100×40, 200×40, 200×80, 400×80, and 400×160 grids.

TABLE 2.4: Test simulation 3, performances of the schemes.

Grid	Scheme	CPU (sec)	Nb.Iter[a]	MBE[b] (%)	RE[b] (%)
100×20	implicit	23	3789	2.23e-08	40.96
	AIADI	18	3832	8.63e-05	41.51
100×40	implicit	98	7071	1.58e-08	33.46
	AIADI	71	7071	3.68e-05	33.46
200×40	implicit	303	7739	1.37e-08	22.43
	AIADI	162	7741	1.87e-05	22.43
200×80	implicit	1716	14985	4.01e-09	17.06
	AIADI	572	15069	5.90e-05	16.24
400×80	implicit	3761	15979	3.83e-09	6.75
	AIADI	1252	16051	2.87e-04	5.53
400×160	implicit	21949	31900	5.99e-06	-
	AIADI	5579	31935	1.10e-04	-

[a] Nb.Iter is the total number of iteration.
[b] MBE and RE are evaluated at the end of simulation.

Fig. 2.8 describes the discharge from the lower end of the slope. It is confirmed that the AIADI and implicit schemes give a similar result. Table 2.4 shows the performances of the schemes. The solution of the finest-resolution grid was considered a surrogate for the exact solution in order to estimate the RE. The AIADI scheme was faster than the implicit scheme and had a several-order higher MBE than the implicit scheme as shown by Tests 1 and 2 while generating the same order of RE. Fig. 2.9 shows the normalized CPU time divided by the number of cells and the total number of iterations. The value of a of the AIADI scheme was 1.04, and that of the implicit scheme was 1.36; these values agree with the expected values. Therefore, it can be concluded that the relative efficiency of the AIADI scheme increases with an increase in the number of unknowns.

2.3.4 Test 4: Three-dimensional infiltration into layered soil

The ability of simulating three-dimensional problems and flow for layered soil properties were tested. A 1-m^3 soil domain was considered, as shown in Fig. 2.10. The surface was considered to be under a constant infiltration (= 5 mm/h) condition, and the other boundaries had no-flux boundary conditions. The initial pressure head was taken to be

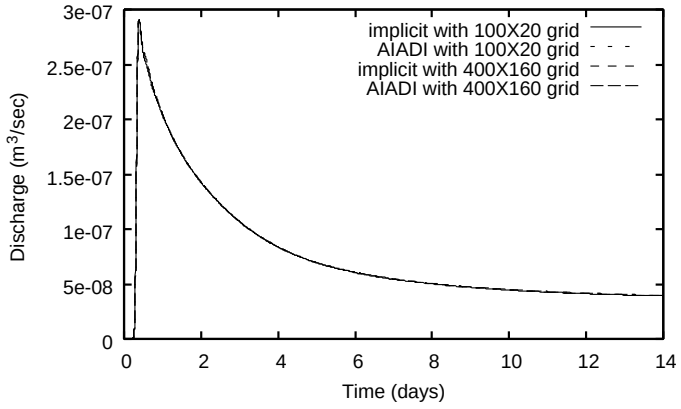

FIGURE 2.8: Test simulation 3, discharge at the end of the slope.

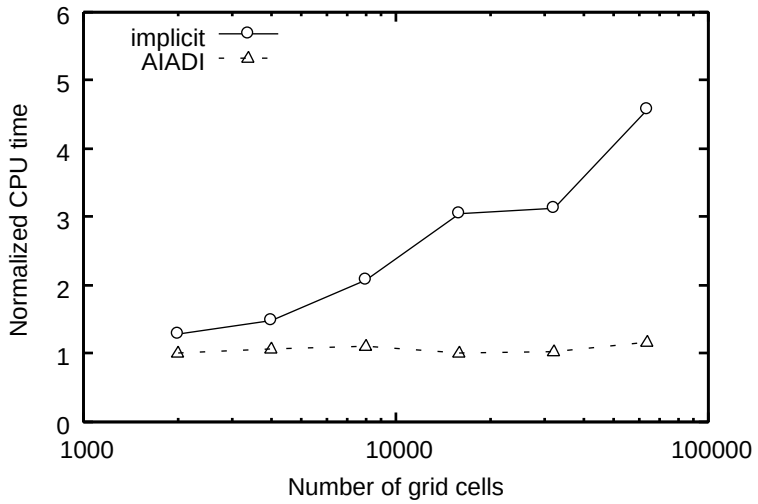

FIGURE 2.9: Test simulation 3, normalized CPU time per iteration.

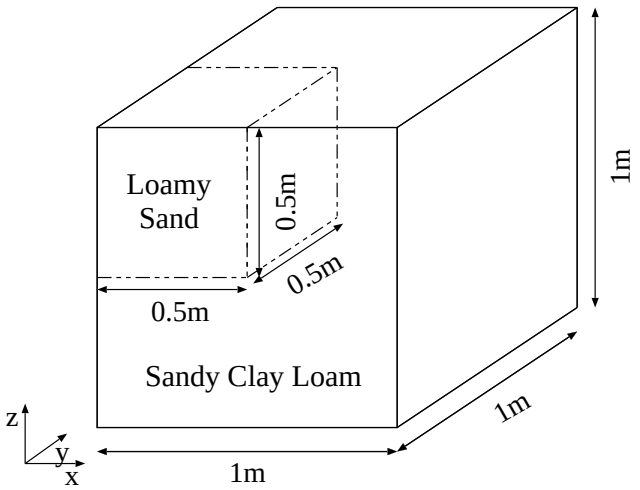

FIGURE 2.10: Layered soil domain for test simulation 4.

$\psi(x, y, z, 0) = -2.0$ m. The time step was $1 \leq \Delta t \leq 3600$ s, and the simulation period was 1 day. Grid refinement was also applied in this test; however, only the z direction refinement was conducted because this test is similar to the one-dimensional infiltration problem of the z-direction. Six levels of grids were used: 20, 40, 80, 160, 320, and 640 z-dimension grids; all the grids had x and y dimensions of 20×20.

The three-dimensional AIADI equation consists of three passes as follows:

$$\begin{aligned} & C_{i,j,k}^{n+1,3m} \frac{\psi_{i,j,k}^{n+1,3m+1} - \psi_{i,j,k}^{n+1,3m}}{\Delta t} + \frac{\theta_{i,j,k}^{n+1,3m} - \theta_{i,j,k}^{n}}{\Delta t} \\ & + I_m \overline{K}_{i,j,k}^{n+1,3m} (\psi_{i,j,k}^{n+1,3m+1} - \psi_{i,j,k}^{n+1,3m}) \\ = \; & \Delta_x (K^{n+1,3m} \Delta_x \psi^{n+1,3m+1}) + \Delta_y (K^{n+1,3m} \Delta_y \psi^{n+1,3m}) \\ & + \Delta_z (K^{n+1,3m} \Delta_z \psi^{n+1,3m}) + \Delta_z (K^{n+1,3m}), \end{aligned} \tag{2.28}$$

$$\begin{aligned} & C_{i,j,k}^{n+1,3m} \frac{\psi_{i,j,k}^{n+1,3m+2} - \psi_{i,j,k}^{n+1,3m+1}}{\Delta t} + I_m \overline{K}_{i,j,k}^{n+1,3m} (\psi_{i,j,k}^{n+1,3m+2} - \psi_{i,j,k}^{n+1,3m+1}) \\ = \; & \Delta_y (K^{n+1,3m} \Delta_y \psi^{n+1,3m+2}) - \Delta_y (K^{n+1,3m} \Delta_y \psi^{n+1,3m}), \end{aligned} \tag{2.29}$$

$$\begin{aligned} & C_{i,j,k}^{n+1,3m} \frac{\psi_{i,j,k}^{n+1,3m+3} - \psi_{i,j,k}^{n+1,3m+2}}{\Delta t} + I_m \overline{K}_{i,j,k}^{n+1,3m} (\psi_{i,j,k}^{n+1,3m+3} - \psi_{i,j,k}^{n+1,3m+2}) \\ = \; & \Delta_z (K^{n+1,3m} \Delta_z \psi^{n+1,3m+3}) - \Delta_z (K^{n+1,3m} \Delta_z \psi^{n+1,3m}), \end{aligned} \tag{2.30}$$

$$\begin{aligned} \overline{K}_{i,j,k}^{n+1,3m} = \; & K_{i+1/2,j,k}^{n+1,3m} + K_{i-1/2,j,k}^{n+1,3m} + K_{i,j+1/2,k}^{n+1,3m} + K_{i,j-1/2,k}^{n+1,3m} \\ & + K_{i,j,k+1/2}^{n+1,3m} + K_{i,j,k-1/2}^{n+1,3m}, \end{aligned} \tag{2.31}$$

where subscripts i, j, and k denote the spatial coordinates of the point in the x, y, and z axes, respectively. The three-dimensional implicit scheme linearized by the Picard method is given as

$$\begin{aligned} & C_{i,j,k}^{n+1,m} \frac{\psi_{i,j,k}^{n+1,m+1} - \psi_{i,j,k}^{n+1,m}}{\Delta t} + \frac{\theta_{i,j,k}^{n+1,m} - \theta_{i,j,k}^{n}}{\Delta t} \\ = \; & \Delta_x (K^{n+1,m} \Delta_x \psi^{n+1,m+1}) + \Delta_y (K^{n+1,m} \Delta_y \psi^{n+1,m+1}) \\ & + \Delta_z (K^{n+1,m} \Delta_z \psi^{n+1,m+1}) + \Delta_z (K^{n+1,m}). \end{aligned} \tag{2.32}$$

In our experience, the traditional IADI scheme does not successfully converge in the case

of several three-dimensional problems, e.g., a simple infiltration problem or a rainfall-runoff problem with a very low infiltration intensity and a small time step. This is because the Peaceman-Rachford ADI method, which is the fundamental algorithm for the traditional IADI scheme, is conditionally stable for three-dimensional problems and unconditionally stable for two-dimensional problems.

TABLE 2.5: Test simulation 4, performances of the schemes.

Grid (20×20)	Scheme	CPU (sec)	Nb.Iter[a]	MBE[b] (%)	RE[b] (%)
×20	implicit	45	1719	6.64e-09	19.42
	AIADI	33	1718	1.67e-08	19.42
×40	implicit	161	2830	3.92e-09	11.62
	AIADI	101	2830	1.35e-08	11.62
×80	implicit	703	4959	2.50e-09	6.77
	AIADI	389	4959	6.15e-09	6.77
×160	implicit	3125	9538	1.06e-09	3.18
	AIADI	1498	9538	1.60e-09	3.18
×320	implicit	13990	18027	4.60e-10	1.01
	AIADI	5579	18158	5.49e-10	1.00
×640	implicit	75419	38999	2.01e-10	-
	AIADI	21366	35137	1.94e-10	-

[a] Nb.Iter is the total number of iteration.
[b] MBE and RE are evaluated at the end of simulation.

Fig. 2.11 describes the ψ calculated using the AIADI scheme with a 20×20×160 grid. A saturated flow did not appear in this test. Table 2.5 shows the performances of the schemes. The AIADI scheme performed faster than the implicit scheme in this three-dimensional test case. Fig. 2.12 shows the normalized CPU time. The value of a of the AIADI scheme was almost 1, and that of the implicit scheme was 1.24 in this test. Furthermore, unlike in the case of a saturated flow such as Test 1∼3, the AIADI scheme generated the same order of MBE for an unsaturated flow. For confirming that both schemes have the same accuracy for an unsaturated flow, the residual norm was evaluated and is shown in Fig. 2.13. Both schemes converged within the same order of residual norm at each calculation time step.

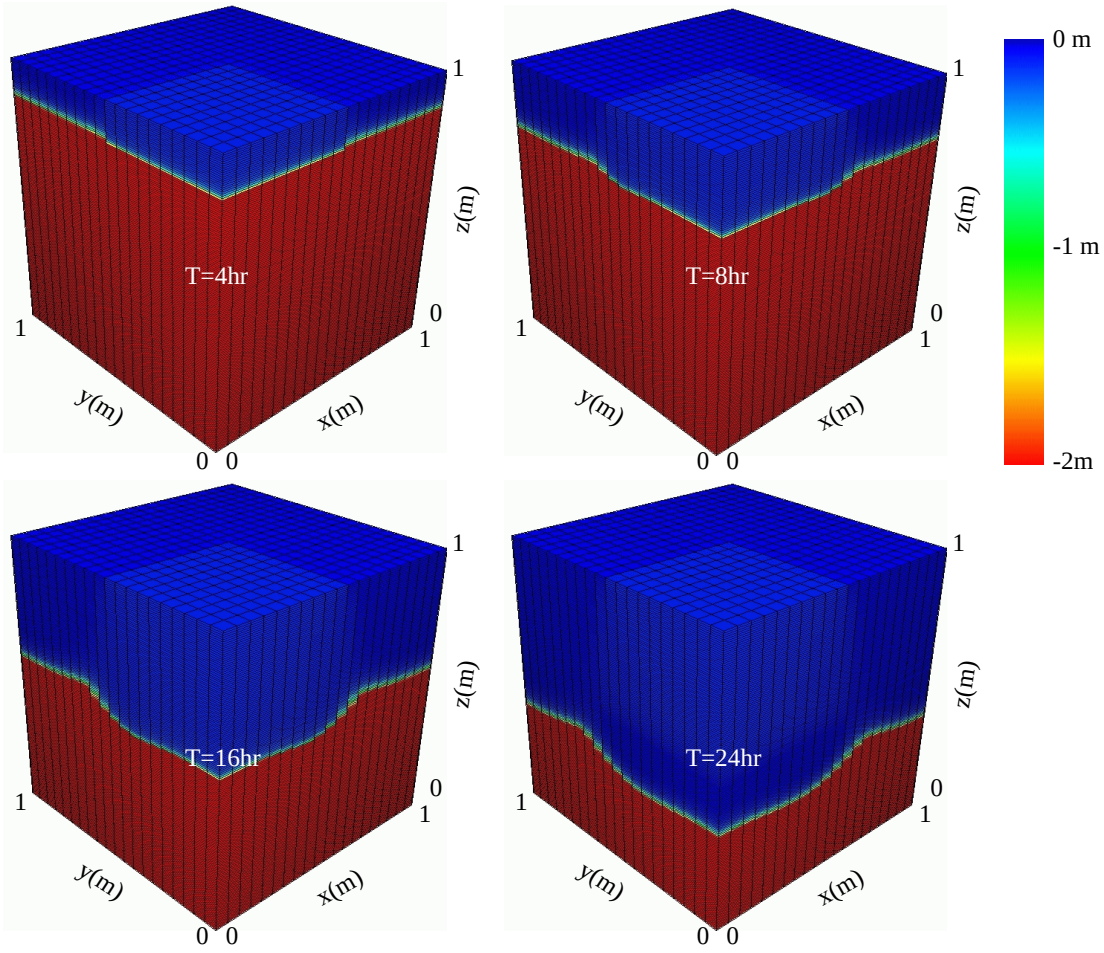

FIGURE 2.11: Test simulation 4, ψ of vertical cross section for $y = 0$ calculated by the AIADI scheme with 20×20×160 grid

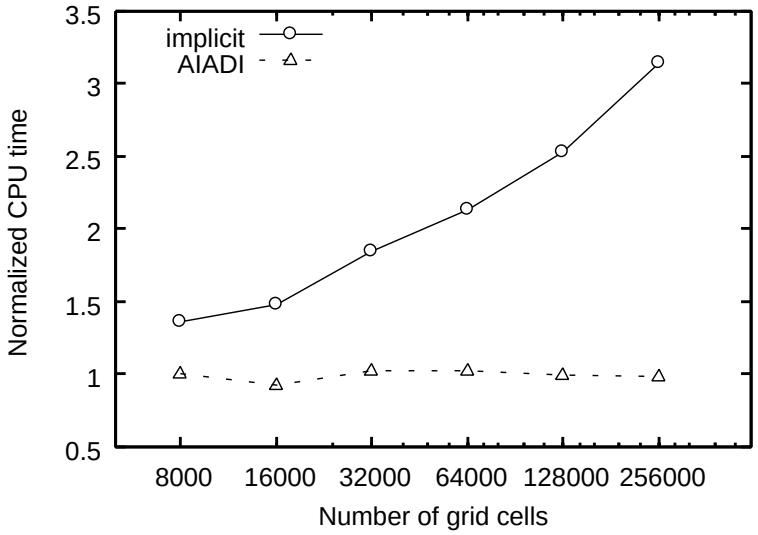

FIGURE 2.12: Test simulation 4, normalized CPU time per iteration.

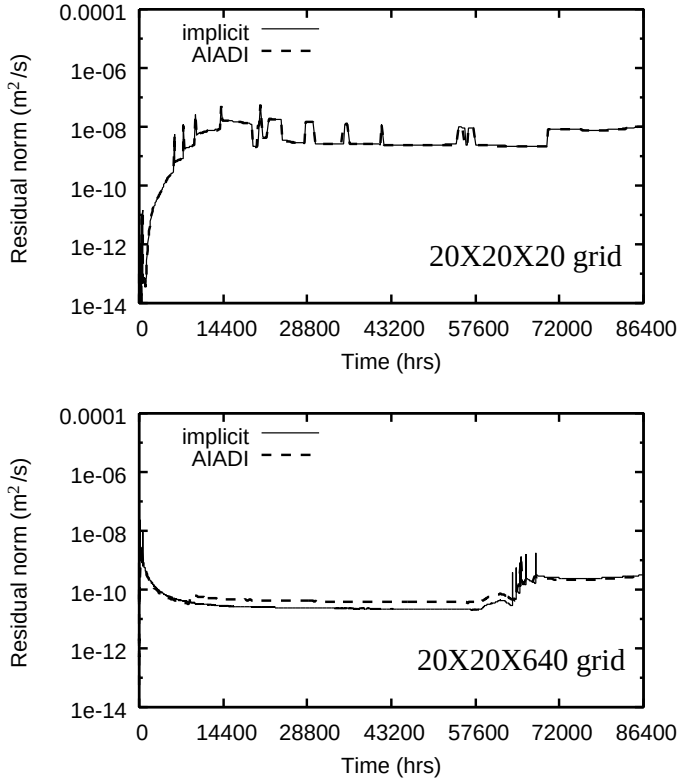

FIGURE 2.13: Test simulation 4, residual norm at each calculation time step.

2.4 Summary

The performance of a new IADI algorithm for two and three-dimensional saturated–unsaturated flows was evaluated. The proposed scheme is based on the Douglas-Rachford ADI method and is a perturbation form of the backward Euler difference equation linearized by the Picard method. The proposed scheme is mathematically clear and has certain advantages over the conventional IADI scheme in terms of applicability to three-dimensional problems. In order to eliminate the mass balance problem, a mixed form of the Richards equation was used. Because only tridiagonal linear systems are involved, the proposed scheme is simpler to implement and has a lower computational cost per iteration than the preconditioned Krylov solver. Further, the computational cost of the proposed scheme is expected to increase linearly when the problem dimension increases. Furthermore, parallelization for the proposed scheme is also very simple using a parallelization framework such as OpenMP. In the computation of a direction pass, each set of simultaneous equations can be solved separately because each set of simultaneous equations is not involved in the set of equations in one pass.

The performance of the proposed scheme was evaluated by carrying out four test simulations. In some test cases, the conventional IADI scheme exhibited numerical instability or divergence, but the proposed scheme exhibited a stable behavior under the same conditions. The proposed scheme could simulate a three-dimensional flow, whereas the conventional IADI scheme could simulate only a two-dimensional flow. This is because the proposed scheme is derived from the Douglas-Rachford ADI scheme, which is unconditionally stable for two- and three-dimensional parabolic partial differential equations while the Peaceman-Rachford ADI scheme is unconditionally stable only for two-dimensional equations that the conventinal IADI scheme is based upon.

The proposed scheme was faster than the Euler backward implicit scheme linearized by the Picard iteration method in all cases in which the linear systems were solved by the SSOR-preconditioned BICGSTAB solver. As expected, the computational cost of the proposed scheme was proportional to the number of unknowns, as shown in Tests 3 and 4. However, in the saturated zone, the proposed scheme had comparatively lower numerical accuracy than the implicit scheme even though the proposed scheme converged within the same order of residual norm as that of the implicit scheme for unsaturated flows and had the same order of MBE, as shown in Tests 1 and 4. The added terms of the proposed scheme may cause the comparatively larger residual norm in the case of a saturated flow. There is a trade-off between the implicit scheme conserving the accuracy of the residual norm and the AIADI scheme consuming less CPU resources and having a relatively greater ease of implementation.

However, the proposed scheme was applied only to relatively simple geometries that can be described by orthogonal grids in this chapter. This issue will be discussed in Chapter 3.

Chapter 3

Three-dimensional saturated–unsaturated flow modeling with non-orthogonal grids

3.1 Introduction

If the domain shape is curvilinear, the FDM becomes inefficient as compared to the other models that can treat non-orthogonal grid because higher-resolution grid is required to represent complex flow domain using orthogonal grid. Further, the principle axes of anisotropy should be aligned with the orthogonal axes in FDMs.

To overcome this disadvantage of the conventinoal finite-difference model, we applied a coordinate transformation method to a saturated–unsaturated flow equation. Because saturated–unsaturated flow equation systems are highly nonlinear, implicit temporal discretization and iterative procedures are generally needed for numerical stability. Coordinate transfomation represents diffusion with cross-derivative terms. Therefore, the transformed equation requires a 19-point stencil instead of a 7-point stencil, which is required by a conventional FDM with an orthogonal grid. The complexity of the algorithm and the additional storage makes the scheme less attractive. Furthermore, implicit evaluation of all the terms can be inefficient for quasi-orthogonal grids, because

the cross-derivative terms make minor contributions to the overall system. Therefore, in this Chapter we implement an approach in which the cross-derivative terms are evaluated at the previous iterative level, and the other terms are evaluated at the current iterative level. In this approach, the 7-point stencil is calculated implicitly by the iterative calculation, as in the case of the conventional FDM. In this way, the proposed scheme can treat curvilinear coordinate systems while retaining computational efficiency and simplicity of discretization. The implementation of 19-stencil in three-dimensional problems will be discussed in Chapter 4.

This chapter is organized as follows. In Section 3.2, the governing equation for saturated–unsaturated flow in porous media is presented. This physical-space equation is then transformed into a computational-space equation. In Section 3.3, spatial and temporal discretization and a method for evaluating cross-derivative terms and metrics are described. In Section 3.4, four test simulations are carried out to verify the performance and accuracy of the proposed model. In the first test simulation, a steady-state problem is considered for studying the applicability of the proposed model with a highly skewed grid. In the second test simulation, an infiltration problem is considered for evaluating the effect of mesh skewness on the performance and accuracy of the proposed model. The third test simulation is a transient, variably saturated flow simulation that is carried out to demonstrate the performance of the proposed model with a non-rectangular flow domain. The fourth test simulation is a three-dimensional rainfall-runoff simulation that is carried out to show the performance of the proposed model for a three-dimensional curvilinear-shaped flow domain. Results of these simulations are compared with a numerical solution of both the conventional FDM that utilizes a high-resolution, stepwise orthogonal grid, and a HYDRUS [81], which is widely used commercial FEM.

3.2 Coodinate Transformation

Coordinate transformation is a common technique used in CFD for the general Navier-Stokes equation. This technique transforms a curvilinear grid into a rectangular grid, as shown in Fig. 3.1. The curvilinear coordinate system in physical space, $(x_1, x_2, x_3) = (x, y, z)$, can be transformed into a new coordinate system in computational space, $(\xi_1,$

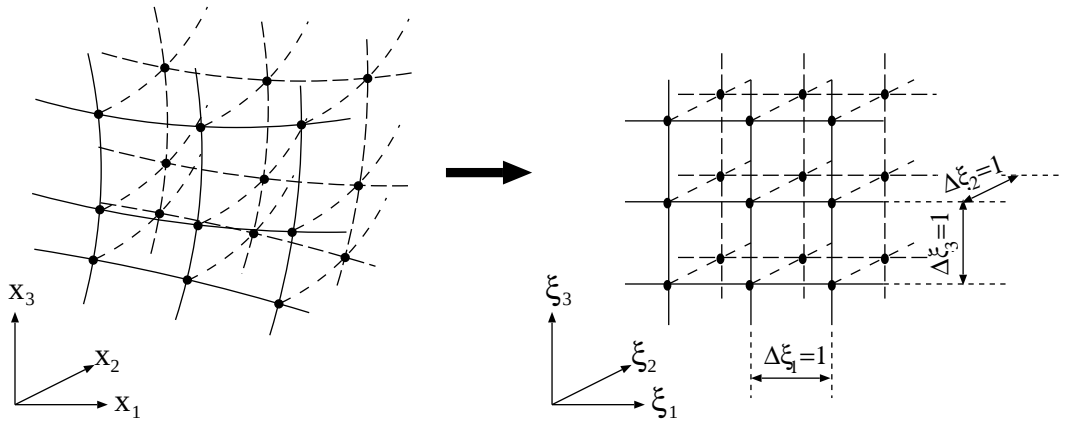

FIGURE 3.1: Concept of coordinate transformation: an arbitrarily shaped mesh in physical space is transformed into an orthogonal mesh in computational space.

ξ_2, ξ_3)=(ξ, η, ζ). The summation form of Richards' equation, Eq. (2.1), is expressed as

$$\frac{\partial\theta}{\partial t} - \sum_{r=1}^{3}\sum_{s=1}^{3}\frac{\partial}{\partial x_r}\left\{KK^e_{r,s}\frac{\partial\psi}{\partial x_s}\right\} - \sum_{r=1}^{3}\frac{\partial KK^e_{r,3}}{\partial x_3} = 0, \tag{3.1}$$

where K is an unsaturated hyraulic conductivity function of ψ, $K^e_{r,s}$ is anisotropy tensor element of the hydraulic conductivity. If the diagonal entries of $K^e_{r,s}$ equal one and off-diagonal entries zero, it express an isotropic medium.

A chain rule operation was performed to transform a partial derivative of a function f in physical space (x_1, x_2, x_3) into computational space (ξ_1, ξ_2, ξ_3) as follows:

$$\frac{\partial}{\partial x_r} = \sum_{p=1}^{3}\frac{\partial\xi_p}{\partial x_r}\frac{\partial}{\partial\xi_p} \tag{3.2}$$

or

$$\frac{\partial}{\partial x_r} = \sum_{p=1}^{3} J\frac{\partial}{\partial\xi_p}\left(\frac{1}{J}\frac{\partial\xi_p}{\partial x_r}\right). \tag{3.3}$$

Applying Eq. (3.3) to Eq. (3.1) gives

$$\frac{\partial\theta}{\partial t} - \sum_{r=1}^{3}\sum_{r=s}^{3}\sum_{p=1}^{3} J\frac{\partial}{\partial\xi_p}\left\{\frac{KK^e_{r,s}}{J}\frac{\partial\xi_p}{\partial x_r}\frac{\partial\psi}{\partial x_s}\right\} - \sum_{r=1}^{3}\sum_{p=1}^{3} J\frac{\partial}{\partial\xi_p}\left(\frac{KK^e_{r,3}}{J}\frac{\partial\xi_p}{\partial x_3}\right) = 0. \tag{3.4}$$

Applying Eq. (3.2) to $\partial\psi/\partial x_r$ of Eq. (3.4) gives

$$\frac{\partial\theta}{\partial t} - \sum_{r=1}^{3}\sum_{s=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3} J\frac{\partial}{\partial\xi_p}\left\{\frac{KK_{r,s}^{e}}{J}\frac{\partial\xi_p}{\partial x_r}\frac{\partial\xi_q}{\partial x_s}\frac{\partial\psi}{\partial\xi_q}\right\} - \sum_{r=1}^{3}\sum_{p=1}^{3} J\frac{\partial}{\partial\xi_p}\left(\frac{KK_{r,3}^{e}}{J}\frac{\partial\xi_p}{\partial x_3}\right) = 0. \tag{3.5}$$

This equation can be rewritten as

$$\frac{1}{J}\frac{\partial\theta}{\partial t} - \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K\frac{\partial\psi}{\partial\xi_q}\right\} - \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^{p}K\right) = 0, \tag{3.6}$$

where

$$J = \frac{\partial(\xi_1,\xi_2,\xi_3)}{\partial(x_1,x_2,x_3)}, \qquad G^{p,q} = \sum_{r=1}^{3}\sum_{s=1}^{3}\frac{K_{r,s}^{e}}{J}\frac{\partial\xi_p}{\partial x_r}\frac{\partial\xi_q}{\partial x_s}, \qquad H^{p} = \sum_{r=1}^{3}\frac{K_{r,3}^{e}}{J}\frac{\partial\xi_p}{\partial x_3}, \tag{3.7}$$

where J is the Jacobian determinant, meaning the ratio of the control volume in the physical space to that in the computational space, and $G^{p,q}$ and H^{p} represent the mesh skewness and the anisotropy tensor. The terms with $p = q$ of Eq. (3.6) represent normal-derivative contributions and the other terms ($p \neq q$) represent cross-derivative contributions. Detailed evaluation of J and the metrics, $\partial\xi_p/\partial x_r$, will be shown in Section 3.3.2. Eq. (3.6) is mass conservative because the mesh skewness tensors lie inside the differential operators [44, 45, 60].

3.3 Numerical modeling

3.3.1 Temporal discretization

We implemented the backward Euler scheme and Picard iteration method in this chapter because of the reason mentioned in Section 2.2.1 As the same way in Section 2.2.1, the backward Euler approximation and Picard linearization of Eq. (3.3) is expressed as

$$\begin{aligned}&\frac{C^{n+1,m}}{J}\frac{\psi^{n+1,m+1}-\psi^{n+1,m}}{\Delta t} + \frac{1}{J}\frac{\theta^{n+1,m}-\theta^{n}}{\Delta t}\\ =\ &\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m+1}\right\} + \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^{p}K^{n+1,m}\right),\end{aligned} \tag{3.8}$$

where superscripts n and m denote the time level and iteration level, respectively, and $C(= d\theta/d\psi)$ is the specific moisture capacity function.

3.3.2 Finite-difference discretization

Because the transformed equation gives the cross-derivative terms as in Eq. (3.8), a 19-point stencil is required to solve the equation fully implicitly. Considering all the terms implicitly is often inefficient because the cross-derivative terms usually make minor contributions to the overall system. Moreover, it is difficult to code a 19-point-stencil simultaneous equation. Hence, we propose to evaluate the cross-derivative terms at the previous iteration level. The cross-derivative terms are solved implicitly in the temporal discretization and explicitly in the spatial discretization.

In the computational space, the grid is orthogonal and all grid sizes ($= \Delta\xi_p$) are practically set to 1. The grid sizes do not virtually have a consequence on the result. Any computational grid dimension can be used. However, using a dimension other than 1 may introduce a round-off error or slow the code down, and dividing by 1 is very convenient. Hence, grid sizes are usually set to 1. A finite-difference approximation of Eq. (3.8) can be expressed as

$$\begin{aligned} &\frac{C_{i,j,k}^{n+1,m}}{J}\frac{\psi_{i,j,k}^{n+1,m+1}-\psi_{i,j,k}^{n+1,m}}{\Delta t}+\frac{1}{J}\frac{\theta_{i,j,k}^{n+1,m}-\theta_{i,j,k}^{n}}{\Delta t} \\ =\ &\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m+1}\right\}_{i,j,k}+\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^{p}K^{n+1,m}\right)_{i,j,k}, \end{aligned} \tag{3.9}$$

where the first term on the right-hand side of Eq. (3.9) is evaluated partially implicitly by

$$
\begin{aligned}
& \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m+1}\right\}_{i,j,k} \qquad (3.10)\\
= \;& \Big\{G^{1,1}K^{n+1,m}_{i+1/2,j,k}(\psi^{n+1,m+1}_{i+1,j,k}-\psi^{n+1,m+1}_{i,j,k})\\
& - G^{1,1}K^{n+1,m}_{i-1/2,j,k}(\psi^{n+1,m+1}_{i,j,k}-\psi^{n+1,m+1}_{i-1,j,k})\Big\}\\
+ \;& \Big\{G^{2,2}K^{n+1,m}_{i,j+1/2,k}(\psi^{n+1,m+1}_{i,j+1,k}-\psi^{n+1,m+1}_{i,j,k})\\
& -G^{2,2}K^{n+1,m}_{i,j-1/2,k}(\psi^{n+1,m+1}_{i,j,k}-\psi^{n+1,m+1}_{i,j-1,k})\Big\}\\
+ \;& \Big\{G^{3,3}K^{n+1,m}_{i,j,k+1/2}(\psi^{n+1,m+1}_{i,j,k+1}-\psi^{n+1,m+1}_{i,j,k})\\
& -G^{3,3}K^{n+1,m}_{i,j,k-1/2}(\psi^{n+1,m+1}_{i,j,k}-\psi^{n+1,m+1}_{i,j,k-1})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{1,2}K^{n+1,m}_{i+1/2,j,k}(\psi^{n+1,m}_{i+1/2,j+1,k}-\psi^{n+1,m}_{i+1/2,j-1,k})\\
& - G^{1,2}K^{n+1,m}_{i-1/2,j,k}(\psi^{n+1,m}_{i-1/2,j+1,k}-\psi^{n+1,m}_{i-1/2,j-1,k})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{1,3}K^{n+1,m}_{i+1/2,j,k}(\psi^{n+1,m}_{i+1/2,j,k+1}-\psi^{n+1,m}_{i+1/2,j,k-1})\\
& - G^{1,3}K^{n+1,m}_{i-1/2,j,k}(\psi^{n+1,m}_{i-1/2,j,k+1}-\psi^{n+1,m}_{i-1/2,j,k-1})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{2,1}K^{n+1,m}_{i,j+1/2,k}(\psi^{n+1,m}_{i+1,j+1/2,k}-\psi^{n+1,m}_{i-1,j+1/2,k})\\
& - G^{2,1}K^{n+1,m}_{i,j-1/2,k}(\psi^{n+1,m}_{i+1,j-1/2,k}-\psi^{n+1,m}_{i-1,j-1/2,k})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{2,3}K^{n+1,m}_{i,j+1/2,k}(\psi^{n+1,m}_{i,j+1/2,k+1}-\psi^{n+1,m}_{i,j+1/2,k-1})\\
& - G^{2,3}K^{n+1,m}_{i,j-1/2,k}(\psi^{n+1,m}_{i,j-1/2,k+1}-\psi^{n+1,m}_{i,j-1/2,k-1})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{3,1}K^{n+1,m}_{i,j,k+1/2}(\psi^{n+1,m}_{i+1,j,k+1/2}-\psi^{n+1,m}_{i-1,j,k+1/2})\\
& -G^{3,1}K^{n+1,m}_{i,j,k-1/2}(\psi^{n+1,m}_{i+1,j,k-1/2}-\psi^{n+1,m}_{i-1,j,k-1/2})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{3,2}K^{n+1,m}_{i,j,k+1/2}(\psi^{n+1,m}_{i,j+1,k+1/2}-\psi^{n+1,m}_{i,j-1,k+1/2})\\
& -G^{3,2}K^{n+1,m}_{i,j,k-1/2}(\psi^{n+1,m}_{i,j+1,k-1/2}-\psi^{n+1,m}_{i,j-1,k-1/2})\Big\},
\end{aligned}
$$

and the second term on the right-hand side of Eq. (3.9) is given as

$$\begin{aligned} & \sum_{p=1}^{3} \frac{\partial}{\partial \xi_p} \left(H^p K^{n+1,m} \right)_{i,j,k} \\ = \; & \left(H^1 K^{n+1,m} \right)_{i+1/2,j,k} - \left(H^1 K^{n+1,m} \right)_{i-1/2,j,k} \\ & + \left(H^2 K^{n+1,m} \right)_{i,j+1/2,k} - \left(H^2 K^{n+1,m} \right)_{i,j-1/2,k} \\ & + \left(H^3 K^{n+1,m} \right)_{i,j,k+1/2} - \left(H^3 K^{n+1,m} \right)_{i,j,k-1/2} , \end{aligned} \tag{3.11}$$

where subscripts i, j, and k denote the spatial coordinates in the ξ_1, ξ_2, and ξ_3 axes, respectively, in the computational space. The 1/2 coefficients are originated from the terms of $\partial\psi/\partial\xi_q$ on $G^{p,q}$, where p is not equal to q. For example, where $p = 1$ and $q = 2$ in the fourth term on the right-hand side of Eq. (3.10), $(\partial\psi/\partial\xi_q)_{i+1/2,j,k}$ is evaluated as $(\psi^{n+1,m}_{i+1/2,j+1,k} - \psi^{n+1,m}_{i+1/2,j-1,k})/2\Delta\xi_2$. Because the grid size in the computational space is equal to 1, the term yields $(\psi^{n+1,m}_{i+1/2,j+1,k} - \psi^{n+1,m}_{i+1/2,j-1,k})/2$. The first, second, and third terms on the right-hand side of Eq. (3.10), including ψ^{m+1}, are calculated implicitly, and the other terms on the right-hand side of Eq. (3.10), including ψ^m, are calculated explicitly.

If all the terms are considered implicitly, the 19 unknown variables need to be included in the linearized simultaneous equation, as shown in Fig. 3.2-(a). In contrast, in the scheme proposed here, the linearized simultaneous system includes only seven unknown variables ($\psi^{n+1,m+1}_{i+1,j,k}$, $\psi^{n+1,m+1}_{i,j+1,k}$, $\psi^{n+1,m+1}_{i,j,k+1}$, $\psi^{n+1,m+1}_{i,j,k}$, $\psi^{n+1,m+1}_{i,j,k-1}$, $\psi^{n+1,m+1}_{i,j-1,k}$, and $\psi^{n+1,m+1}_{i-1,j,k}$) from Eq. (3.10), as shown in Fig. 3.2-(b). This makes the proposed scheme both simpler and more efficient when the cross-derivative terms are not dominant in the system. The cross-derivative terms make negligible contributions when the grid is not highly skewed. However, if the grid is highly skewed, the cross-derivative terms make a large contribution to the overall system, and the method of evaluating the cross-derivative terms might require a small time-step duration. This issue will be discussed in Section 3.4.2. Irrespective of the scheme used, the 7-point stencil of $G^{p,q}K$ is utilized, as shown in Fig. 3.2-(c).

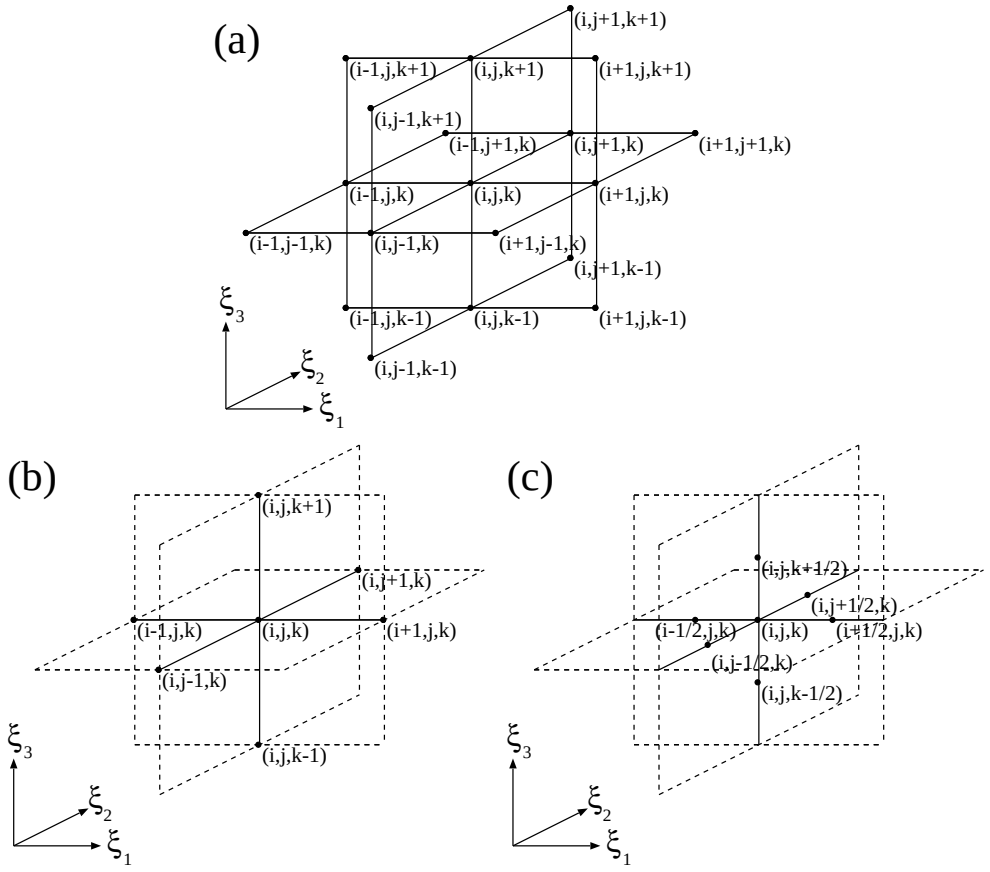

FIGURE 3.2: Evaluated (a) 19-point stencil of ψ and (b) 7-point stencil of ψ. (c) Utilized 7-point stencil of K and G in an iterative procedure.

The hydraulic conductivity and ψ of the boundary between the adjacent nodes used the arithmetic mean is given as

$$(K_{r,s})_{i\pm1/2,j,k} = \frac{1}{2}\left\{(K_{r,s})_{i\pm1,j,k} + (K_{r,s})_{i,j,k}\right\}, \tag{3.12}$$

$$\psi_{i\pm1/2,j,k} = \frac{1}{2}\left(\psi_{i\pm1,j,k} + \psi_{i,j,k}\right). \tag{3.13}$$

The linearized simultaneous system are solved by LIS in this chapter. On the basis of test simulations, we selected a pair of symmetric successive over-relaxation (SSOR) preconditioners and a biconjugate gradient (BICG) method, which was shown to be faster than the other pairs provided by the LIS.

The iteration of Eq. (3.9) continues until the difference between the calculated values of ψ or θ of two successive iteration levels becomes less than the user-specified tolerances,

i.e., until the following inequality is satisfied for all cells:

$$|\psi^{n+1,m+1} - \psi^{n+1,m}| \leq \delta_\psi, \tag{3.14}$$

$$|\theta^{n+1,m+1} - \theta^{n+1,m}| \leq \delta_\theta, \tag{3.15}$$

where δ_ψ and δ_θ are the convergence tolerances. Results of a previous study showed that the θ-based convergence tolerance of Eq. (3.13) could help the model converge when simulating infiltration problems with dried soil [32]. In the test simulations of this chapter, $\delta_\psi = 0.001$ m and $\delta_\theta = 0.0001$ were used, except in Test 1 where the steady-state condition was simulated. The convergence tolerance in Test 1 is $\delta_\psi = 10^{-4}$ m.

3.3.3 Metrics evaluation

The metrics set and Jacobian determinant are defined as

$$\begin{aligned}
&\xi_x = J(y_\eta z_\zeta - y_\zeta z_\eta), \quad \xi_y = J(z_\eta x_\zeta - z_\zeta x_\eta), \quad \xi_z = J(x_\eta y_\zeta - x_\zeta y_\eta), \\
&\eta_x = J(y_\zeta z_\xi - y_\xi z_\zeta), \quad \eta_y = J(z_\zeta x_\xi - z_\xi x_\zeta), \quad \eta_z = J(x_\zeta y_\xi - x_\xi y_\zeta), \\
&\zeta_x = J(y_\xi z_\eta - y_\eta z_\xi), \quad \zeta_y = J(z_\xi x_\eta - z_\eta x_\xi), \quad \zeta_z = J(x_\xi y_\eta - x_\eta y_\xi), \\
&\frac{1}{J} = x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) + x_\eta(y_\zeta z_\xi - y_\xi z_\zeta) + x_\zeta(y_\xi z_\eta - y_\eta z_\xi),
\end{aligned} \tag{3.16}$$

where $(\xi_p)_{x_r} = \partial \xi_p / \partial x_r$ and $(x_r)_{\xi_p} = \partial x_r / \partial \xi_p$. The Jacobian determinant J is used at a nodal point, as shown in Fig. 3.2-(a), and the mesh skewness tensor G is used between adjacent nodes, as shown in Fig. 3.2-(c). Because the computational cell size is equal to 1, $\partial x_r / \partial \xi_p$ at a nodal point (i, j, k) is evaluated as

$$\begin{aligned}
\left.\frac{\partial x_r}{\partial \xi_1}\right|_{i,j,k} &= \frac{(x_r)_{i+1,j,k} - (x_r)_{i-1,j,k}}{2}, \qquad \left.\frac{\partial x_r}{\partial \xi_2}\right|_{i,j,k} = \frac{(x_r)_{i,j+1,k} - (x_r)_{i,j-1,k}}{2}, \\
\left.\frac{\partial x_r}{\partial \xi_3}\right|_{i,j,k} &= \frac{(x_r)_{i,j,k+1} - (x_r)_{i,j,k-1}}{2}.
\end{aligned} \tag{3.17}$$

For evaluating metrics at a boundary node, an additional row of ghost nodes with zero thickness is assumed to be added at the outside nodes, as shown in Fig. 3.3 reffering to *Peric* [70]. These ghost nodes are used only for evaluating the metrics at the boundary nodes. For example, if $i = 0$, which is at the real boundary node, $\partial x_r / \partial \xi_1$ of Eq. (3.17)

cannot be calculated because $(x_r)_{-1,j,k}$ is not defined. However, using the ghost nodes as $(x_r)_{-1,j,k} = (x_r)_{0,j,k}$, the value of $\partial x_r/\partial\xi_1$ can be easily evaluated. The other metrics at boundary nodes, such as $\partial x_r/\partial\xi_2$ and $\partial x_r/\partial\xi_3$, are evaluated in the same way.

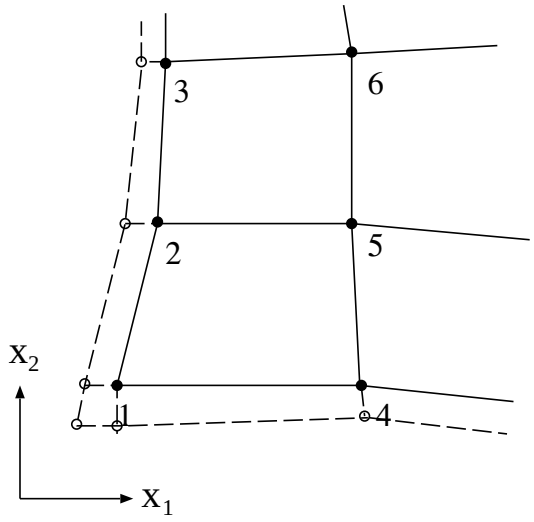

FIGURE 3.3: Additional row of ghost nodes (unfilled circle) with zero thickness for evaluating the metrics at boundary nodes.

The metrics between adjacent nodes for calculating G are evaluated as

$$\left.\frac{\partial x_r}{\partial\xi_1}\right|_{i+1/2,j,k} = (x_r)_{i+1,j,k} - (x_r)_{i,j,k}, \qquad \left.\frac{\partial x_r}{\partial\xi_2}\right|_{i,j+1/2,k} = (x_r)_{i,j+1,k} - (x_r)_{i,j,k}, \tag{3.18}$$
$$\left.\frac{\partial x_3}{\partial\xi_r}\right|_{i,j,k+1/2} = (x_r)_{i,j,k+1} - (x_r)_{i,j,k}.$$

3.3.4 Boundary condition

There are two main types of boundary conditions, namely, the Dirichlet boundary condition and the Neumann boundary condition. Dirichlet boundaries are treated in the same way as in the case of the other FDMs (predetermined ψ). Under the Neumann boundary condition, the flux has to be transformed. If the flux $q_{p'}$ is defined as coming from outside to the (i,j,k)-node and the element of $q_{p'}$ is (q_1, q_2, q_3) in the physical space, $q_{p'}$ can be transformed as

$$q_{p'} = \sum_{r=1}^{3} \frac{q_r}{J}\frac{\partial\xi_{p'}}{\partial x_r}, \tag{3.19}$$

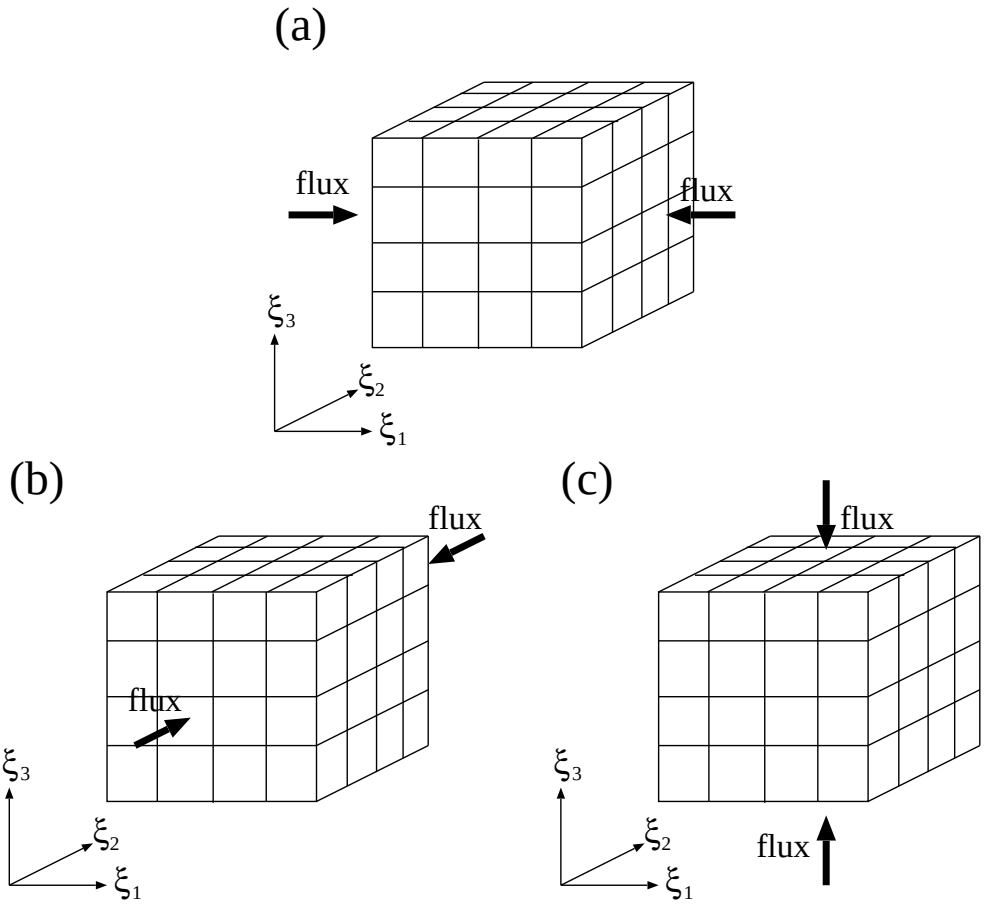

FIGURE 3.4: Flux directions under various boundary conditions.

where p' is 1, 2, and 3 when the flux passes through surfaces of $\xi_2\xi_3$, $\xi_3\xi_1$, and $\xi_1\xi_2$, as shown in Fig. 3.4-(a), (b), and (c).

The seepage face is treated as follows [15]. If the location of the seepage face is known, all the nodes along the seepage face can be treated as the Dirichlet boundary ($\psi = 0$). However, the exact range of the seepage face is usually unknown until the problem is solved. Hence, the seepage face has to be determined iteratively. At the first iteration, the location of the seepage face is approximated at the same location as in the previous time step. If the guess is correct, it is assumed that the flux of the nodes along the seepage face must be outward, and the values of ψ at the boundary nodes above the seepage face are negative. If the nodes where $\psi = 0$ have a net inward flux, the nodes where the flux is inward are assumed to be non-seepage faces. If the nodes above the seepage face have positive values for ψ, it means that the nodes where ψ is positive should be set as the seepage face. On the basis of these principles, the location

of the seepage face is determined iteratively. Details of the determination procedure are described by *Neuman* [64].

3.3.5 Grid generation

Grid generation is a large area of research by itself. Most studies on grid generation for a block-structured grid have been conducted by the use of either algebraic methods or partial differential equation (PDE) methods. Although algebraic methods have a major advantage of a rapid computation, they are generally less preferred to PDE methods owing to the lack of grid smoothness [45]. It seems that the most widely used system of PDE methods is the Poisson equation with specified control functions. These specified control functions allow interior grid nodes to be concentrated in specific regions or orthogonally positioned at the boundaries. *Thompson et al.* [87] presented a comprehensive review of numerical grid generation methods and a detailed procedure of grid generation for block-structured grids. In this chapter, an algebraic method was used for generating grids for Test 1∼3, and a Poisson equation system was used for Test 4.

3.4 Numerical simulation

The test simulations were carried out to verify the performance of the proposed model. The first and second test simulations were carried out to evaluate the performance of the proposed model with different mesh skewness. The third and fourth test simulations were carried out to examine the performance of the proposed model with non-rectangular and curvilinear-shaped flow domains. The results of the first, second, and third test simulations were compared with either an exact solution or a numerical solution with higher resolution. For the sake of comparison, the test simulations were also carried out using HYDRUS or the conventional FDM, and the resulting performance of HYDRUS or the conventional FDM was compared with that of the proposed model.

Van Genuchten [91]'s equation for the soil water retention curve and *Mualem* [61]'s equation for the unsaturated hydraulic conductivity function were used. Detail of these models are described in as Section 2.3. The mass balance error and the relative error are also defined in Section 2.3. In test simulations of this chapter, the soil properties listed in Table 2.1 are used. These values represent average values for the selected soil

water retention and hydraulic conductivity parameters for major soil textural groups. The time-step duration are adjusted automatically as Section 2.3.

3.4.1 Test 1: steady-state simulation with a highly skewed mesh

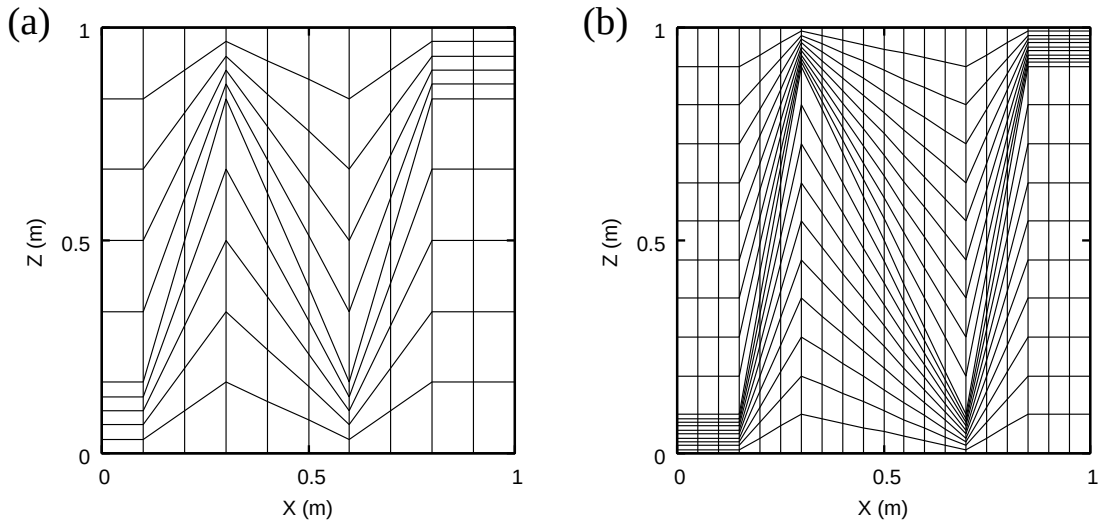

FIGURE 3.5: (a) The 10 × 10 and (b) 20 × 20 Kershaw meshes used in Test 1.

To test the model performance, a two-dimensional steady-state simulation with a highly skewed mesh was carried out using *Kershaw*'s [1981] mesh, which is often used for testing the accuracy of a diffusion equation. A 1-m square domain was considered using two types of meshes, as shown in Fig. 3.5. The 20 × 20 mesh had a more severely skewed shape than the 10 × 10 mesh. The properties of silt soil, listed in Table 2.1, were considered. The top and bottom boundaries were $\psi_{top} = -0.5$ m and $\psi_{bot} = 0.5$ m, respectively, and both side boundaries were assumed to be no-flow boundaries. In fact, the correct solution is a linear distribution as $\psi(z) = 0.5 - z$ m.

Isolines of the pressure head calculated by the proposed model are shown in Figs. 3.6-(a) and (b). The isoline plots of the pressure head are straight lines that are independent of the mesh shape. Even with the highly skewed mesh, the model correctly described the distribution of the pressure head. The root mean square errors of the pressure head are shown in Figs. 3.6-(c) and (d). The overall error of the 20 × 20 mesh was larger than that of the 10 × 10 mesh, even though the 20 × 20 mesh had a finer spatial resolution. In general, the accuracy of a model using a non-orthogonal grid is influenced by the mesh quality, with high non-orthogonality reducing the numerical

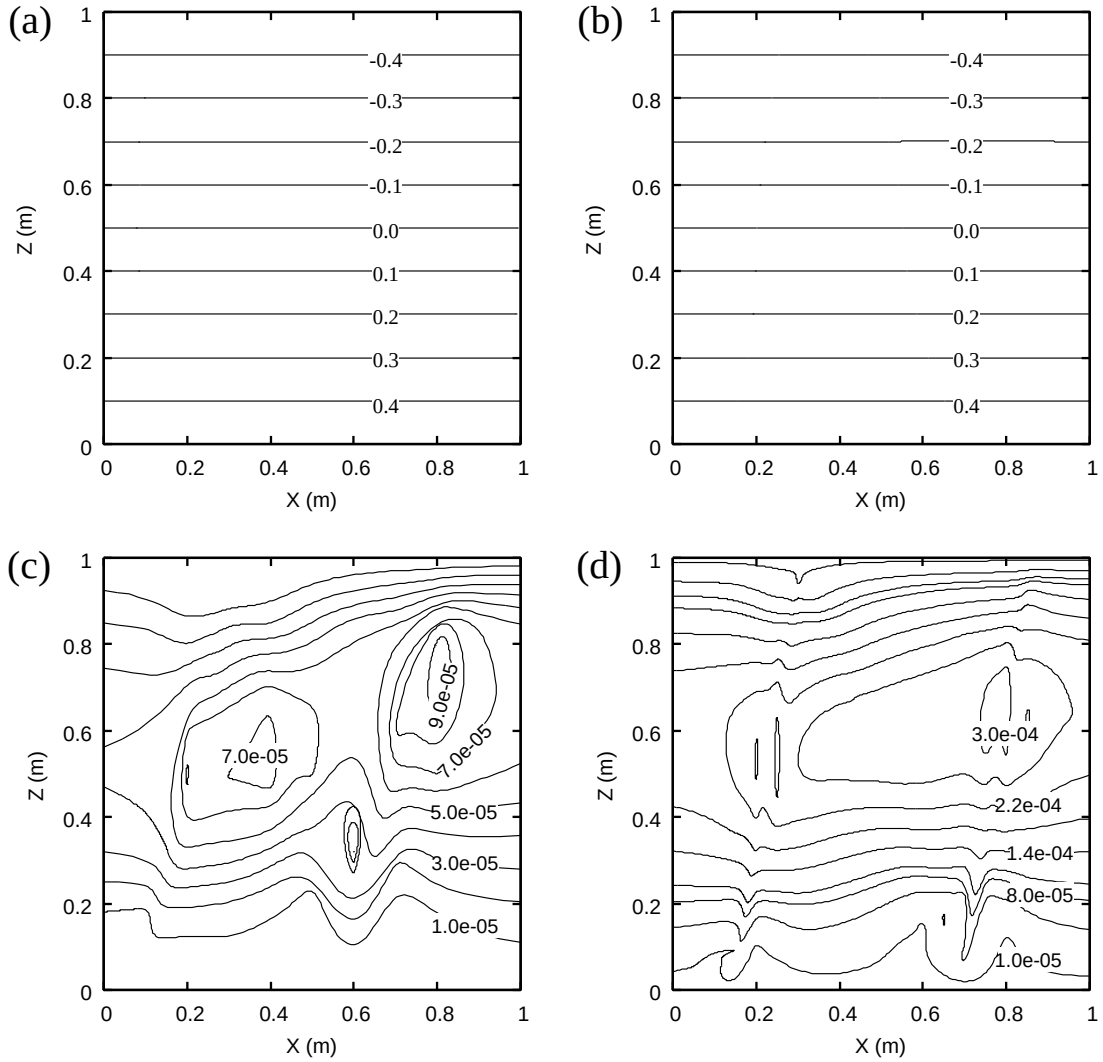

FIGURE 3.6: Isolines of pressure head with (a) 10 × 10 mesh and (b) 20 × 20 mesh and the root mean square error of pressure head for the (c) 10 × 10 mesh and (d) 20 × 20 mesh results in Test 1.

accuracy for truncation error [58]. Therefore, it is preferable to avoid highly skewed grids for numerical accuracy.

3.4.2 Test 2: unsteady-state simulation to investigate the non-orthogonality effect

This test simulation was carried out to investigate the effect of grid skewness on the performance and accuracy of the proposed model. As discussed in Section 3.3.2, evaluation of the cross-derivative terms might incur small time-step durations with a highly skewed grid. To study the effect of calculating the cross-derivative terms with a partial

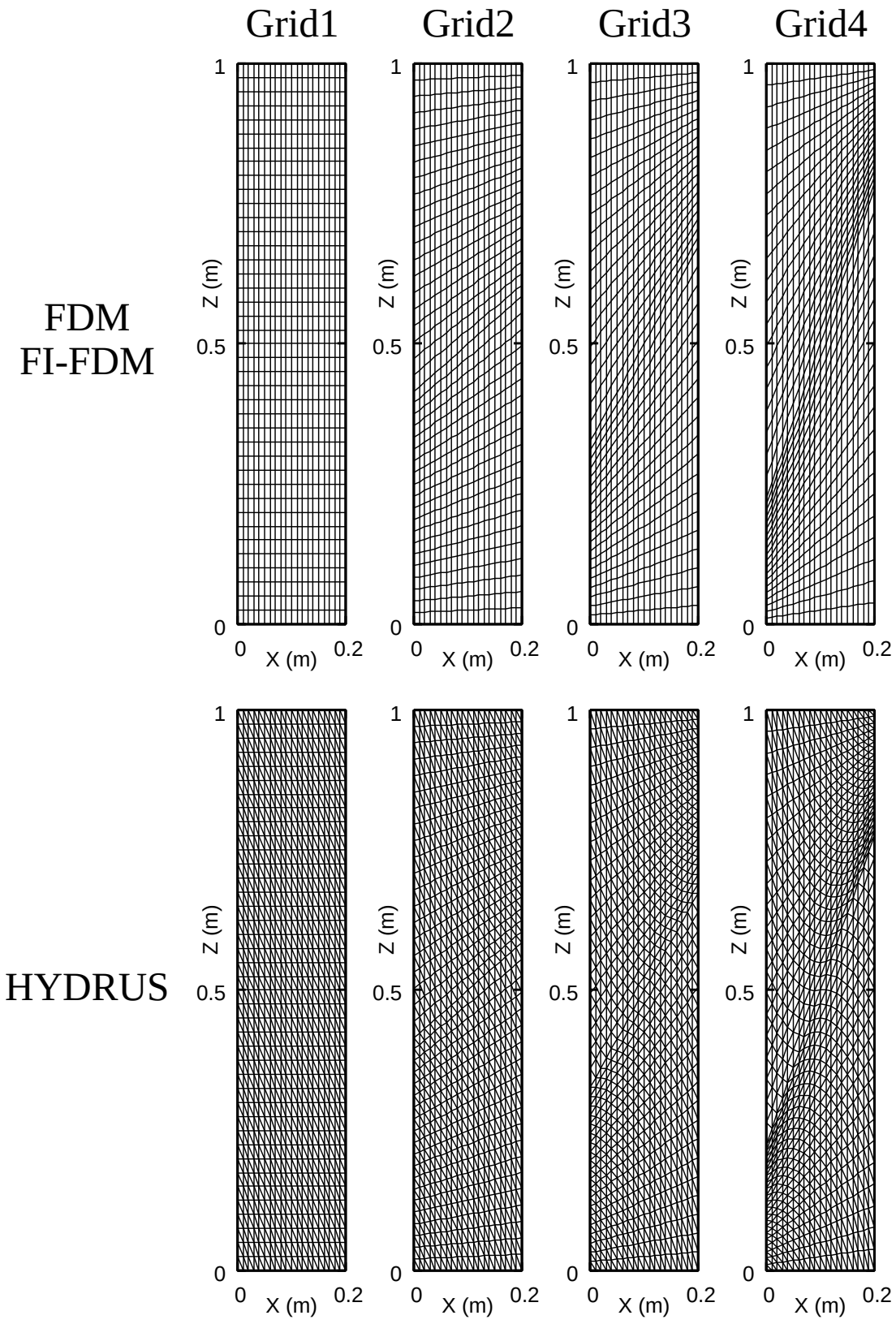

FIGURE 3.7: Four types of grids used in Test 2 (20×40).

Table 3.1: Test simulation 2, Relative error.

Grid	Number of cells	Relative error[a]		
		FDM	FI-FDM	HYDRUS
Grid1	20×20	6.09e-02	6.09e-02	5.66e-02
	20×40	2.29e-02	2.29e-02	1.78e-02
	20×80	7.17e-03	7.17e-03	5.46e-03
Grid2	20×20	4.89e-02	5.00e-02	6.90e-02
	20×40	1.66e-02	1.76e-02	1.65e-02
	20×80	4.24e-03	4.86e-03	3.90e-02
Grid3	20×20	2.75e-02	3.02e-02	8.73e-02
	20×40	1.50e-02	1.36e-02	2.83e-02
	20×80	1.53e-02	1.22e-02	5.71e-03
Grid4	20×20	6.53e-02	5.88e-02	1.07e-01
	20×40	6.86e-02	5.70e-02	3.00e-02
	20×80	7.00e-02	5.77e-02	5.40e-02

[a] Relative error is evaluated at the end of simulation.

implicit method, the proposed model (FDM) and a fully implicit FDM (FI-FDM) were used. Except for the treatment of the cross-derivative terms, the FI-FDM is identical to the FDM presented in this paper. The preconditioned biconjugate gradient method permits the system matrix to be non-diagonally dominant. The FI-FDM converges with effective preconditioning on a 19-point stencil. HYDRUS was also run for comparison. As discussed in Section 3.4.1, a highly skewed mesh might reduce the accuracy of the results. To assess the effect of grid skewness on the accuracy of the proposed model, four different types of grids with a 0.2 × 1 m rectangular domain were considered, as shown in Fig. 3.7. To estimate the numerical accuracy, grids of 20 × 20 mesh, 20 × 40 mesh, and 20 × 80 mesh were used. Grid 1 was orthogonal. The smallest angles of grids 2, 3, and 4 were 50°, 30°, and 20°, respectively. Because HYDRUS uses a triangular unstructured grid, HYDRUS grids were generated by modifying the structured FDM grid. The nodes of both the grids were arranged such that they were located at the same coordinate positions. Loam soil properties listed in Table 2.1 were considered. The top and both side boundaries were no-flow boundaries, and the bottom boundary was $\psi_{bot} = 0.5$ m. The initial condition was $\psi_{init} = -1$ m. The simulation period was 1 day. The time-step duration was adjusted automatically by the algorithm presented earlier.

Fig. 3.8 shows the pressure head profiles for $x = 0.1$ m at the end of the simulation carried out using the three models. In this test simulation, the dense orthogonal grid

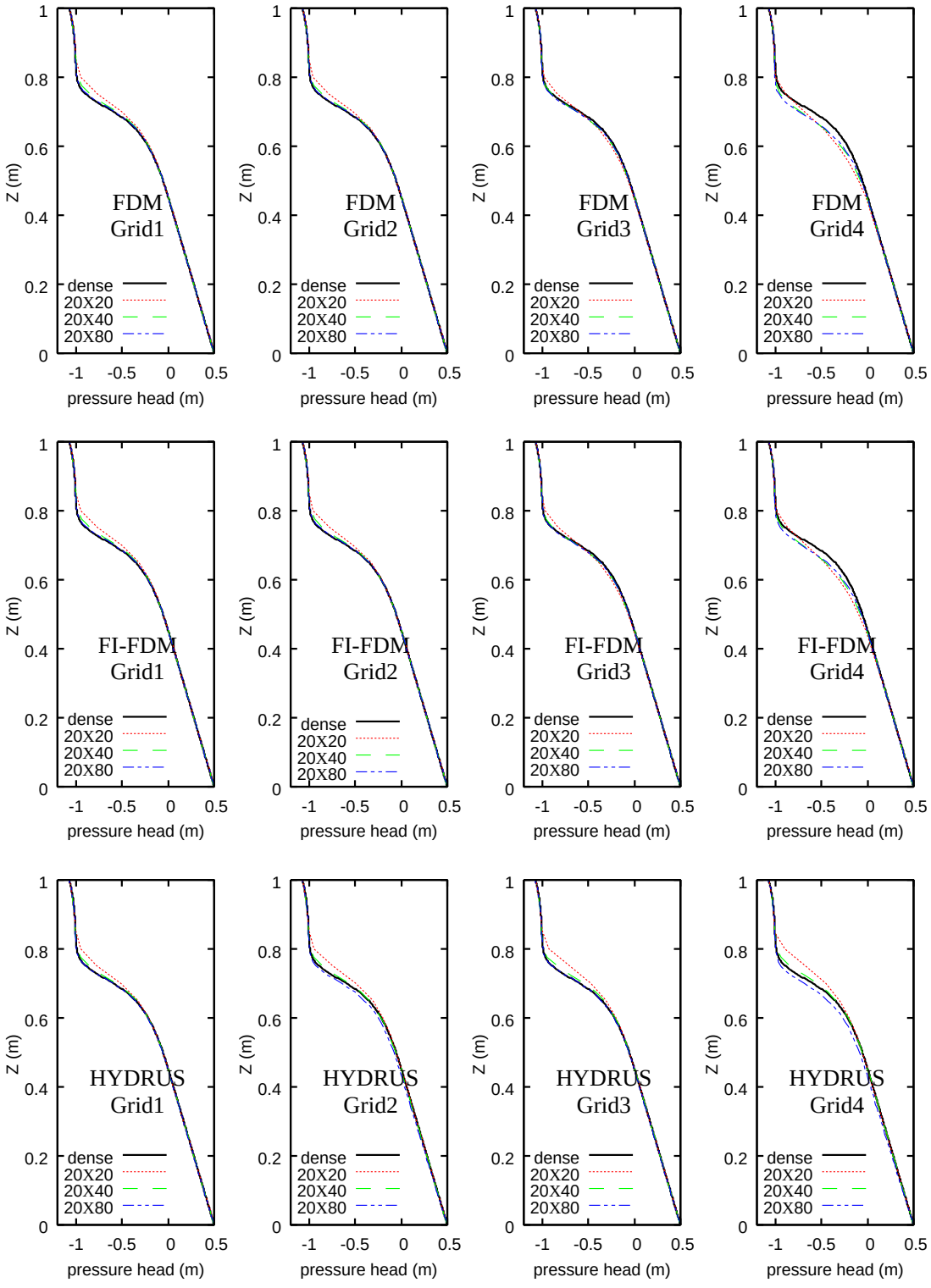

FIGURE 3.8: Pressure head profiles on $x = 0.1$ m obtained at the end of simulations carried out by FDM, FI-FDM, and HYDRUS in Test 2.

Table 3.2: Test simulation 2, Mass balance error.

Grid	Number of cells	MBE[a] (%)		
		FDM	FI-FDM	HYDRUS
Grid1	20×20	2.66e-02	2.66e-02	3.51
	20×40	2.44e-02	2.44e-02	3.31
	20×80	1.92e-02	1.92e-02	3.65
Grid2	20×20	4.22e-03	5.66e-04	8.83e-01
	20×40	3.42e-04	7.66e-04	4.20
	20×80	1.72e-03	2.08e-03	11.56
Grid3	20×20	4.57e-03	6.90e-04	3.32e-01
	20×40	1.15e-04	4.69e-04	9.96e-01
	20×80	2.01e-03	6.89e-04	3.94
Grid4	20×20	1.28e-02	9.34e-04	6.01e-01
	20×40	1.94e-04	3.27e-03	2.68
	20×80	1.80e-03	9.56e-03	18.51

[a] Mass balance error are evaluated at the end of simulation.

($\Delta z = 0.0025$ m) solution was considered to be surrogate solution for the exact solution. When Grids 1 and 2 were used, it was observed that the solutions obtained by the three models approached the exact solution when a finer grid was used; however when Grids 3 and 4 were used, it was observed that the solutions obtained by the three models did not approach the exact solution. Table 3.1 and 3.2 list the relative error of the pressure head and the mass balance error generated by the three models. The FDM and the FI-FDM gave the same orders of relative errors in all cases, indicating that the method of evaluating the cross-derivative terms does not affect numerical accuracy in this test simulation. HYDRUS also produced almost the same order of errors. FDM has a first-order temporal and a second-order spatial precision in the computational space. HYDRUS also has a first-order temporal and a second-order spatial precision in the physical space. When Grid 1, i.e., the orthogonal grid, was used, it was observed that the relative errors of the three models decreased with an increase in the spatial resolution of the grid. However, when Grids 3 and 4 were used, it was observed that the relative errors of the three models were larger than the relative errors obtained when Grid 1 was used, and the errors did not decrease as the grid became finer. These results suggest that the highly skewed mesh reduced the numerical accuracy of the models. The FDM and the FI-FDM showed virtually perfect mass conservation in this test simulation, whereas HYDRUS generated comparatively large mass balance errors, especially when Grids 2 and 4 with dimension of 20×80 were used. The mass balance performances of the three models are shown in Fig. 3.9. Even though the FDM and the FI-FDM underestimated

TABLE 3.3: Test simulation 2, model performance.

Grid	Number of cells	CPU time (sec)			Iteration		
		FDM	FI-FDM	HYDRUS	FDM	FI-FDM	HYDRUS
Grid1	20×20	4	4	4	787	787	2074
	20×40	14	15	25	1279	1279	5959
	20×80	35	44	140	1843	1843	16555
Grid2	20×20	3	4	5	803	765	2421
	20×40	14	16	25	1470	1460	6278
	20×80	60	79	90	2959	2938	11254
Grid3	20×20	5	3	4	1007	602	1734
	20×40	16	14	21	1892	1247	5111
	20×80	75	77	109	3310	2766	13063
Grid4	20×20	5	3	5	1227	531	2479
	20×40	21	14	33	2502	1178	7679
	20×80	81	71	84	4242	2446	9976

the total amount of additional mass and net flux when Grid 4 was used, their balances were virtually perfect. On the other hand, HYDRUS apparently overestimated the net flux flowing into the domain, resulting in the generation of mass balance errors.

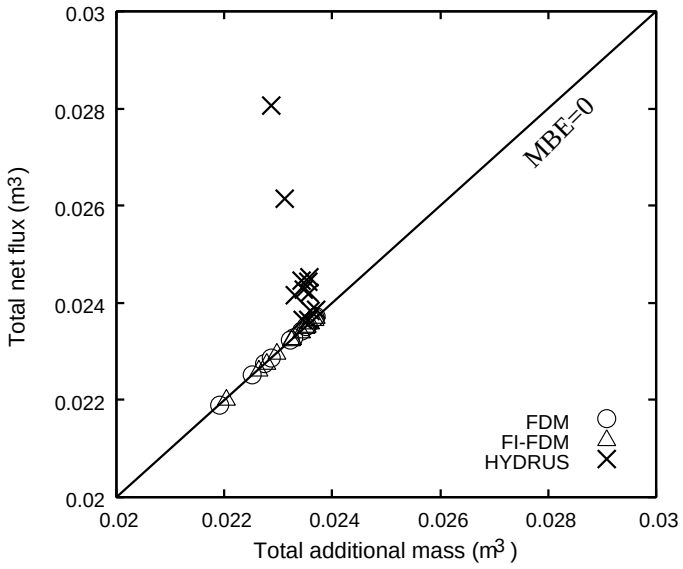

FIGURE 3.9: Mass balances of FDM, FI-FDM, and HYDRUS in Test 2.

Table 3.3 lists the CPU time and the total number of iterations required for the three models. In the cases of Grids 3 and 4, the FDM required more iterations and CPU time than the FI-FDM, whereas there was no significant difference in the total iteration number. The CPU time consumed by Grids 3 and 4 was less than that consumed by Grids 1 and 2. When Grids 3 and 4 were used, it was found that implicit evaluation

of the cross-derivative terms of the FI-FDM permitted large time-step durations. On the other hand, the FDM consumed fewer CPU resources per iteration than the FI-FDM, because the FDM included fewer unknown values in its simultaneous equation than the FI-FDM, as mentioned in Section 3.3.2. There exists a trade-off between the FDM consuming fewer CPU resources per iteration and the FI-FDM requiring less iterations when the cross-derivative terms are dominant. The total CPU times consumed by the FDM and FI-FDM were not significantly different in this test case even when a highly skewed mesh was used. However, for very difficult problems (e.g., problems with high degree of heterogeneity and anisotropy with highly skewed grids), the explicit evaluation of the cross-derivative terms might significantly affect the convergence speed and accuracy. HYDRUS required a greater number of iterations than did FDM and FI-FDM, in all cases. A comparison between the CPU time consumed by the FDM and HYDRUS cannot be drawn because the two models are running on different operating systems and coded by different programming languages. Hence, the CPU time consumed by HYDRUS was just shown for reference.

This test simulation was carried out to estimate the effect of grid skewness on the model performance and accuracy. A comparison between the FDM and the FI-FDM, considering all the terms fully implicitly, showed that the proposed method of evaluating the cross-derivative terms did not reduce numerical accuracy in this test case while it incurred small time-step durations with highly skewed grids, even though other cases may exist in which the numerical accuracy could be affected, as mentioned above. We also confirmed that the proposed model and the finite-element model generally had the same order of numerical accuracy with the same node positions. Furthermore, the mass conservation behavior of the proposed model was better than that of the FEM in this simulation, considering that these two models used the same mixed-form of Richards' equation.

3.4.3 Test 3: transient variably saturated flow in two dimensions

The model performance for a non-rectangular domain was tested in this simulation. A trapezoid domain shown in Fig. 3.10 was considered, using sandy loam soil properties listed in Table 2.1. The no-flow condition was applied to the top and bottom boundaries. The left-side boundary was given as $\psi_{ls} = 1.5 - z$ m where $z \leq 1.5$ m and a no-flow

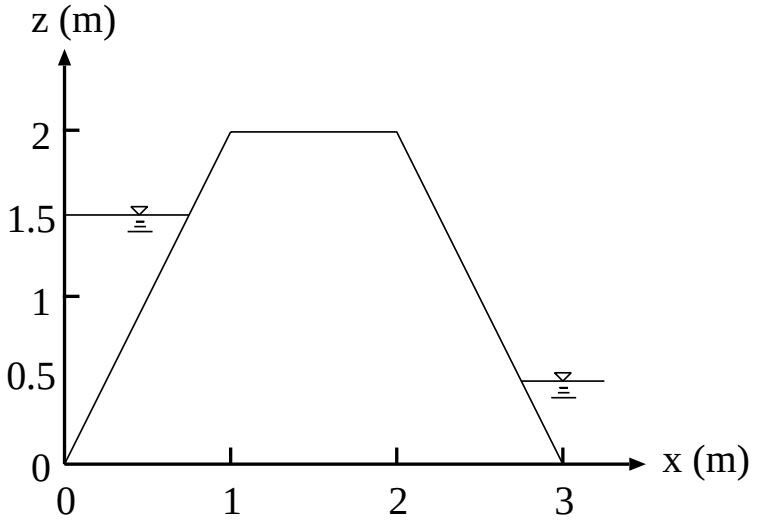

FIGURE 3.10: Flow domain in Test 3.

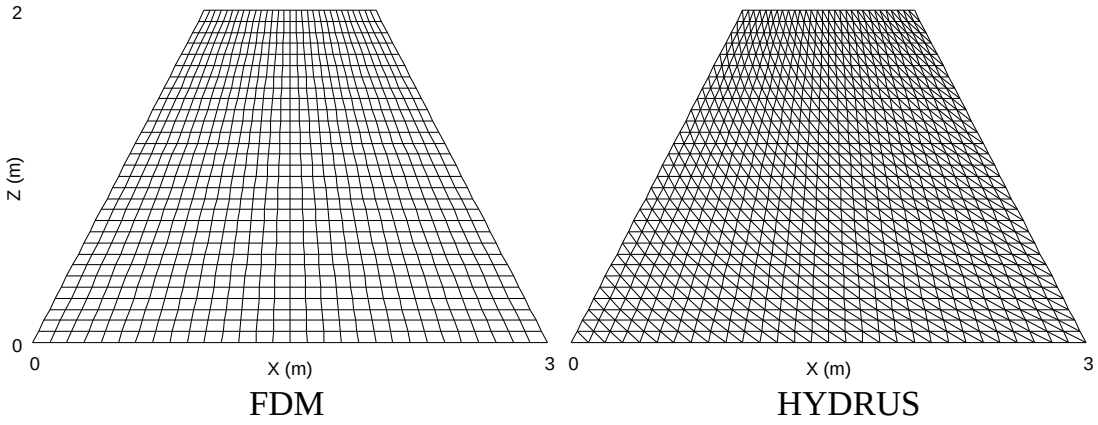

FIGURE 3.11: Grids used in Test 3.

boundary where 1.5 m $< z$. The right side boundary was given as $\psi_{rs} = 0.5 - z$ m where $z \leq 0.5$ m and a seepage-face boundary where 0.5 m $< z$. The initial condition was $\psi_0 = 0.5 - z$ m, and the simulation period was 1 week. The time-step duration was adjusted automatically as $1 \leq \Delta t \leq 60$ min. Fig. 3.11 shows the grids (30×30) for the FDM and HYDRUS. The nodes of both the grids were arranged to be in the same positions.

Fig. 3.12 shows the ψ contours simulated by the FDM and HYDRUS. The two results are in good agreement with each other and successfully describe water flows in the soil. Table 3.4 lists the relative error and the model performance for Test 3. In this test simulation, a dense grid (150×150) solution was used as a surrogate solution for the

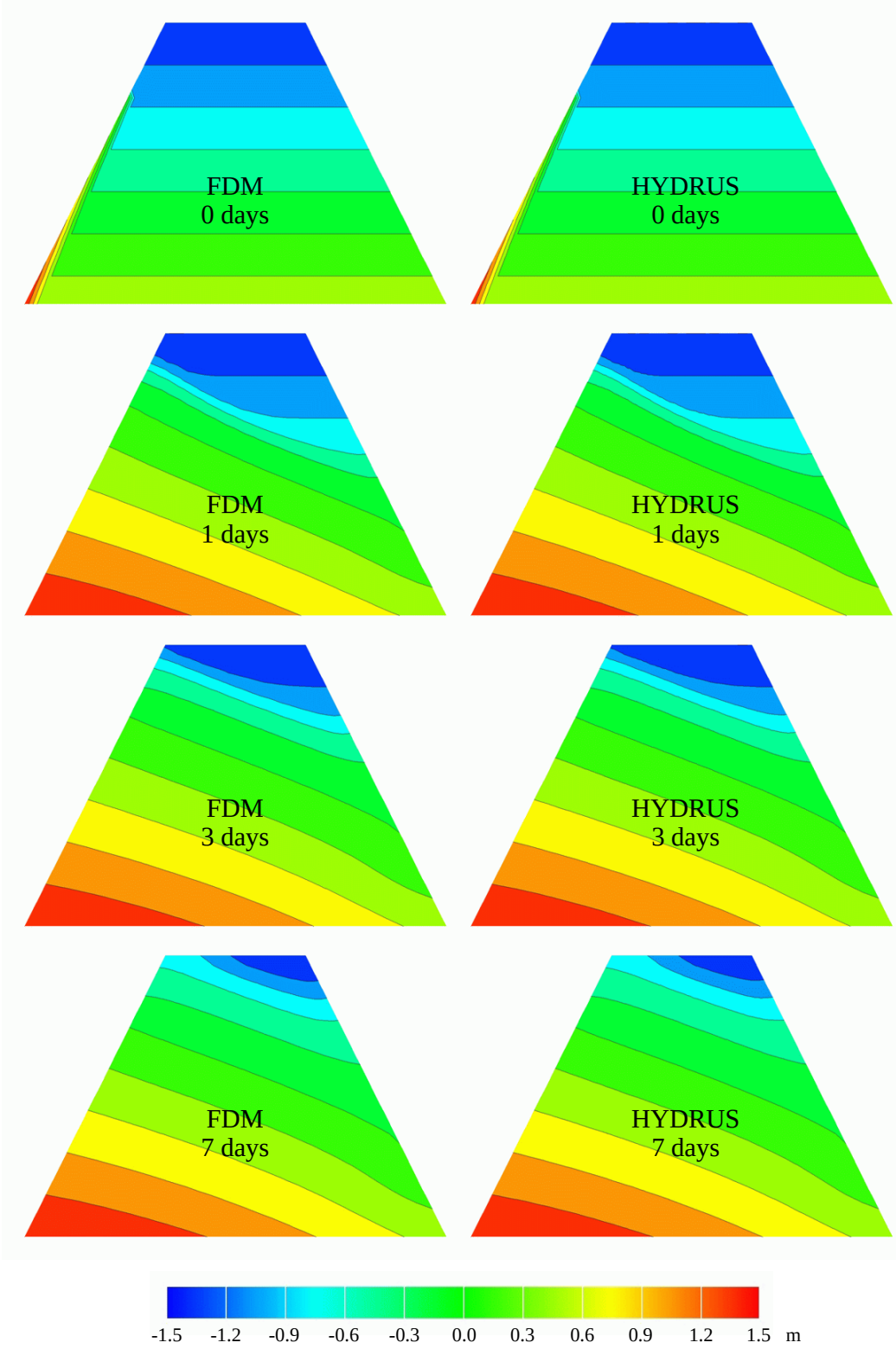

FIGURE 3.12: Contours of pressure head simulated in Test 3; the left and right sides show the results obtained by FDM and HYDRUS, respectively.

TABLE 3.4: Test simulation 3, numerical accuracy and performance.

Model	Relative error[a]	MBE[a] (%)	CPU (sec)	Iteration
FDM	8.75e-02	6.99e-03	12	1692
HYDRUS	8.99e-02	0.32	36	7978

[a] Relative error and mass balace error are evaluated at the end of simulation.

exact solution. As in the case of Test 2, the proposed model and HYDRUS gave the same order of relative error. The number of iteration required by the proposed model was less than that required by HYDRUS. The two models were run using the same computer; however, the FDM was coded by C++ and operated on the Linux system, whereas HYDRUS was coded by Fortran and operated on the Windows system. The two models also used different types of grids. It is not possible to assess model efficiency on the basis of the CPU time required for calculation. However, when the same node number and positions were used, it was clear that the FDM converged more quickly than HYDRUS did because the total number of iterations required for calculation was related to the numerical algorithm and independent of programming languages and operating system used; Fig. 3.13 confirmed this observation, showing the time-step durations of the two models for this calculation. The time-step durations of both the models are controlled by the same rule and adjusted on the basis of the number of iterations required for convergence at the previous time step. As shown in Fig. 3.13, the FDM converged much more rapidly than HYDRUS in this simulation.

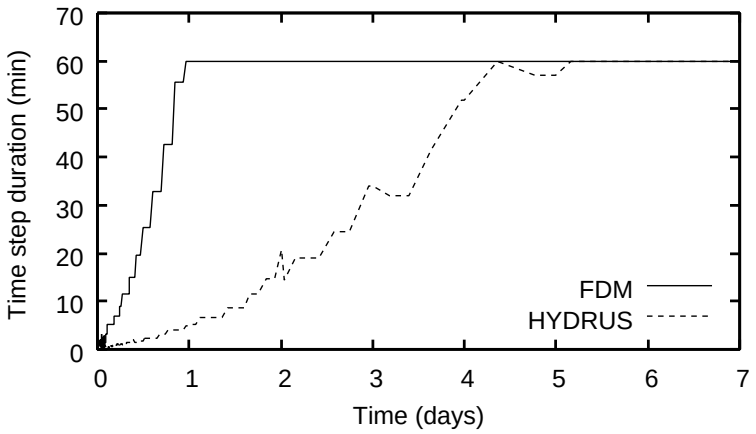

FIGURE 3.13: Time-step durations of FDM and HYDRUS in Test 3.

3.4.4 Test 4: rainfall-runoff simulation for a slope in three dimensions

In Test 4, the FDM and the conventional FDM that uses high-resolution, stepwise, orthogonal grids were compared in a three-dimensional curvilinear domain. A rainfall-runoff simulation for a slope was carried out. Many unsolved problems remain in simulating the rainfall-runoff processes for a real slope, including insufficient knowledge of detailed runoff processes, slope heterogeneity, and high computational cost. However, even given these problems, the numerical approach can be a useful tool for studying the runoff processes of a slope (e.g., [38, 49, 54, 85]).

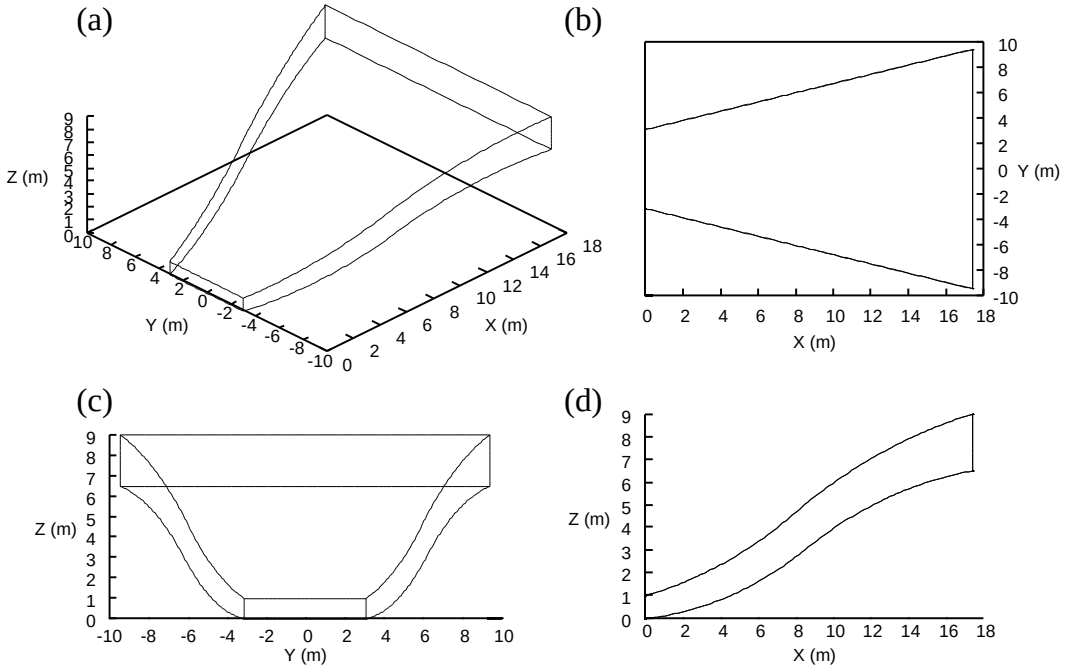

FIGURE 3.14: An inclined domain used in Test 4. (a) Perspective view; (b) plan view; (c) vertical cross section from x,y coordinate (0,0); (d) vertical section from x,y coordinate (9,-10).

A curvilinear slope, as illustrated in Fig. 3.14, was considered, using the properties for sandy soil listed in Table 2.1. The mesh shapes for the FDM, HYDRUS, and the conventional FDM simulations were shown in Fig. 3.15. The number of nodes for FDM, HYDRUS, and the conventional FDM were 4896, 4560, and 34441, respectively. The no-flow condition was applied to the upper, bottom, and both side boundaries. The surface boundary was a flux boundary, and the lower boundary was a seepage-face boundary. The initial conditions were $\psi = 0$ on the intersection of the bottom and the lower nodes and $\psi = -1$ m on the intersection of the surface and the upper nodes; the values for

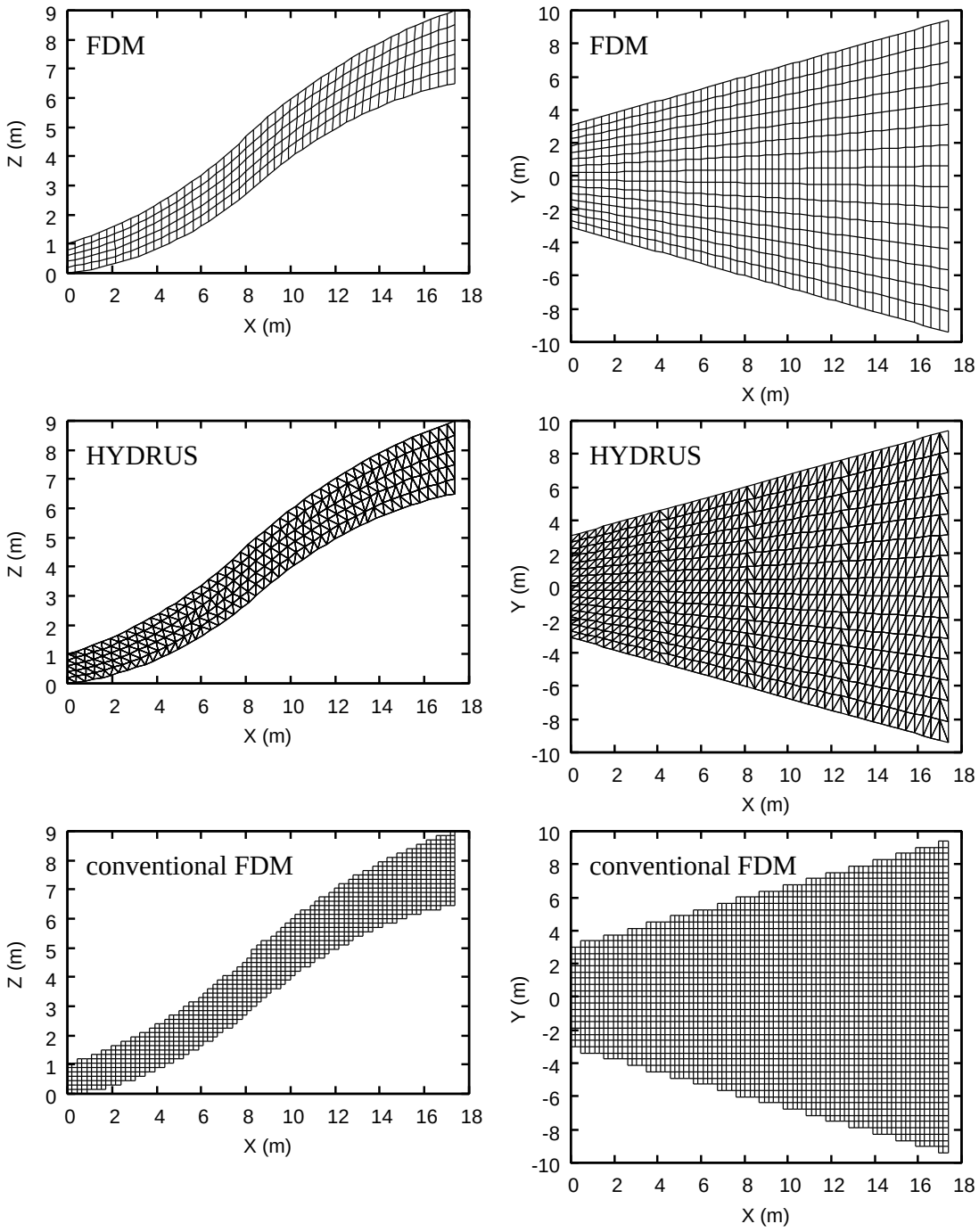

FIGURE 3.15: Grids used in Test 4; left side describes vertical sections from x,y coordinate (9,-10) and right side gives plan views.

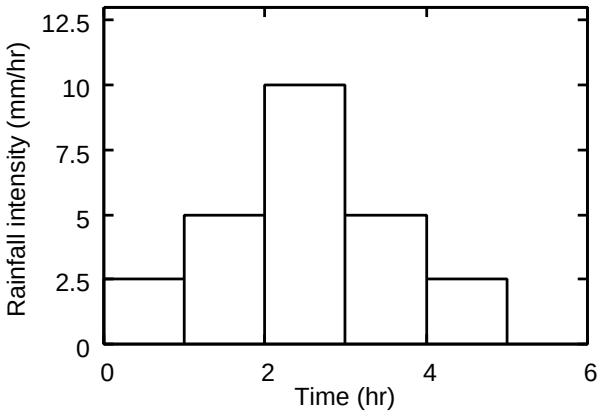

FIGURE 3.16: Rainfall intensity for Test 4.

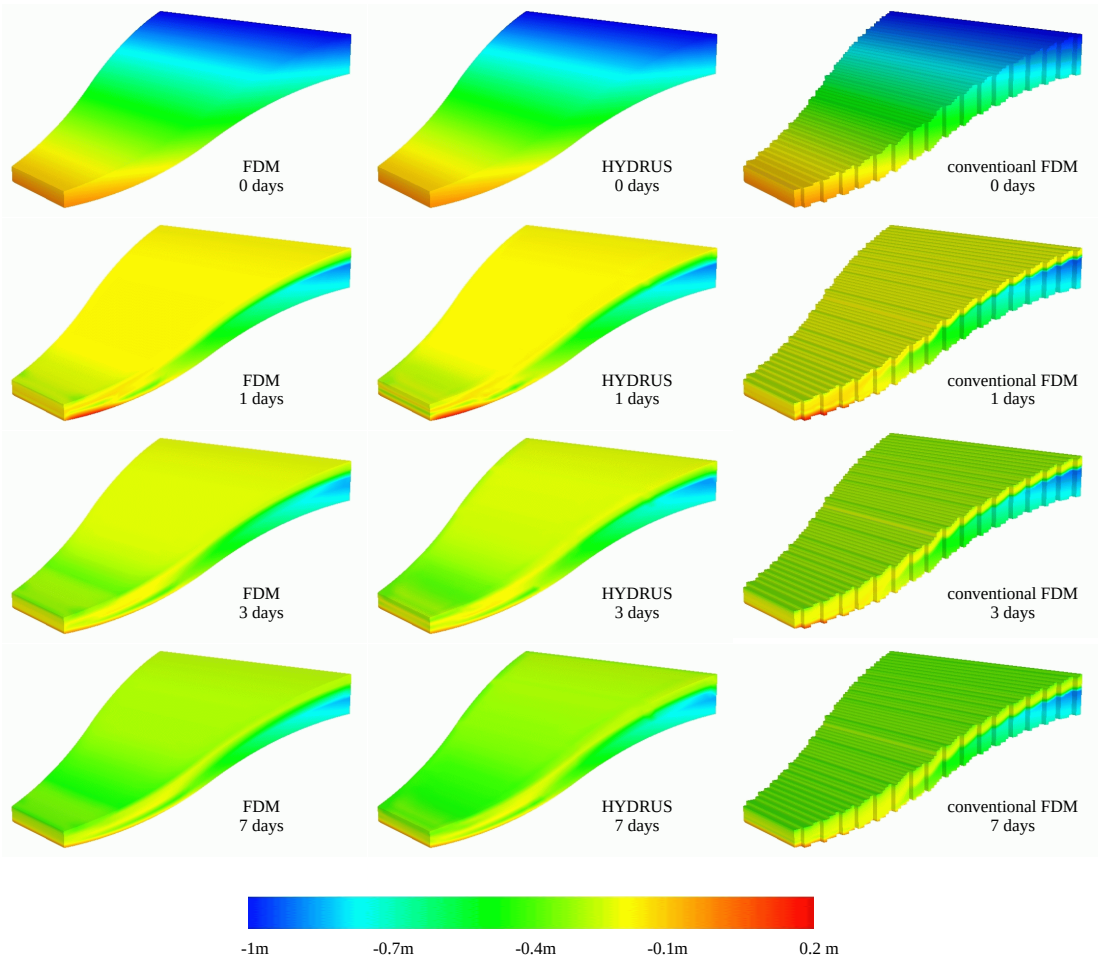

FIGURE 3.17: Pressure head results of Test 4; the left, middle, and right side show the results obtained by FDM, HYDRUS, and conventional FDM, respectively.

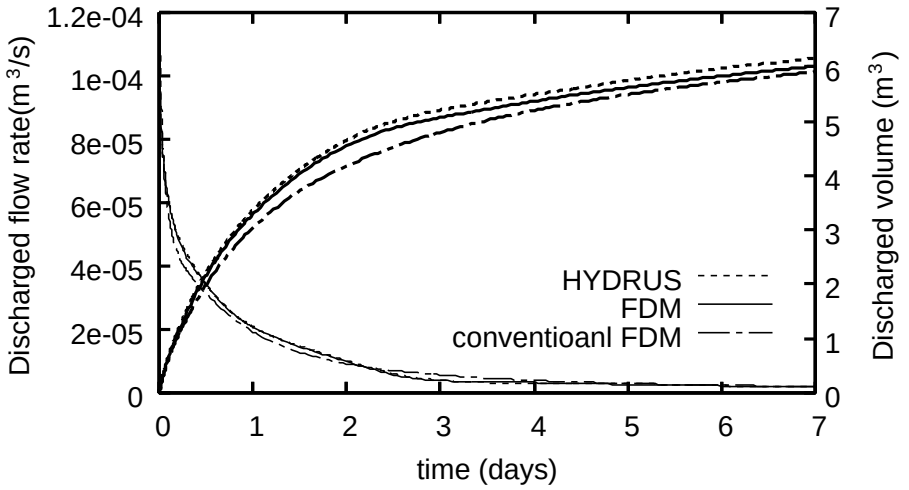

FIGURE 3.18: Discharge flow rate (thin line) and cumulative water volume (thick line) at the lower end of the slope in Test 4.

the other nodes were distributed between those of the top and bottom nodes as a linear function of z. The rainfall intensity was as shown in Fig. 3.16. The simulation period was 2 weeks, and the time-step duration was $1 \leq \Delta t \leq 3600$ sec.

Fig. 3.17 shows the FDM, HYDRUS, and the conventional FDM results obtained for the pressure head. The discharge flow rates and the cumulative discharge volumes at the lower end of the slope are illustrated in Fig. 3.18. The three results are in good agreement with each other, and the FDM successfully simulated the three-dimensional flow on the curvilinear slope, despite a low-resolution grid. The conventional FDM also seems to successfully simulate the test problem. However in order to successfully simulate the test problem, the conventional FDM must have a high-resolution grid, as shown in Fig. 3.15. If a low-resolution grid is used in the conventional FDM, a poor result will be obtained because of insufficient representation of the curvilinear domain. Table 3.5 lists the statistics of the model performance. Owing to additional resolution, the conventional FDM consumed more CPU time than the FDM. If the domain complexity increases, higher additional cost involved with higher-resolution will be required. This implies that the FDM may save considerable computational time as compared to ¡the conventional FDM, e.g., in the case of iterative parameter estimation or Monte Carlo exercise, which usually requires hundreds or thousands of runs to arrive at an ideal parameter set or an objective function. Fig. 3.19 shows the time-step durations. As shown in this figure, the FDM and the conventional FDM converged with almost the

TABLE 3.5: Test simulation 4, mass balance error and model performance.

Model	MBE[a] (%)	CPU (sec)	Iteration	Number of nodes
FDM	0.25	39	702	4896
HYDRUS	0.24	140	3347	4560
conventional FDM	0.48	577	678	34441

[a] MBE is evaluated at the end of simulation.

same speed, though the conventional FDM used a higher-resolution grid than the FDM. It is supposed that partially explicit evaluation of the cross-derivative terms of FDM affected the convergence speed, and the convergence speed of the FDM became almost same as that of the conventional FDM.

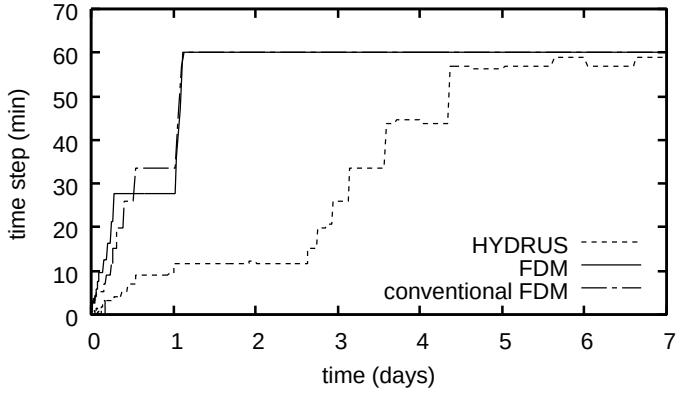

FIGURE 3.19: The time-step durations of FDM, HYDRUS, and conventional FDM for Test 4.

3.5 Summary

In this paper, a finite-difference saturated–unsaturated flow model was presented. This model can fit a curvilinear flow domain. A coordinate transformation method was applied, which enabled the model to handle complex geometries and anisotropies. The proposed scheme has first-order temporal and second-order spatial discretization precisions. From a practical point of view, it is found that the proposed scheme has advantages in terms of ease of coding and less-consuming computation storage because the cross-derivative terms are evaluated at the previous iteration step, and the linearized equation uses a 7-point stencil, as in the case of the conventional finite-different model on an orthogonal grid. Test simulations were carried out to examine the effect of the method of cross-derivative-term evaluation.

Four simulations were carried out to assess the performance of the proposed model. In the first test simulation, a two-dimensional steady-state condition with a highly skewed grid was simulated. The isolines of the pressure head produced by the proposed model were insensitive to the grid shape, and the results were in good agreement with the correct solution. The numerical accuracy was also found to reduce when a highly skewed grid was used. The second test simulation was a two-dimensional unsteady-state simulation; here, the mesh skewness and mesh size were varied. To evaluate the effect of the manner in which cross-derivative terms were treated, simulation with the FI-FDM was also carried out. A comparison between the accuracy and CPU time of the proposed model and the FI-FDM showed that the method of treating the cross-derivative terms did not affect the numerical accuracy in this test case. However, it was found that the FI-FDM was faster than the proposed model with a highly skewed mesh, whereas the proposed model was faster than the FI-FDM with a non-highly skewed mesh. If extremely complex domains have to be used, FI-FDM might be more preferable than the proposed model. The FEM with a triangular unstructured grid was also used in the second, third, and fourth test simulations in order to compare the performance of the FEM with that of the proposed model. In the second test simulation, the proposed model converged more rapidly than the FEM; however both models gave the same order of relative errors when their nodes were placed at the same positions. Two-dimensional, transient, variably saturated flow with a trapezoid flow domain was simulated in the third test simulation. The proposed model performed well with the non-rectangular flow domain and again converged more rapidly than the FEM, with both models showing the same order of relative error. The fourth test simulation was a three-dimensional runoff simulation for a curvilinearly shaped slope. The proposed model successfully described the three-dimensional subsurface flow, producing results that agreed well with those obtained using the FEM and the conventional FDM that used higher-resolution orthogonal grids. It was shown that the proposed model was more efficient than the conventional FDM because it could represent a curvililear shape using a comparatively low-resolution grid. As in the second and third test simulations, the proposed model converged more rapidly than the finite element model did. Overall, for the test simulations, the proposed model exhibited smaller mass balance error, the same order of relative error, and a faster convergence speed than the FEM.

Chapter 4

Comparison of iteration methods for saturated–unsaturated flow model

4.1 Introduction

Because the saturated–unsaturated flow equation system is highly nonlinear, implicit temporal discretization and iterative procedures are generally needed for numerical stability. The Picard and Newton iteration methods have been widely used in the saturated–unsaturated flow model. *Paniconi et al.* [65], *Paniconi & Putti* [66], and *Lehmann & Ackerer* [52] well investigated the Picard and Newton iteration methods for FEM in 1D, 2D, and 3D problems. However, the Picard and Newton methods of FDM implementing 7-point stencil strategy are expected to show different performances as compared to those of previous studies. The coordinate-transformed equation requires a 19-point stencil but it is extremely complex to evaluate all the terms in implicit manner. In practical aspect, it is an usual choice that the cross-derivative terms are evaluated in explicit manner and the other normal-derivative terms are evaluated in implicit manner, which is called as 7-point stencil strategy in this thesis.

There is a iteration method that can consider 19-point stencil without complexity of coding; the Newton-Krylov method does not require forming 19-stencil directly. Instead of it, it is required to calculate the matrix-vector product which can be approximated by

taking differences of the original nonlinear function in the procedure of Newton-Krylov method. It is a major advantage of the Newton-Krylov method that all the terms can be evaluated in implicit manner without forming 19-stencil matrix, which may give higher convergence speed and allow larger time step than the scheme evaluating the terms in partially implicit manner. But there is additional cost of the Newton-Krylov method for calculating the matrix-vector product at each Krylov iteration step. Hence, it is considered that there is trade-off between the Newton-Krylov method and normal Newton iteration schemes using partially implicit manner. In this Chapter, we compare three iteration methods: the Picard method with 7-point stencil, the Newton method with 7-point stencil, and Newton-Krylov method.

The remainder of this paper is organized as follows. In Section 4.2, we discuss the three different iteration methods in detail. Next, the spatial discretization and treating of cross-derivative terms is described in Section 4.3. Section 4.4 discusses three test simulations, which were carried out to compare the performance of three iteration methods. The first test is a steady-state pumping well problem to investigate the convergence behavior of the iteration methods. The second test is an unsteady-state infiltration problem to evaluate the effect of mesh skewness and anisotropy on the performance of the iteration methods. The third test is a rainfall-runoff simulation for a curvilinear slope with different anisotropies.

4.2 Iteration schemes

As shown in Chapter 3, the backward Euler approximation of coordinate-transformed Richards' equation is written as

$$\frac{1}{J}\frac{\theta^{n+1}-\theta^{n}}{\Delta t}-\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1}\right\}+\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^{p}K^{n+1}\right)=0. \qquad (4.1)$$

4.2.1 Newton method

In the Newton method, all nonlinearities of equation are taken into consideration as follows:

$$f(\psi^{n+1,m}) - \left.\frac{df(\psi)}{d\psi}\right|^{n+1,m} \delta\psi^m = 0, \tag{4.2}$$

where $f(\psi)$ is Richards' equation, superscripts m denotes the iteration level and $\delta\psi^m$ ($= \psi^{n+1,m+1} - \psi^{n+1,m}$) is updating value. Nonlinearities of Richards equation is involved with the term of $\theta(\psi)$ and $K(\psi)$. By applying Eq.(4.2) to Eq.(4.1), we have

$$\begin{aligned}
&\frac{1}{J}\frac{\theta^{n+1,m} - \theta^n}{\Delta t} - \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m}\right\} - \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^pK^{n+1,m}\right) \quad (4.3)\\
&+\frac{C^{n+1,m}}{J\Delta t}\delta\psi^m - \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\frac{\partial(\delta\psi^m)}{\partial\xi_q}\right\}\\
&-\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}\,\delta\psi^m\left.\frac{dK}{d\psi}\right|^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m}\right\} - \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^p\,\delta\psi^m\left.\frac{dK}{d\psi}\right|^{n+1,m}\right) = 0.
\end{aligned}$$

4.2.2 Picard method

Details of the Picard linearization are described in Chapter 2. The nonlinearities related to $K(\psi)$ are not considered in the Picard iteration method:

$$\begin{aligned}
&\frac{1}{J}\frac{\theta^{n+1,m} - \theta^n}{\Delta t} - \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m}\right\} - \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^pK^{n+1,m}\right)\\
&+\frac{C^{n+1,m}}{J\Delta t}\delta\psi^m - \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\frac{\partial(\delta\psi^m)}{\partial\xi_q}\right\} = 0. \quad (4.4)
\end{aligned}$$

The $K(\psi)$ terms are estimated in the previous iteration step and their nonlinearities are not considered in an implicit manner. On comparing Eqs. (4.3) and (4.4), it is found that the Picard method can be viewed as an approximation of the Newton method by neglecting the terms that involve nonlinearities of $K(\psi)$, i.e., the 6th and 7th terms on the left side of Eq. (4.3). It is known that the Newton scheme converges quadratically whereas the Picard scheme converges linearly.

4.2.3 Newton-Krylov method

The Newton-Krylov method is a combination of the Newton iteration method for the linearization of nonlinear systems and the Krylov subspace method for solving linear simultaneous equation. These two methods are related to each other by the Jacobian-vector product, which can be approximated from the original nonlinear equation without estimating the real Jacobian matrix. *Knoll & Keyes* [43] extensively reviewed the Newton-Krylov method and its application cases.

In the Newton-Krylov method, linearization is the same as that in the pure Newton method, given in Eq. (4.2). The linearized equation is solved by the Krylov subspace method. An initial residual vector, r_0, is defined as

$$r_0 = f(\psi^{n+1,m}) - A\delta\psi^0, \tag{4.5}$$

where A is a matrix of $df(\psi)/d\psi$, $\delta\psi^0$ is an initial guess of $\delta\psi$, r_0 is an initial residual vector, and the time index n and iteration index m are dropped because the Krylov iteration is performed at fixed n and m. Hence, based on the Krylov subspace method, the solution of Eq. (4.2) can be expressed as

$$\delta\psi^l = \sum_{k=0}^{l-1} \beta^k (A)^k r_0, \tag{4.6}$$

here the superscripted l denotes the Krylov iteration step and β refers to the scalar values determined to minimize the residual vector in the Generalized Minimal RESidual (GMRES) algorithm. It is found that Eq. (4.6) requires the matrix A only in the form of a matrix-vector product, which can be approximated by taking differences of the original nonlinear function, as suggested by [8, 11].

$$Av \approx \frac{f(\psi + \epsilon v) - f(\psi)}{\epsilon}, \tag{4.7}$$

where v is a vector of $(A)^{k-1} r_0$ and ϵ is a scalar value, much smaller than the scalar elements of v. Eq. (4.7) is simply a rewritten form of the first-order Taylor series expansion approximation of $f(\psi + \epsilon v)$. The error in this approximation is proportional to perturbation ϵ; there are various options for selecting the perturbation parameter

[43]. The following formulation was used in this study, by referring to [8]

$$\epsilon = \frac{b}{||\psi||_2}\max[|\psi \cdot v|, ||v||_1]sign(\psi \cdot v) \tag{4.8}$$

where b is a constant whose magnitude is within a few orders of the magnitude of the square root of the machine roundoff, typically 10^{-6} for 64-bit double precision.

The difference between the pure Newton method and the Newton-Krylov method is that in the pure Newton method, the matrix A is formed as in Eq. (4.2). In the Newton-Krylov method, the approximation of the matrix-vector product, Eq. (4.7), is used for the Krylov iteration rather than forming A. It is a significant advantage of the Newton-Krylov method that A is not required to be formed. However, it should also be pointed out that the Newton-Krylov method also involves an additional cost of calculating the approximation of Eq. (4.7) at each Krylov iteration, as compared to the pure Newton iteration method, in which the cost for the calculation of Av is negligibly small.

4.3 Finite-difference discretization

Finite-difference approximation of Eq.(4.3) is as follows:

$$\begin{aligned}
&\frac{1}{J_{i,j,k}}\frac{\theta^{n+1,m}_{i,j,k} - \theta^{n}_{i,j,k}}{\Delta t} \\
&-\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial \xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial \psi}{\partial \xi_q}\right|^{n+1,m}\right\}_{i,j,k} - \sum_{p=1}^{3}\frac{\partial}{\partial \xi_p}\left(H^p K^{n+1,m}\right)_{i,j,k} \\
&+\frac{C^{n+1,m}_{i,j,k}}{J_{i,j,k}\Delta t}\delta\psi^{n+1,m}_{i,j,k} - \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial \xi_p}\left\{G^{p,q}K^{n+1,m}\frac{\partial(\delta\psi^m)}{\partial \xi_q}\right\}_{i,j,k} \\
&\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial \xi_p}\left\{G^{p,q}\left.\frac{\partial \psi}{\partial \xi_q}\right|^{n+1,m}\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^m\right\}_{i,j,k} \\
&-\sum_{p=1}^{3}\frac{\partial}{\partial \xi_p}\left(H^p\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^m\right)_{i,j,k} = 0,
\end{aligned} \tag{4.9}$$

where subscripts i, j, and k denote the spatial coordinates in the ξ_1, ξ_2, and ξ_3 axes, respectively, in the computational space. This discretization requires a 19-point stencil to solve the equation. Because the formation of the 5th and 6th terms make

the scheme extremely complex for programming, the usual option is to treat the cross-derivative terms in an explicit manner, which requires a 7-point stencil instead of a 19-point stencil, as follows:

$$
\begin{aligned}
&\frac{1}{J_{i,j,k}}\frac{\theta_{i,j,k}^{n+1,m}-\theta_{i,j,k}^{n}}{\Delta t} \\
&-\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m}\right\}_{i,j,k}-\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^{p}K^{n+1,m}\right)_{i,j,k} \\
&+\frac{C_{i,j,k}^{n+1,m}}{J_{i,j,k}\Delta t}\delta\psi_{i,j,k}^{n+1,m}-\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,p}K^{n+1,m}\frac{\partial(\delta\psi^{m})}{\partial\xi_p}\right\}_{i,j,k} \\
&-\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,p}\left.\frac{\partial\psi}{\partial\xi_p}\right|^{n+1,m}\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^{m}\right\}_{i,j,k} \\
&-\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^{p}\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^{m}\right)_{i,j,k}=0,
\end{aligned}
\tag{4.10}
$$

where the second term on the left-hand side of equation is evaluated as

$$
\begin{aligned}
& \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m}\right\}_{i,j,k} \qquad (4.11)\\
= \;& \Big\{G^{1,1}_{i+1/2,j,k}K^{n+1,m}_{i+1/2,j,k}(\psi^{n+1,m}_{i+1,j,k}-\psi^{n+1,m}_{i,j,k})\\
& - G^{1,1}_{i-1/2,j,k}K^{n+1,m}_{i-1/2,j,k}(\psi^{n+1,m}_{i,j,k}-\psi^{n+1,m}_{i-1,j,k})\Big\}\\
+ \;& \Big\{G^{2,2}_{i,j+1/2,k}K^{n+1,m}_{i,j+1/2,k}(\psi^{n+1,m}_{i,j+1,k}-\psi^{n+1,m}_{i,j,k})\\
& -G^{2,2}_{i,j-1/2,k}K^{n+1,m}_{i,j-1/2,k}(\psi^{n+1,m}_{i,j,k}-\psi^{n+1,m}_{i,j-1,k})\Big\}\\
+ \;& \Big\{G^{3,3}_{i,j,k+1/2}K^{n+1,m}_{i,j,k+1/2}(\psi^{n+1,m}_{i,j,k+1}-\psi^{n+1,m}_{i,j,k})\\
& -G^{3,3}_{i,j,k-1/2}K^{n+1,m}_{i,j,k-1/2}(\psi^{n+1,m}_{i,j,k}-\psi^{n+1,m}_{i,j,k-1})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{1,2}_{i+1/2,j,k}K^{n+1,m}_{i+1/2,j,k}(\psi^{n+1,m}_{i+1/2,j+1,k}-\psi^{n+1,m}_{i+1/2,j-1,k})\\
& - G^{1,2}_{i-1/2,j,k}K^{n+1,m}_{i-1/2,j,k}(\psi^{n+1,m}_{i-1/2,j+1,k}-\psi^{n+1,m}_{i-1/2,j-1,k})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{1,3}_{i+1/2,j,k}K^{n+1,m}_{i+1/2,j,k}(\psi^{n+1,m}_{i+1/2,j,k+1}-\psi^{n+1,m}_{i+1/2,j,k-1})\\
& - G^{1,3}_{i-1/2,j,k}K^{n+1,m}_{i-1/2,j,k}(\psi^{n+1,m}_{i-1/2,j,k+1}-\psi^{n+1,m}_{i-1/2,j,k-1})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{2,1}_{i,j+1/2,k}K^{n+1,m}_{i,j+1/2,k}(\psi^{n+1,m}_{i+1,j+1/2,k}-\psi^{n+1,m}_{i-1,j+1/2,k})\\
& - G^{2,1}_{i,j-1/2,k}K^{n+1,m}_{i,j-1/2,k}(\psi^{n+1,m}_{i+1,j-1/2,k}-\psi^{n+1,m}_{i-1,j-1/2,k})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{2,3}_{i,j+1/2,k}K^{n+1,m}_{i,j+1/2,k}(\psi^{n+1,m}_{i,j+1/2,k+1}-\psi^{n+1,m}_{i,j+1/2,k-1})\\
& - G^{2,3}_{i,j-1/2,k}K^{n+1,m}_{i,j-1/2,k}(\psi^{n+1,m}_{i,j-1/2,k+1}-\psi^{n+1,m}_{i,j-1/2,k-1})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{3,1}_{i,j,k+1/2}K^{n+1,m}_{i,j,k+1/2}(\psi^{n+1,m}_{i+1,j,k+1/2}-\psi^{n+1,m}_{i-1,j,k+1/2})\\
& -G^{3,1}_{i,j,k-1/2}K^{n+1,m}_{i,j,k-1/2}(\psi^{n+1,m}_{i+1,j,k-1/2}-\psi^{n+1,m}_{i-1,j,k-1/2})\Big\}\\
+ \;& \frac{1}{2}\Big\{G^{3,2}_{i,j,k+1/2}K^{n+1,m}_{i,j,k+1/2}(\psi^{n+1,m}_{i,j+1,k+1/2}-\psi^{n+1,m}_{i,j-1,k+1/2})\\
& -G^{3,2}_{i,j,k-1/2}K^{n+1,m}_{i,j,k-1/2}(\psi^{n+1,m}_{i,j+1,k-1/2}-\psi^{n+1,m}_{i,j-1,k-1/2})\Big\},
\end{aligned}
$$

the third term on the left-hand side of equation is evaluated as

$$
\begin{aligned}
& \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^pK^{n+1,m}\right)_{i,j,k} \qquad (4.12)\\
= \;& \left(H^1K^{n+1,m}\right)_{i+1/2,j,k}-\left(H^1K^{n+1,m}\right)_{i-1/2,j,k}\\
& +\left(H^2K^{n+1,m}\right)_{i,j+1/2,k}-\left(H^2K^{n+1,m}\right)_{i,j-1/2,k}\\
& +\left(H^3K^{n+1,m}\right)_{i,j,k+1/2}-\left(H^3K^{n+1,m}\right)_{i,j,k-1/2},
\end{aligned}
$$

the fifth term on the left-hand side of equation is evaluated as

$$
\begin{aligned}
& \sum_{p=1}^{3} \frac{\partial}{\partial \xi_p} \left\{ G^{p,p} K^{n+1,m} \frac{\partial \delta\psi^m}{\partial \xi_p} \right\}_{i,j,k} \qquad (4.13)\\
= \quad & \left\{ G^{1,1}_{i+1/2,j,k} K^{n+1,m}_{i+1/2,j,k} (\delta\psi^m_{i+1,j,k} - \delta\psi^m_{i,j,k}) \right.\\
& \left. - G^{1,1}_{i-1/2,j,k} K^{n+1,m}_{i-1/2,j,k} (\delta\psi^m_{i,j,k} - \delta\psi^m_{i-1,j,k}) \right\}\\
+ \quad & \left\{ G^{2,2}_{i,j+1/2,k} K^{n+1,m}_{i,j+1/2,k} (\delta\psi^m_{i,j+1,k} - \delta\psi^m_{i,j,k}) \right.\\
& \left. - G^{2,2}_{i,j-1/2,k} K^{n+1,m}_{i,j-1/2,k} (\delta\psi^m_{i,j,k} - \delta\psi^m_{i,j-1,k}) \right\}\\
+ \quad & \left\{ G^{3,3}_{i,j,k+1/2} K^{n+1,m}_{i,j,k+1/2} (\delta\psi^m_{i,j,k+1} - \delta\psi^m_{i,j,k}) \right.\\
& \left. - G^{3,3}_{i,j,k-1/2} K^{n+1,m}_{i,j,k-1/2} (\delta\psi^m_{i,j,k} - \delta\psi^m_{i,j,k-1}) \right\},
\end{aligned}
$$

the sixth term on the left-hand side of equation is evaluated as

$$
\begin{aligned}
& \sum_{p=1}^{3} \frac{\partial}{\partial \xi_p} \left\{ G^{p,p} \left.\frac{\partial \psi}{\partial \xi_p}\right|^{n+1,m} \left.\frac{dK}{d\psi}\right|^{n+1,m} \delta\psi^m \right\}_{i,j,k} \qquad (4.14)\\
= \quad & \left\{ G^{1,1}_{i+1/2,j,k} (\psi^{n+1,m}_{i+1,j,k} - \psi^{n+1,m}_{i,j,k}) \left(\left.\frac{dK}{d\psi}\right|^{n+1,m} \delta\psi^m \right)_{i+1/2,j,k} \right.\\
& \left. - G^{1,1}_{i-1/2,j,k} (\psi^{n+1,m}_{i,j,k} - \psi^{n+1,m}_{i-1,j,k}) \left(\left.\frac{dK}{d\psi}\right|^{n+1,m} \delta\psi^m \right)_{i-1/2,j,k} \right\}\\
& \left\{ G^{2,2}_{i,j+1/2,k} (\psi^{n+1,m}_{i,j+1,k} - \psi^{n+1,m}_{i,j,k}) \left(\left.\frac{dK}{d\psi}\right|^{n+1,m} \delta\psi^m \right)_{i,j+1/2,k} \right.\\
& \left. - G^{2,2}_{i,j-1/2,k} (\psi^{n+1,m}_{i,j,k} - \psi^{n+1,m}_{i,j-1,k}) \left(\left.\frac{dK}{d\psi}\right|^{n+1,m} \delta\psi^m \right)_{i,j-1/2,k} \right\}\\
& \left\{ G^{3,3}_{i,j,k+1/2} (\psi^{n+1,m}_{i,j,k+1} - \psi^{n+1,m}_{i,j,k}) \left(\left.\frac{dK}{d\psi}\right|^{n+1,m} \delta\psi^m \right)_{i,j,k+1/2} \right.\\
& \left. - G^{2,2}_{i,j,k-1/2} (\psi^{n+1,m}_{i,j,k} - \psi^{n+1,m}_{i,j,k-1}) \left(\left.\frac{dK}{d\psi}\right|^{n+1,m} \delta\psi^m \right)_{i,j,k-1/2} \right\},
\end{aligned}
$$

and the seventh term on the left-hand side of equation is evaluated as

$$\begin{aligned}
&\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^p\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^m\right)_{i,j,k} \\
=\;&\left(H^1\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^m\right)_{i+1/2,j,k}-\left(H^1\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^m\right)_{i-1/2,j,k} \\
&+\left(H^2\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^m\right)_{i,j+1/2,k}-\left(H^2\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^m\right)_{i,j-1/2,k} \\
&+\left(H^3\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^m\right)_{i,j,k+1/2}-\left(H^3\left.\frac{dK}{d\psi}\right|^{n+1,m}\delta\psi^m\right)_{i,j,k-1/2},
\end{aligned} \tag{4.15}$$

where the values of K, ψ, $\delta\psi$, G, and H of the boundary between the adjacent nodes used the arithmetic mean as follows:

$$\begin{aligned}
V_{i\pm1/2,j,k} &= \frac{1}{2}\left(V_{i\pm1,j,k}+V_{i,j,k}\right), \quad V_{i,j\pm1/2,k}=\frac{1}{2}\left(V_{i,j\pm1,k}+V_{i,j,k}\right), \\
V_{i,j,k\pm1/2} &= \frac{1}{2}\left(V_{i,j,k\pm1}+V_{i,j,k}\right).
\end{aligned} \tag{4.16}$$

where V represents the values of K, ψ, $\delta\psi$, G, and H.

The iteration process of Eq (4.10) continues until $\delta\psi$ becomes less than the user-specified tolerances for all grids ($|\delta\psi^m| \le \delta_\psi$), where δ_ψ denotes the convergence tolerance. In the test simulations of this study, $\delta_\psi = 10^{-4}$ m is used for Tests 1 and 3 case, which are transient flow simulation tests, and $\delta_\psi = 10^{-8}$ m is used for Test 2, which is a steady-state flow simulation test.

The equation linearized by the Picard method is discretized partially implicitly as follows:

$$\begin{aligned}
&\frac{1}{J_{i,j,k}}\frac{\theta^{n+1,m}_{i,j,k}-\theta^{n}_{i,j,k}}{\Delta t} \\
&-\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m}\right\}_{i,j,k}-\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^pK^{n+1,m}\right)_{i,j,k} \\
&+\frac{C^{n+1,m}_{i,j,k}}{J_{i,j,k}\Delta t}\delta\psi^{n+1,m}_{i,j,k}-\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,p}K^{n+1,m}\frac{\partial(\delta\psi^m)}{\partial\xi_p}\right\}_{i,j,k}=0, j
\end{aligned} \tag{4.17}$$

Eq. (4.17) was used to successfully simulate 2D and 3D saturated–unsaturated flows in Chapter 3. The contribution of the cross-derivative terms is negligible when the grid is not highly skewed and the soil properties of the flow domain are isotropic. However, if

the grid is highly skewed or if the soil properties shows a high level of anisotropy, the contribution of the cross-derivative terms to the overall system is large and the evaluation of the the cross-derivative terms might require a small time-step duration. On the other hand, the Newton-Krylov method can treat all terms in an implicit manner without evaluating a 19-stencil, which might allow the scheme to converge faster and take a larger time step duration. However, the Newton-Krylov method involves an additional cost as mentioned previously.

The linear equation systems of the Newton, the Picard, and the Newton-Krylov methods are solved by the GMRES algorithm [79],which is generally used as a Krylov subspace solver for a nonsymmetric linear matrix. For a robust and fast convergence of the GMRES method, an incomplete lower-upper (ILU) factorization [78] is used as a preconditioner. The ILU preconditioner is created by decomposition of the linear matrix. However, the purpose of the Newton-Krylov method is to avoid the direct formation of the linear system matrix; the ILU preconditioner of the Newton-Krylov method is created by decomposition of the 7-point stencil matrix, which is the same matrix used in the pure Newton method.

The Newton method is sensitive to the initial guess and is inefficient in particular steady-state problems. To improve the robustness of the Newton method, we implemented the line-search approach. In this approach, the update vector, $\delta\psi$, is assumed to indicate the correct direction. Thus, a line search uses a scalar ($s \leq 1$) as follows:

$$\psi^{n+1,m+1} = \psi^{n+1,m} + s\delta\psi^{m}. \tag{4.18}$$

Further, s decreases unless the nonlinear residual is less than the previous value:

$$f(\psi^{n+1,m} + s\delta\psi^{m}) < f(\psi^{n+1,m}). \tag{4.19}$$

The simplest approach is that s is decreased as $s = 1, 0.5, 0.25, ...$ until Eq. (4.19) is satisfied; this approach has been used by [39]. In our experience, this approach is simple but very effective in enhancing the convergence behavior of the numerical schemes.

4.4 Numerical simulations

Three iteration methods—the Picard method using a 7-point stencil, the Newton method using a 7-point stencil and the Newton-Krylov method—are evaluated throughout the three test simulations. The first test simulation is a steady-state pumping well problem, used to evaluate the convergence behavior of the three methods. The second test simulation is a simple infiltration problem which involves variation of mesh skewness and anisotropy. In this test simulation, the effect of mesh skewness and anisotropy on the model performance and robustness are assessed. The third test simulation is a rainfall-runoff simulation for a curvilinear slope. This test simulation is used to evaluate the performances of the three iteration methods for a curvilinear shape domain.

Van Genuchten [91]'s equation for the soil water retention curve and *Mualem* [61]'s equation for the unsaturated hydraulic conductivity function were used. Detail of these models are described in as Section 2.3. The mass balance error and the relative error are also defined in Section 2.3. In test simulations of this chapter, the soil properties listed in Table 2.1 are used. These values represent average values for the selected soil water retention and hydraulic conductivity parameters for major soil textural groups. The time-step duration are adjusted automatically as Section 2.3. The parameter of time-step duration control are set as $\mathrm{C}_m = 1.2$, $\mathrm{C}_r = 0.8$, $\mathrm{C}_b = 0.5$, $\mathrm{N}_m = 6$, $\mathrm{N}_r = 10$ and $\mathrm{N}_b = 20$.

It is expected that if the cross-derivative terms are dominant in the overall system, the 7-point stencil Newton and Picard methods may not easily converge in the iterative procedure. Hence, grid skewness and anisotropy are expected to affect the performances of the iteration methods because they determine the value of $G^{p,q}$, which affect the degree of dominance of the cross-derivative terms. To assess the degree of dominance of the cross-derivative terms, we defined a ratio of the cross-derivative terms (RCT) using the values of $G^{p,q}$ as follows:

$$\mathrm{RCT} = 1 - \frac{\sum_p^3 |G^{p,p}|}{\sum_p^3 \sum_q^3 |G^{p,q}|} \tag{4.20}$$

where RCT is defined at internodes such as $(i+1/2, j, k)$. When the degree of dominance of the cross-derivative terms is high, the value of RCT increases. For example, if the grid is orthogonal, RCT is 0, and if the degree of grid skewness and anisotropy is high,

TABLE 4.1: Test simulation 1 with line search method, total iteration number and CPU time.

Scheme	Total iteration number			CPU time (s)		
	L = 10	L = 20	L = 30	L = 10	L = 20	L = 30
Picard	342	151	83	76	30	15
Newton	36	25	21	8	5	3
Newton-Krylov	22	13	11	34	17	14

the value of RCT is close to 1. Maximum RCT (MRCT) refer to the maximum value of RCT at all places where RCTs are defined.

The models using the three iteration methods have first-order temporal and second-order spatial discretization precisions. The precision of the models was not significantly affected by the type of iteration method employed. Therefore, this paper compares only the convergence behavior and CPU time required by the three iteration methods.

4.4.1 Test 1: pumping well, steady state

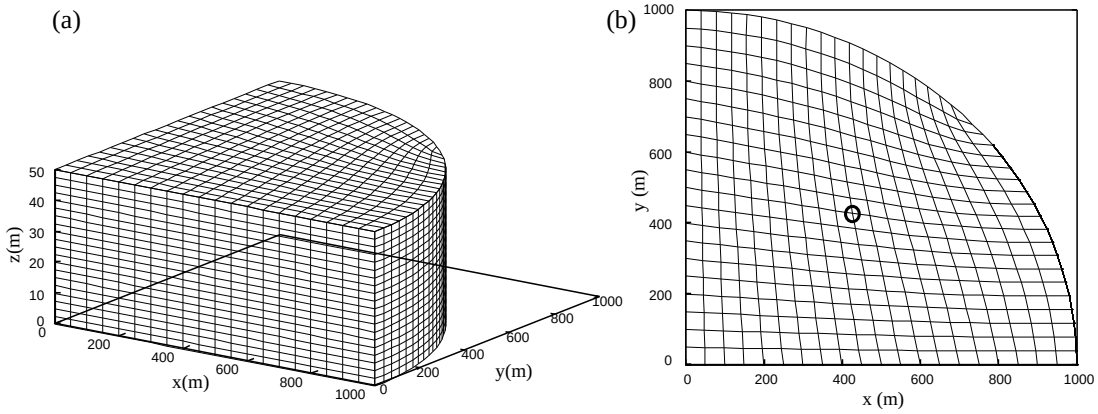

FIGURE 4.1: Flow domain and grid used in Test 1. (a) perspective view; (b) plane view and circle represents the place of pumping well.

The performances of three iteration methods for steady state problem are investigated in this simulation. Curvilinear domains with $21 \times 21 \times 21$ mesh are considered, as shown in Fig. 4.1. Further, the loam soil properties listed in Table 2.1 are considered. The initial condition is $\psi_0(z) = 45$ - z (m). All side boundaries are Dirichlet boundaries as $\psi_b(z) = 45$ - z (m), and there are no flux boundaries for the top and bottom. A pumping well is placed at $(x = 626, y = 626)$. To evaluate the performances of the iteration methods under different problem difficulties, the water levels at the well are

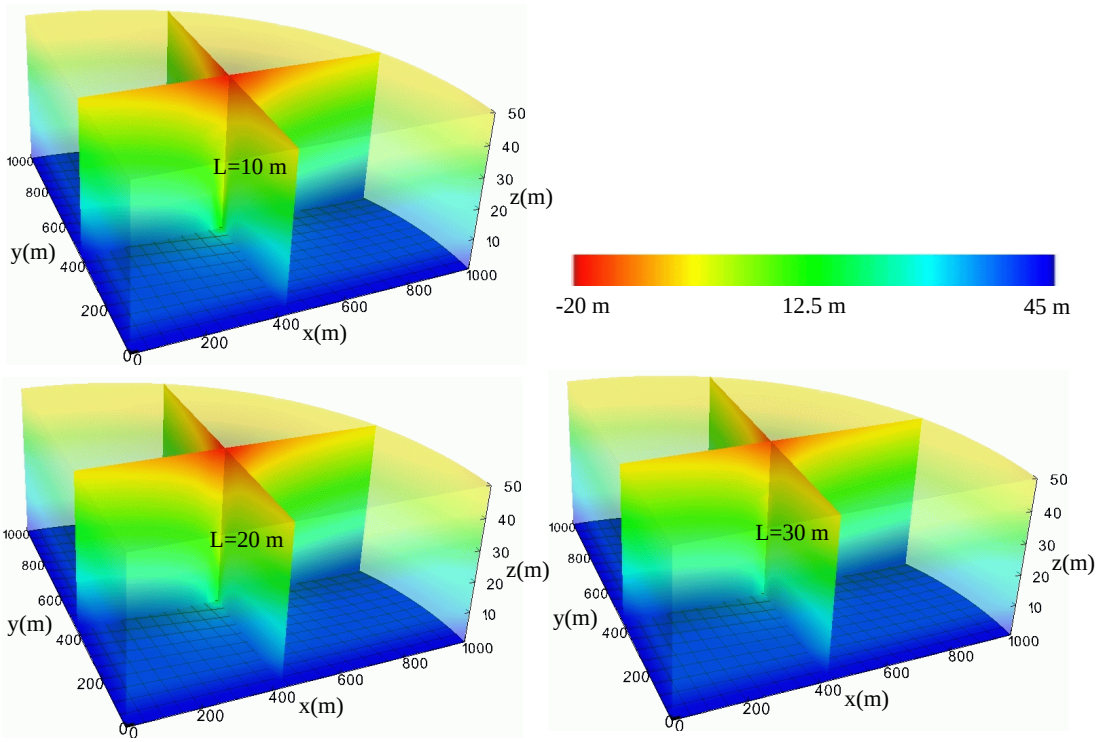

FIGURE 4.2: Pressure head results of Test 2 performed by the Newton method. A cross sectional distribution for $x = 626$, $y = 626$ and $z = 0$ m.

varied as $L = 10$, 20, and 30 (m). When $L = 10$, the problem is comparatively difficult because the gradient of the pressure head distribution is large. In contrast, the problem is easy in the case of $L = 30$. The well nodes are set as $\psi_w(z) = L - z$ (m) in $z \leq L$ and at the seepage face boundary in $z > L$.

To confirm the effectiveness of the line-search approach in the case of steady-state problems, test simulations were conducted with and without employing this approach. Without the line-search approach, the three iteration methods failed to converge for all the cases. On the other hand, with the line-search approach, the three iteration methods converged stably for all cases. This proved that the line-search method effectively improved the numerical stability of the three iteration methods.

Fig. 4.2 describes the pressure head results of the Newton method. The pumping well problems with different well conditions were described. As previously mentioned, the precision was not significantly affected by the type of iteration method selected. The other two methods resulted in almost the same pressure head distribution shown in Fig. 4.2. Table 4.1 lists the total number of iterations taken in the three iteration methods, and Fig. 4.3 shows the convergence behavior of the three iteration methods. As expected,

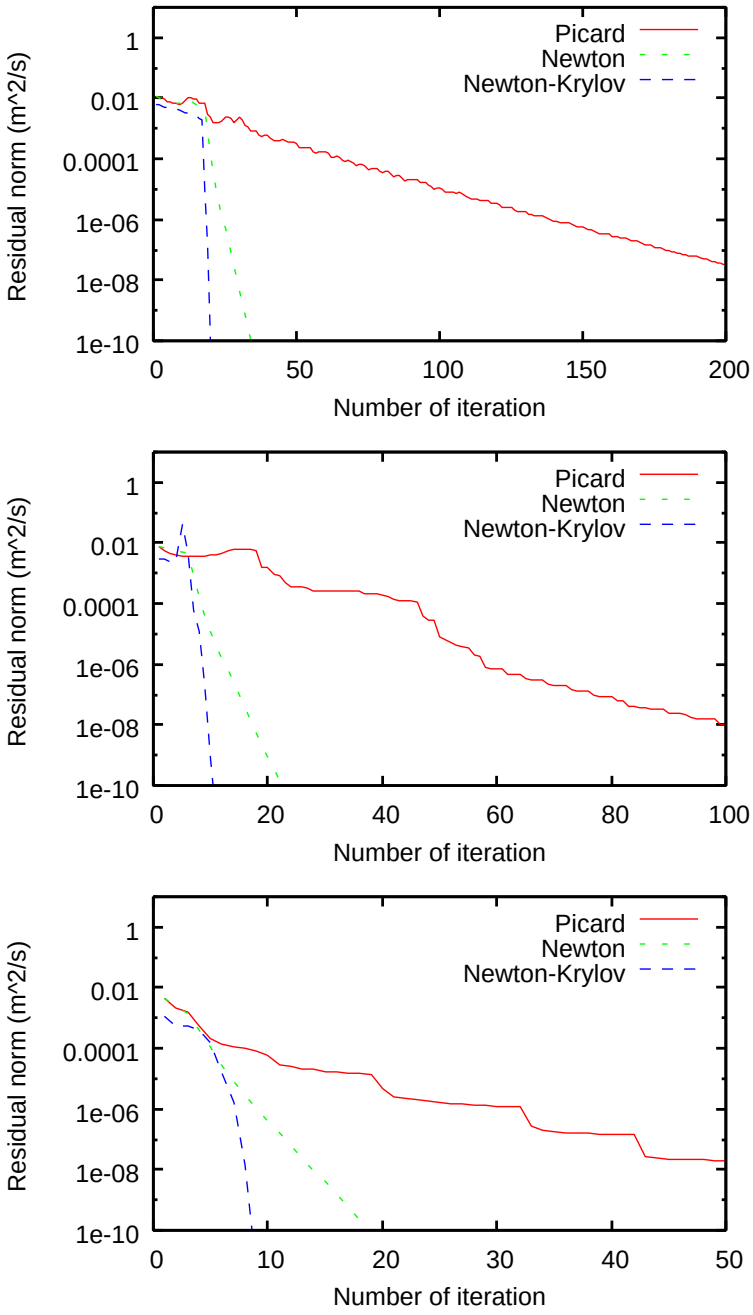

FIGURE 4.3: Convergence behavior of three iteration methods with line search method; L_1 residual norm are plotted.

the Newton-Krylov method converged more quickly than the other two methods. The Newton method converged more quickly than the Picard method but converged slower than the Newton-Krylov method. In the first few iterations, the Newton and Picard methods showed similar convergence behaviors; however, after the first few iterations, the Newton method converged more drastically than the Picard method. The convergence behaviors of all the iteration methods were similar, while the problem difficulties varied. Table 4.1 also lists the CPU time consumed by the three iteration methods. Even though the Newton-Krylov method converged most quickly, the Newton method consumed the least CPU time in this test simulation. Average CPU costs per iteration of the Newton-Krylov and Newton method were 1.38 and 0.19 sec, respectively. Compared to the Newton method, the Newton-Krylov method requires additional cost for solving the matrix-vector product in each Krylov iteration. In this test simulation, 70∼90 Krylov iterations were required to be solved for one outer Newton iteration. This additional calculation consumed more than 80 % of the CPU resources in the Newton-Krylov iteration method. In conclusion, the Newton method consumed the least CPU times in this test simulation and it is confirmed that the Picard method was not suitable iteration method for steady-state problems.

4.4.2 Test 2: infiltration problem with different skewnesses and anisotropies

The relation between the performance of the iteration method and the degree of dominance of the cross-derivative terms is investigated in this test simulation. Simple infiltration problems are considered in a 0.2m×0.2m×1m rectangular domain. The sand and loam soil properties listed in Table 2.1 are used. The top and side boundaries are no-flow boundaries, and the bottom boundary is $\psi_{bot} = 0.5$ m. The initial condition is $\psi_{init} = -1$ m. The simulation period is one day.

The degree of dominance of the cross-derivative terms is affected by the grid skewness and soil anisotropy. Five different grids having different grid skewnesses, as shown in Fig. 4.4, are used in this simulation. Grid0 is an orthogonal grid with a constant grid size (=2.5 cm); other grids in which nodes are moved as much as 0.25, 0.5, 0.75, and 1.0 cm in randomly determined directions are adjusted. To obtain the different anisotropies, the diagonal entries of $K^e_{r,s}$ are given as 10^a, where a is a randomly generated numbers

TABLE 4.2: Test simulation 2, MRCT.

Grid	Total iteration number					
	$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
Grid0	1.0	1.0	1.0	1.0	1.0	1.0
Grid1	0.67	0.64	0.49	0.33	0.21	0.14
Grid2	0.48	0.40	0.28	0.20	0.13	0.10
Grid3	0.38	0.32	0.24	0.17	0.13	0.10
Grid4	0.29	0.28	0.22	0.17	0.12	0.09

following a normal distribution; its mean value is zero and standard deviation is σ. The non-diagonal entries of $K^e_{r,s}$ are zero. Large values of σ implys a high degree of anisotropy in the flow domain. In this simulation, a is generated when σ is 0.1, 0.2, 0.3, 0.4, and 0.5, as shown in Fig. 4.5. This condition could be considered as unrealistic because there are no correlations between the adjacent values which is not the case for a real flow domain. However, we believe that this difference should not be considered a critical factor when assessing the performance of the iteration methods.

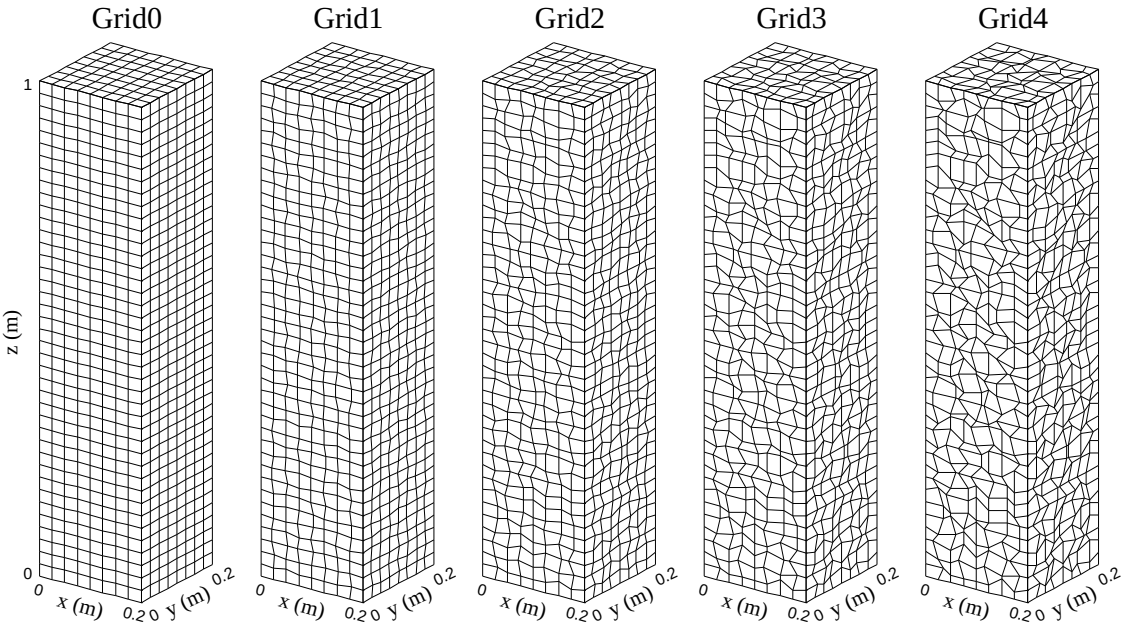

FIGURE 4.4: Different grids used in Test 2.

Table 4.2 lists the values of MRCT. Fig. 4.6 describes the distribution of RCT. It is found that MRCT and the distribution of RCT showed high values when the degree of grid skewness and anisotropy was high. Fig. 4.7 shows the pressure head results for sand at the end of the simulations performed using the Newton method; $\sigma = 0$ for Grid0 and Grid4, σ =0.5 for Grid0, and σ =0.2 for Grid4. The other two methods resulted in

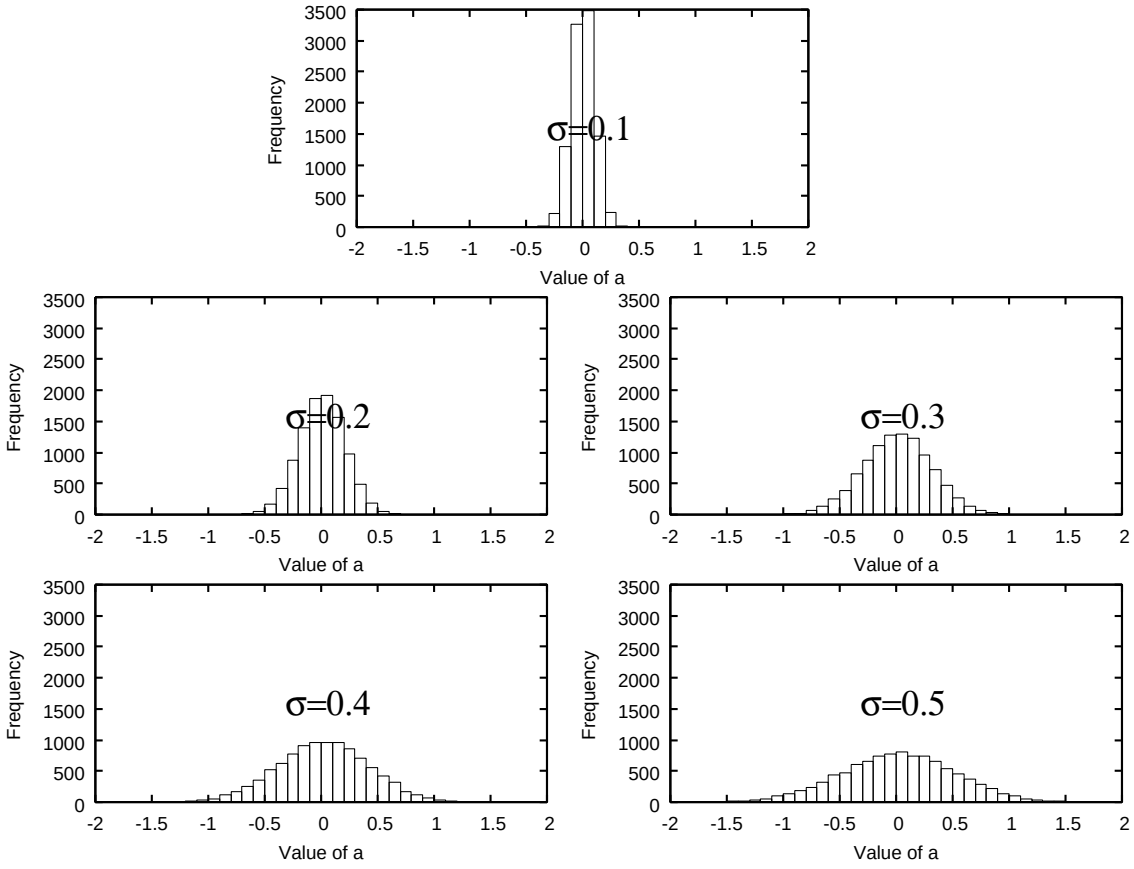

FIGURE 4.5: Frequency of a when σ is 0.1, 0.2, 0.3, 0.4 and 0.5 (10000 samples).

almost the same pressure head distribution, as shown in Fig. 4.7. The results of Grid0 and Grid4 with $\sigma = 0$ were in good agreement even when different grids were used and show a clearly horizontal distribution of the pressure head. On the other hand, in the figures, where $\sigma = 0.5$ and 0.2, there were perturbations in the distribution of the pressure head because of the heterogeneity and anisotropy of the flow domain.

Tables 4.3 and 4.4 show the total number of iterations taken in the three methods when sand and loam are used, respectively. The three methods did not converge for sand when MRCT was over 0.8 (Grids 3 and 4 with $\sigma = 0.3$, 0.4, and 0.5; Grid 2 with $\sigma =$ 0.4 and 0.5; Grid 1 with $\sigma = 0.5$). Infiltration flows into dried soil are challenging issues especially when the nonlinearity of the equation is high, as in the case of sand, because the model should ideally represent a sharp infiltration front. The high level of anisotropy and mesh skewness made the solution of such problems more difficult. Further, none of the three methods could converge within a predetermined tolerance level for such problems even when the Newton-Krylov method was used, although the Newton-Krylov

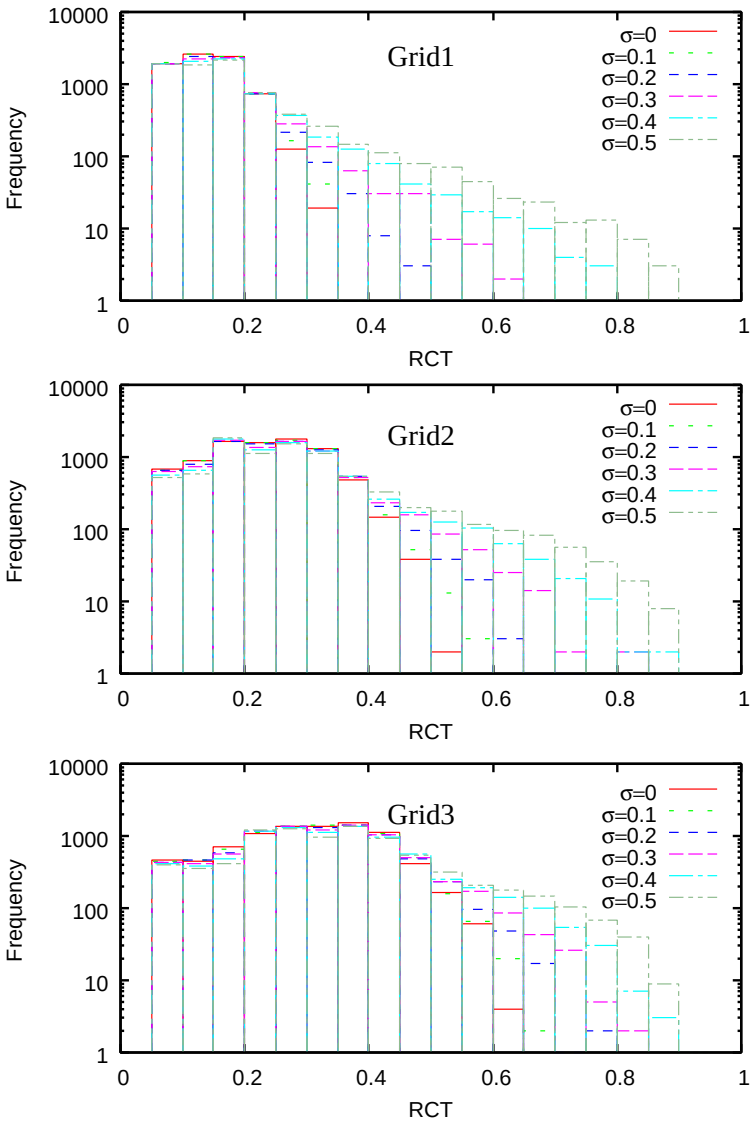

FIGURE 4.6: Test simulation 2, distribution of RCT.

TABLE 4.3: Test simulation 2, total iteration number when sand is used.

Grid	Scheme	Total iteration number					
		$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
Grid0	Picard	1365	1752	2226	2535	2911	3325
	Newton	556	844	1008	1158	1235	1345
	Newton-Krylov	556	844	1008	1158	1235	1345
Grid1	Picard	1706	1926	2319	2640	3063	-
	Newton	742	953	1067	1211	1457	-
	Newton-Krylov	675	901	985	1118	1080	-
Grid2	Picard	2082	2274	2598	2992	-	-
	Newton	1244	1368	1531	1886	-	-
	Newton-Krylov	764	947	1032	1165	-	-
Grid3	Picard	2660	2748	2996	-	-	-
	Newton	1941	2081	2321	-	-	-
	Newton-Krylov	1024	1105	1121	-	-	-
Grid4	Picard	3437	3487	3692	-	-	-
	Newton	3022	3143	3578	-	-	-
	Newton-Krylov	1173	1211	1200	-	-	-

TABLE 4.4: Test simulation 2, total iteration number when loam is used.

Grid	Scheme	Total iteration number					
		$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
Grid0	Picard	1528	7658	9919	14167	16831	18714
	Newton	423	475	503	570	565	653
	Newton-Krylov	423	475	503	532	565	653
Grid1	Picard	3892	8500	9383	13914	16226	20023
	Newton	524	616	582	641	766	1271
	Newton-Krylov	435	486	509	516	549	607
Grid2	Picard	4531	7241	9185	13546	16214	17894
	Newton	898	981	1002	1355	2974	5228
	Newton-Krylov	483	525	554	571	576	617
Grid3	Picard	5664	8271	9305	12949	15209	17989
	Newton	1532	1827	2576	5542	6686	12110
	Newton-Krylov	536	525	525	595	566	673
Grid4	Picard	6293	7386	8892	11888	14912	18414
	Newton	5786	5164	9146	12309	18339	23137
	Newton-Krylov	520	515	596	632	746	693

TABLE 4.5: Test simulation 2, CPU time when sand is used.

Grid	Scheme	CPU time (s)					
		$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
Grid0	Picard	25	40	44	59	59	67
	Newton	10	19	24	23	26	27
	Newton-Krylov	66	121	140	177	176	175
Grid1	Picard	34	40	48	50	62	-
	Newton	15	19	25	27	39	-
	Newton-Krylov	101	142	141	170	162	-
Grid2	Picard	44	51	55	66	-	-
	Newton	25	39	36	50	-	-
	Newton-Krylov	112	144	147	190	-	-
Grid3	Picard	60	53	61	-	-	-
	Newton	53	49	59	-	-	-
	Newton-Krylov	177	191	176	-	-	-
Grid4	Picard	81	80	84	-	-	-
	Newton	115	77	98	-	-	-
	Newton-Krylov	228	224	231	-	-	-

TABLE 4.6: Test simulation 2, CPU time when loam is used.

Grid	Scheme	CPU time (s)					
		$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
Grid0	Picard	28	143	205	282	305	380
	Newton	8	10	11	14	14	18
	Newton-Krylov	52	68	62	78	82	94
Grid1	Picard	74	147	174	259	330	373
	Newton	11	11	11	14	15	30
	Newton-Krylov	60	73	66	60	74	78
Grid2	Picard	79	134	152	236	301	345
	Newton	17	22	24	34	85	203
	Newton-Krylov	75	73	76	74	87	93
Grid3	Picard	111	161	171	239	299	355
	Newton	37	45	53	192	219	458
	Newton-Krylov	82	83	72	95	94	117
Grid4	Picard	109	148	172	226	279	354
	Newton	178	188	265	353	595	733
	Newton-Krylov	87	81	107	96	114	123

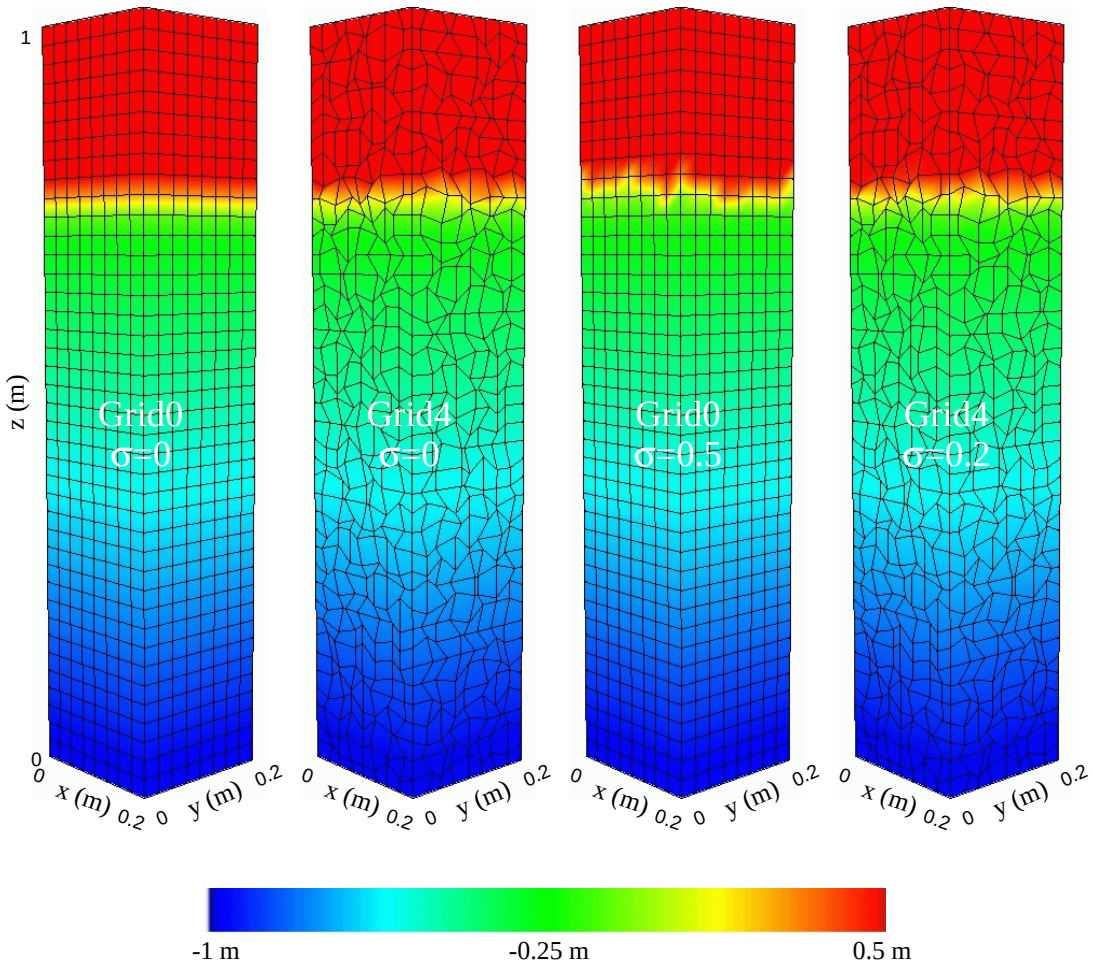

FIGURE 4.7: Test simulation 2, pressure head results of sand performed by the Newton method.

method evaluates all terms in a completely implicit manner and is expected to be the most stable of the three iteration methods.

It is found that the Newton-Krylov method converged the fastest in all cases; further, when Grid0 was used, the Newton and Newton-Krylov methods required the same number of iteration, as expected. If the grid is orthogonal, there are no cross-derivative terms in the system and the Newton and Newton-Krylov methods show the same behavior. It is also noted that when MRCT is less than 0.6, the number of iterations required by Newton and Newton-Krylov methods were similar and that when MRCT is larger than 0.8, the number of iterations required by the Newton method was similar to that required by the Picard method. In other words, if the degree of mesh skewness and anisotropy is low, the convergence behavior of the Newton method becomes similar to that of the Newton-Krylov method, and if the degree of mesh skewness and

anisotropy is high, the convergence behavior of the Newton method becomes similar to that of the Picard method. The reason for the similar behaviors of the Newton and Newton-Krylov methods is clear: These two methods ideally work in the same way when the cross-derivative terms have no effect on the system. However, in the case that the cross-derivative terms have a large effect on the system, it is not so easy to understand the reason for the similar behavior of the Newton and Picard methods. We hypothesized the reason to be the fact that the terms representing the nonlinearity of K, which are neglected in the Picard method, do not contribute to the overall system significantly as compared to the cross-derivative terms; as a result, the convergence behaviors of the Newton and Picard methods become similar.

Tables 4.5 and 4.6 list the CPU time consumed by the three methods. When sand was used, the Newton-Krylov method consumed more CPU time than the Newton method in all cases. However, when loam was used, the Newton-Krylov method consumed less CPU time than the other two methods in the cases where the grid was highly skewed or where both the degrees of grid skewness and anisotropy were high. It might be noted that comparative efficiencies among the three iteration methods depend not only on the grid skewness and anisotropy but also on other simulation conditions, which in this case were soil properties. Let us consider the case where Grid4 was used and $\sigma = 0.2$. The Newton-Krylov method took less CPU time than the Newton method in the case where loam soil was used but vice versa in the case where sand soil was used. Although the Newton-Krylov method showed comparatively good performance when the RCT is high but it did not always show a superior performance as compared to the other two methods. When the mesh skewness or anisotropy was not high, it was found that the Newton method required the least CPU resources in the test simulation. Overall, the Picard method was not an attractive option in almost any case of this simulation.

4.4.3 Test 3: rainfall-runoff simulation for a curvilinear slope

In a curvilinear slope, the performance of the three iteration methods was evaluated. The flow domain and mesh are shown in Fig. 4.8. The sandy loam soil properties, listed in Table 2.1, were considered. The no-flow condition was given for the upper, bottom, and the two side boundaries. The surface boundary and the lower boundary were treated as seepage faces. The initial condition was $\psi_0 = 0.2 - z$ m. Constant rainfall flux into

the domain was considered at the surface nodes. Rainfall intensity was 4 mm/day. The simulation period was 100 days, and the simulation reached the steady state condition at the end of the 100 days. The anisotropy conditions were in the same manner as in Test 2, varying σ as 0.1, 0.2, 0.3, 0.4, and 0.5.

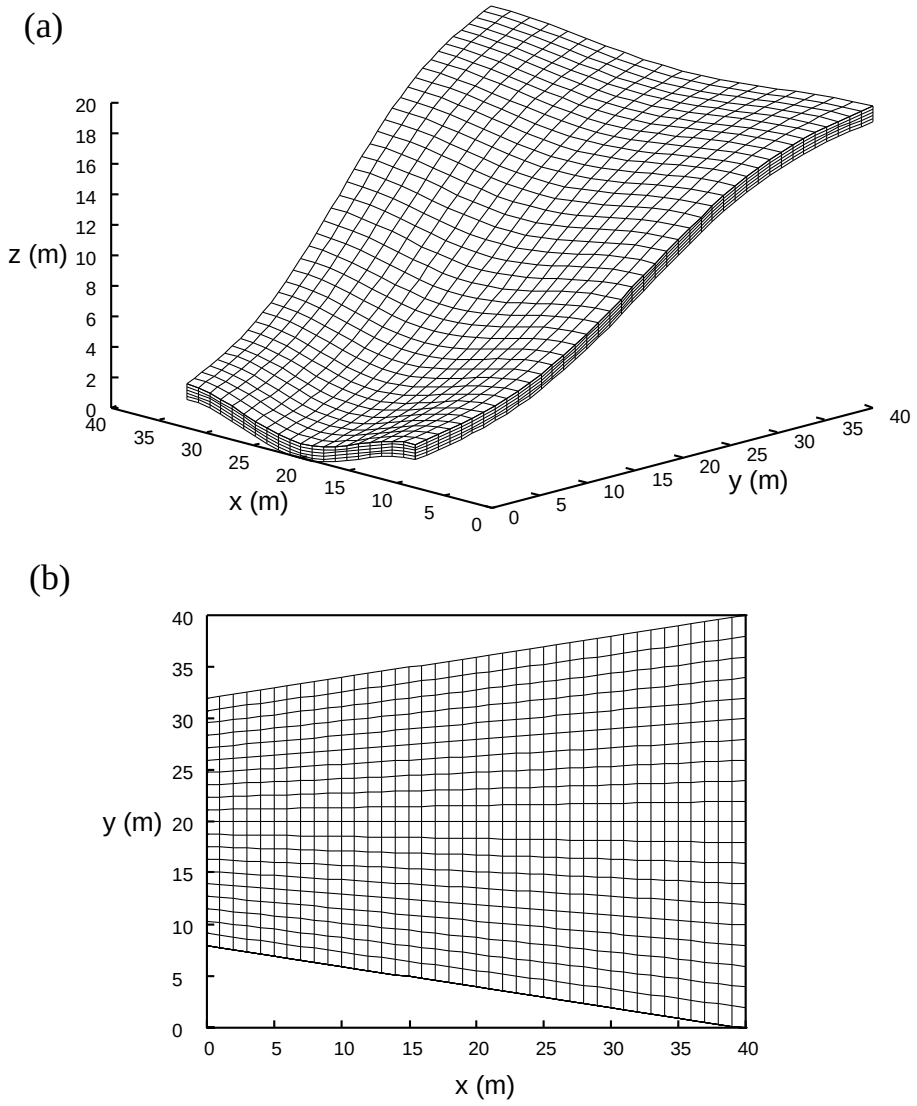

FIGURE 4.8: Curvilinear slope and grid used in Test 3. (a) perspective view; (b) plane view.

Fig. 4.9 describes the pressure head and flux results when σ was 0 and 0.5 at the end of the simulation using the Newton method. The other two methods showed almost the same pressure head distribution results, as shown in Fig. 4.9. The distribution of the pressure head and flux was irregular when σ was 0.5 as compared with when σ was 0. When $\sigma = 0$, a saturated zone developed in a broader area, as shown in Fig. 4.9-(a,c), as compared to the area in which the saturated zone developed when $\sigma = 0.5$

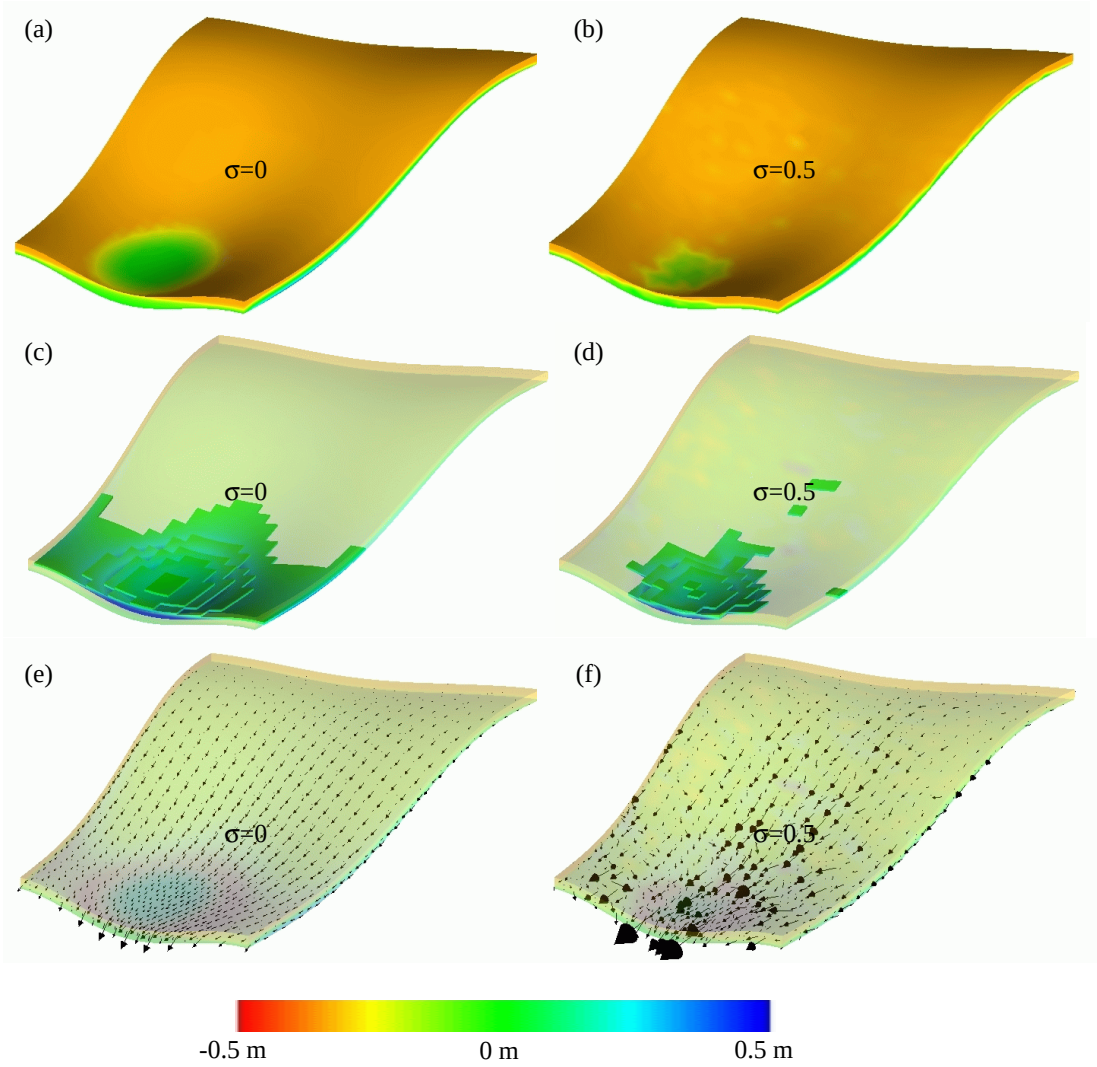

FIGURE 4.9: Pressure head and flux results of Test 3 at the end of simulation performed by the Newton method. (a,b) pressure head at surface; (c,d) pressure head for saturated zone; (e,f) flux vector. Surface pressure head was also plotted transparently in (c,d,e,f).

(Fig. 4.9-(b,d)). It is found that higher heterogeneity and anisotropy led to higher drainage performance when the same average value of K was used. Fig. 4.10 describes the seepage flux at the lower and surface boundaries, when the Newton methods was used. The plot of the seepage flux was almost the same in the case of the other two methods. It was found that when a high value of σ was used, the increase in seepage flux was faster than when a low value of σ was used.

The total number of iterations and the CPU time consumed in the three iteration methods are listed in Tables 4.7 and 4.8. In accordance with Tests 1 and 2, the Newton-Krylov method required the least number of iterations while it consumed more CPU

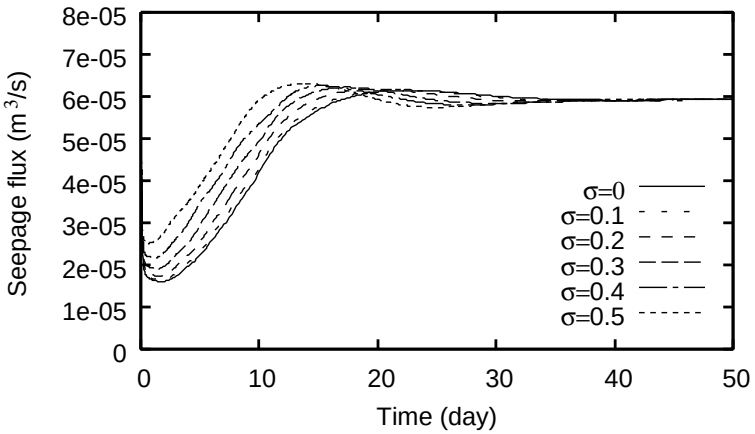

FIGURE 4.10: Seepage flux at the lower and surface boundaries using the Newton method.

TABLE 4.7: Test simulation 3, total iteration number.

Scheme	Total iteration number					
	$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
Picard	1520	1627	1714	1935	2936	3273
Newton	1463	1573	1865	2906	2670	6542
Newton-Krylov	264	288	279	290	298	329

TABLE 4.8: Test simulation 3, CPU time.

Scheme	CPU time (s)					
	$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
Picard	50	59	64	78	105	124
Newton	60	64	79	168	129	286
Newton-Krylov	174	201	210	257	277	280

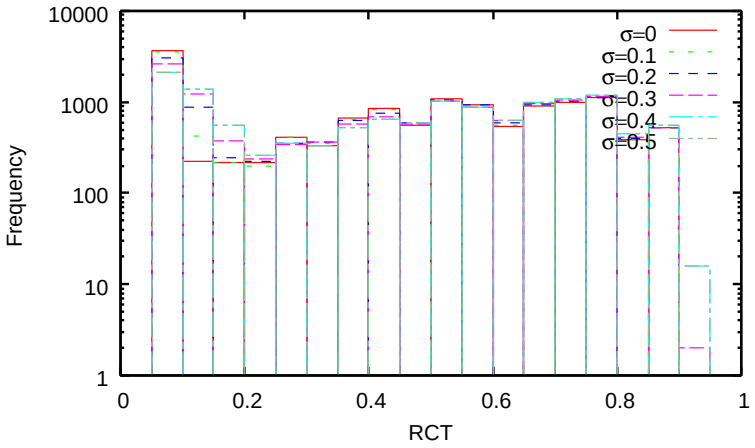

FIGURE 4.11: Test simulation 3, distribution of RCT.

time than the other two iteration methods. Except in the case of $\sigma = 0.5$, the Newton and Picard methods required similar numbers of iterations in this test simulation; when $\sigma = 0.5$, the Picard method required less iterations than the Newton method. The same situation occurred when MRCT was larger than 0.8 in Test 2. Even though the grid of Test 3 seemed to not be as highly skewed as Grid3 or Grid4 in Test 2, the values of MRCT were larger than 0.85 for all cases in Test 3 and the distribution of RCT was large, as shown in Fig. 4.11. It should be noted that the values of RCT are also associated with the ratio between the adjacent grid lengths, which affect the mesh skewness tensor $G^{p,q}$ and the degree of dominance of the cross-derivative terms. As shown in Fig. 4.8, the grid lengths in the vertical and horizontal directions are quite different. In such cases, although the grid skewness is not comparatively large, the value of RCT can be large. Through this test, it is confirmed that when the degree of dominance of the cross-derivative terms is high, the Newton and Picard methods showed similar behavior. As a result, the Picard method consumed less CPU times than the Newton method in this test simulation.

4.5 Summary

The performances of three iteration methods—the Picard, Newton and Newton-Krylov methods—for a three-dimensional non-orthogonal finite-difference model were compared in this study. The Picard and Newton methods employed a 7-point stencil approach in which the cross-derivative terms were evaluated in an explicit manner to avoid extremely complex programming and the cost of forming a 19-point stencil matrix. On the other hand, the Newton-Krylov method considered all the terms in an implicit manner while saving the cost of forming a 19-point stencil and avoiding the complexity of coding; however, the Newton-Krylov method requires the calculation of matrix-vector product instead of the direct calculation of the 19-point stencil matrix.

Three test cases were considered. The first test was a steady-state pumping well problem for comparing the convergence behaviors of the three iteration methods. With the line-search approach, the models using three methods solved the problems stably. The Newton-Krylov method converged faster than the other two methods. However the Newton-Krylov method required considerably more CPU resources per iteration than that required by the other two methods. Besides, the Picard method converged much

more slowly than the other two methods. As a result, the Newton method was found to consume the least CPU resources among the three iteration methods in Test 1. The second test simulation was a simple infiltration problem for a sand and loam soil domain with varying anisotropy and grid skewness. When the grid was highly skewed and the degree of anisotropy was high, the Newton-Krylov method consumed less CPU time than the other two iteration methods in the loam soil cases but it consumed more CPU time than the other two iteration methods in the sand soil cases. The Newton method consumed less CPU resources than the other two methods when the mesh skewness was not high in both soil cases. The third test was a rainfall-runoff simulation for a curvilinear slope with a constant rainfall intensity and varying anisotropy. It was found that the Newton and Picard methods required a similar number of iteration when the cross-derivative terms made large contribution to the system. The reason is considered to be that the terms representing the nonlinearity of K did not contribute significantly to the overall system as compared to the cross-derivative terms when the cross-derivative terms were dominant in the system. As a result, the Picard method consumed less CPU resources than the Newton method. For Test 3, the Newton-Krylov method required more CPU time than the other two iteration methods in all cases because of the additional cost.

From our test simulation, we could not find differences in the robustness and precision of the three iteration methods. Therefore, the CPU cost might be the most important feature to be considered when selecting the iteration method. If the grid is appropriately generated, the 7-point stencil Newton method seemed to generally be the most suitable choice among the iteration methods. In the case of where highly skewed grid and a high degree of anisotropy are required, the Picard method or the Newton-Krylov method could be the alternative iteration method. In addition, if programming complexity is disregarded, the Newton with implementation of the 19-point stencil could be the most efficient iteration method for three-dimensional coordinate-transformed FDM for simulating saturated–unsaturated flow in porous media. The 19-point stencil Newton method is supposed to converge faster and does not require the additional cost like the Newton-Krylov method.

Chapter 5

Development of a coupled model of pipe-matrix subsurface flow

5.1 Indroduction

Many researchers have revealed that the presence of soil pipe which is a chain of interconnected macrospores, develops nearly parallel to the ground surface in hillslopes. First, rainfall infiltrates into the soil matrix and the lateral flow to the pipe starts after the soil matrix becomes saturated. Then the pipe flow begins and when it becomes fulfilled condition, water can flow out from the pipe to the soil matrix. An experimental study using a fiberscope demonstrated that both full and partially filled conditions occurred simultaneously within the same soil pipe [86]. These soil pipes play an important role in hillslope hydrological processes as well as for the hillslope stability. The field observation conducted by *Kitahara* [42] shows that pipe flow can be expressed approximately using Darcy-Weisbach equation. Further he found that the pipe flow drains out water from the hllslope quickly and it increases the slope stability. However, when the closed pipe condition occurred, it may create high pressure zones at the lower part of the pipe, which might be a reason of slope failure.

As described above, the pipe flow has a significant effect on hydrological processes of the hillslope and the pipe flow mechanisms have complex flow dynamics and interaction with water in the surrounding soil. The last two decades have seen the development of a several models that considers pipe flow. Mainly there are two types of models, (1)

conceptual model and (2) physically-based model. As a conceptual model, there are dual-porosity, dual-permeability, multi-porosity, and multi-permeability models [25, 26, 35, 73]. These models assume that soil consists of severa iteracting regions addociated with the pipe and soil matrix inside soil aggregates. Details of conceptual models are well reviewed in *Šimůnek et al.* [82]. However, a few studies have proposed the physically-base models considering detailed interaction between soil matrix flow and pipe flow at the hillslope scale. A model proposed by *Kosugi et al.* [49] treated soil pipe as a highly permeable soil layer and both the matrix flow and the pipe flow were calculated by a saturated–unsaturated flow model. *Tsutsumi et al.* [89] developed a model describing three-dimensional steady-state flow problems in a hillslope with soil pipes. In their model, matrix flow and pipe flow are calculated using Richards and Mannings equation, respectively, while simultaneously considering the interaction between these two flow systems with iterative computations.

The objective of this chapter is to develope physically-based model which can simulate unsteady-state pipe-matrix flow, in which the saturated–unsaturated subsurface flow model and a slot model are combined in an iterative manner. The slot model calculates dynamic flow equation to simulate the pipe flows, which allows the model to represent a interaction between the matrix flow and the pipe flow more physically. For comparison, the two-dimentional (2D) and three-dimensional (3D) models are developed and the performace of 2D and 3D models are compared.

5.2 Model description

5.2.1 Basic concept of the model

As the hydraulic characteristics of pipe flow and soil matrix flow are considerably different, we treat each flow as a separate flow system. Pipe flow is calculated by the slot model and soil matrix flow is calculated by the saturated–unsaturated flow model. The slot model is capable of handling both open-channel and surcharged flows with an identical flow equation and being widely used in the calculation of the urban sewerage network [93]. Then the complex flow dynamics in the hillslope can be calculated by coupling the two models as Fig. 5.1: First, the saturated–unsaturated flow model calculates soil matrix flow using the water depth of pipe region as a pressure head boundary

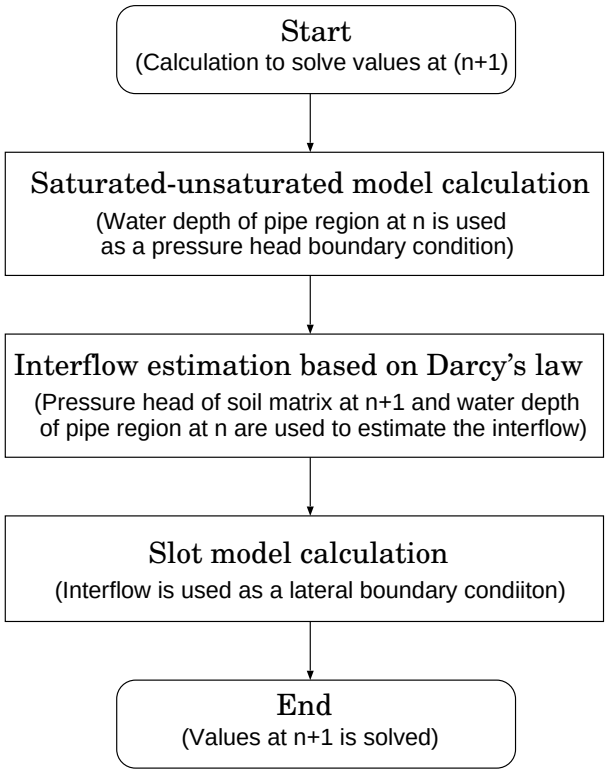

FIGURE 5.1: Flowchart of the coupling between the slot model and the saturated-unsatirated model

condition which is computed by the slot model. Then the interflow between the pipe and soil matrix is estimated by using Darcy's law. Finally, the slot model calculates pipe flow using the estimated interflow as a lateral boundary condition. The following two sections give brief descriptions of the saturated–unsaturated flow model and the pipe flow model.

5.2.2 saturated–unsaturated flow model

Because the shape of target domain was a rectangular parallelepiped, the three-dimensional conventional saturated–unsaturated flow equation was used as follows

$$\frac{\partial \theta}{\partial t} = \nabla \cdot K(\psi) \nabla \psi + \sin w \frac{\partial K(\psi)}{\partial x} + \cos w \frac{\partial K(\psi)}{\partial z}, \tag{5.1}$$

where w is the gradient of the slope. The backward Euler time discretization and the Picard iteration method were implemented as Eq. (2.5). The hydraulic conductivity of

the boundary between adjacent control volume units is defined as

$$K_{i\pm1/2,j,k} = \begin{cases} K_{i,j,k} & (\phi_{i,j,k} > \phi_{i\pm,j,k}) \\ K_{i\pm1,j,k} & (\phi_{i,j,k} \leq \phi_{i\pm,j,k}) \end{cases} \tag{5.2}$$

where ϕ is the hydraulic head. This upwind method gave more stable results than the arithmetic mean, Eq. (2.24), when grid is coarse [80].

5.2.3 Pipe flow model

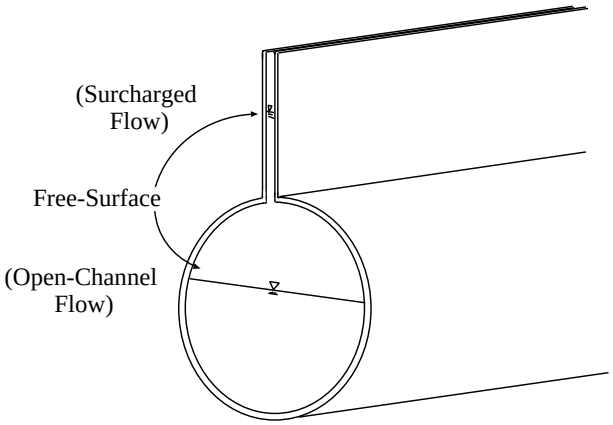

FIGURE 5.2: Schematic representation of slot model

The pipe flow is calculated by the slot model which assumes that a pipe has a slot of which width is vanishingly small and frictional resistance is negligible as Fig 5.2. The fluid motion within the slot model is described by the basic equation of the open-channel flow which is written as

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\cos\theta\frac{\partial h}{\partial x} - gAS_0 + gAS_f - \frac{1}{2}\frac{Q}{A}q = 0 \tag{5.3}$$

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} - q = 0 \tag{5.4}$$

where h is the depth of water measured from the bottom of the pipe, g is the gravitational acceleration, Q is the discharge, A is the area of flow, w is gradient of the pipe which is the same as the gradient of the slope in this study, S_f is the frictional gradient and q is the lateral flow discharge. S_f is written as

$$S_f = \frac{n^2 Q|Q|s^{\frac{4}{3}}}{A^{\frac{10}{3}}} = \frac{Q|Q|s}{C^2A^3}, \tag{5.5}$$

where n is Manning roughness coefficient, C is Chezy coefficient, and s is the length of wetted perimeter. In Eqs. (5.3) and (5.4), the flow between soil matrix and pipe is treated as lateral flow. Eqs. (5.3) and (5.4) are discretized by the Preissmann four-point finite difference scheme as

$$
\begin{aligned}
A &= \frac{1}{2}\left\{f\left(A_{i+1}^{n+1}-A_i^{n+1}\right)+(1-f)\left(A_{i+1}^{n}-A_i^{n}\right)\right\}, \qquad (5.6)\\
q &= \frac{1}{2}\left\{f\left(q_{i+1}^{n+1}-q_i^{n+1}\right)+(1-f)\left(q_{i+1}^{n}-q_i^{n}\right)\right\},\\
S_f &= \frac{1}{2}\left[f\left\{(S_f)_{i+1}^{n+1}-(S_f)_i^{n+1}\right\}+(1-f)\left\{(S_f)_{i+1}^{n}-(S_f)_i^{n}\right\}\right],\\
\frac{Q}{A} &= \frac{1}{2}\left[f\left\{\left(\frac{Q}{A}\right)_{i+1}^{n+1}-\left(\frac{Q}{A}\right)_i^{n+1}\right\}+(1-f)\left\{\left(\frac{Q}{A}\right)_{i+1}^{n}-\left(\frac{Q}{A}\right)_i^{n}\right\}\right],\\
\frac{\partial Q}{\partial t} &= \frac{1}{2\Delta t}\left\{\left(Q_{i+1}^{n+1}+Q_i^{n+1}\right)-\left(Q_{i+1}^{n}+Q_i^{n}\right)\right\}\\
\frac{\partial A}{\partial t} &= \frac{1}{2\Delta t}\left\{\left(A_{i+1}^{n+1}+A_i^{n+1}\right)-\left(A_{i+1}^{n}+A_i^{n}\right)\right\},\\
\frac{\partial Q}{\partial x} &= \frac{1}{\Delta x}\left\{f\left(Q_{i+1}^{n+1}-Q_i^{n+1}\right)+(1-f)\left(Q_{i+1}^{n}-Q_i^{n}\right)\right\},\\
\frac{\partial h}{\partial x} &= \frac{1}{\Delta x}\left\{f\left(h_{i+1}^{n+1}-h_i^{n+1}\right)+(1-f)\left(h_{i+1}^{n}-h_i^{n}\right)\right\},\\
\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) &= \frac{1}{\Delta x}\left[f\left\{\left(\frac{Q^2}{A}\right)_{i+1}^{n+1}-\left(\frac{Q^2}{A}\right)_i^{n+1}\right\}+(1-f)\left\{\left(\frac{Q^2}{A}\right)_{i+1}^{n}-\left(\frac{Q^2}{A}\right)_i^{n}\right\}\right],
\end{aligned}
$$

where f is the time weight parameter ($f = 1.0$, backward Euler; $f = 0.5$, Crank-Nicolson). In this solution, the terms $\partial A/\partial h$ and $\partial s/\partial h$ arise. When h reaches to zero, the terms become undefined ($\partial A/\partial h|_{h=0} = \infty$, $\partial s/\partial h|_{h=0} = \infty$). In order to avoid this difficulty, we define the cross sectional shape of the pipe as shown in Fig. 5.3. There are two shapes, one is almost round but has a small notch on the bottom, and the other is square. For both of them, if h reaches to zero, the values of $\partial A/\partial h$ and $\partial s/\partial h$ become constant.

5.3 Numerical simulation

5.3.1 Simulation Condition

In this study, the simulations were compared with laboratory experiments conducted by *Uchida et al.* [90]. For comparison, the two-dimentional and three-dimensional model The condition of experiment is as follows.

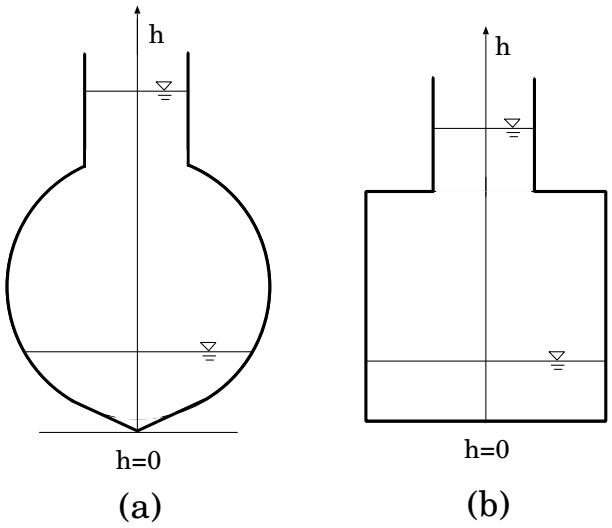

FIGURE 5.3: Cross-sectional shape of the pipe in the slot model

- A flume, 70cm long and 7cm wide, is inclined at 15 and filled with Toyoura standard sand to a thickness of 10 cm. To simulate soil pipe, an acrylic pipe of 1cm outer diameter, 0.8cm inner diameter, and 30cm long is used. Four tiny perforations of 0.2cm diameter are made in the walls of the acrylic pipe at 2cm intervals and the pipe is wrapped with a cotton cloth. The upstream end of the pipe is filled with silicon and closed in order to prevent the sand particles from entering the pipe. The water level at the lower end of the flume is fixed, and water is supplied to the upper tank at a constant rate of 0.5cms by using a rotary pump. *Kosugi* [47, 48]'s soil retention model fitted the the soil used in experiment and was used in simulations. *Kosugi* [47, 48]'s soil retention curve are expressed as follows:

$$\begin{cases} \theta = \theta_s & (\psi \geq 0) \\ \frac{\theta_s - \theta_r}{\theta - \theta_r} = F\left[\frac{\ln(\psi/\psi_m)}{\sigma_p}\right] & (\psi < 0) \end{cases} \tag{5.7}$$

$$\begin{cases} K(\psi) = K_s & (\psi \geq 0) \\ K(\psi) = K_s \left\{F\left[\frac{\ln(\psi/\psi_m)}{\sigma_p}\right]\right\}^{0.5} \left\{F\left[\frac{\ln(\psi/\psi_m)}{\sigma_p}\right] + \sigma\right\}^2 & (\psi < 0) \end{cases} \tag{5.8}$$

where $\Psi_m$is the pressure head corresponding to the median pore radius, σ_p is a dimensionless parameter to characterize the width of the pore-size distribution, and F denotes

TABLE 5.1: Soil properties reffering to *Kosugi et al.* [49].

θ_s	θ_r	Ψ_m	K_s	σ_p
0.368	0.044	-52.5 cm H_2O	2.23e-04 (m/s)	0.363

the complementary normal distribution function, defined as

$$F(x) = (2\pi)^{-0.5} \int_x^{\infty} \exp(-u^2/2) du \tag{5.9}$$

Kosugi et al. [49] simulated the same experiments as this study. The parameter set of soil properties used in this study are reffering to *Kosugi et al.* [49], which are listed in Table 5.1.

The simulation was conducted for three different cases. In the first case, the simulation was carried out without a pipe (no pipe condition). In the second case (open pipe condition), the pipe is buried so that its outlet is connected directly to the free water in the lower end of the flume as Fig. 5.4-(a). In the third case (closed pipe condition), the outlet of the pipe is placed at 15 cm upslope from the lower end of the slope as Fig. 5.4-(b). In case of the simulation of no pipe condition, the saturated–unsaturated flow model is used without being coupled with the slot model. In the simulations of the open and closed pipe conditions, the coupled model is used.

For the numerical stability of the slot model, a very small constant amount of discharge was added from the upstream end of the pipe. That discharge occupied only small fraction (<1%) of the outflow rate at the downstream end of the pipe throughout the simulations. Also, we tested the two types of cross-sectional shape of pipe shown in Figure 3 and found that the calculation with square shape, Fig. 5.3-(a), was stable and robust. The following results are obtained by using a square shape with 0.007×0.007m. The grid size of x, y, and z were 0.005 m and Δt was 10 secy. The parameter C was calibrated using the 3D model and the value was 0.096. For the initial condition of the simulations, we used the steady-state condition when the water level of the downstream boundary are fixed and no water is supplied from the upstream boundary, as in the experiment by *Uchida et al.* [90].

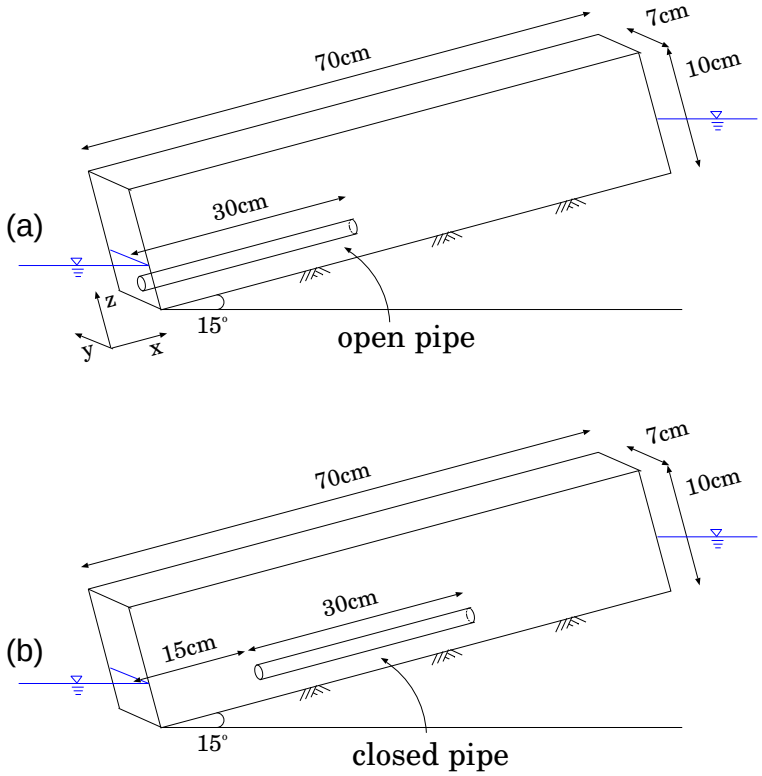

FIGURE 5.4: Schematic diagram of the experimental setup, (a) open pipe; (b) closed pipe

5.3.2 Results and discussions

Fig. 5.5 shows the water surface profiles obtained by the experiment (observation intervals was 10 cm), the 2D model, and the 3D model. In no pipe condition, the water surface profile of the simulation gradually increases in the upslope region with time and it matches well with the experiment. In open pipe condition, the water surface profile of the 3D model gradually increases with time and is in reasonable agreement with that of the experiment. However, in the result of the 2D model, the profile of the lower part does not match with that of the experiment. In closed pipe condition, the water surface profile of the 3D model is similar with that of the experiment. However, the overall water profile of the 2D model is much lower than that of the experiment and the 3D model.

These simulations show that the 2D model tends to predict the lower water table compared to the 3D model. Fig. 5.6 is the ratio of the pipe flow discharge to the total discharge (pipe flow + matrix flow) at the lower end of the flume for the open

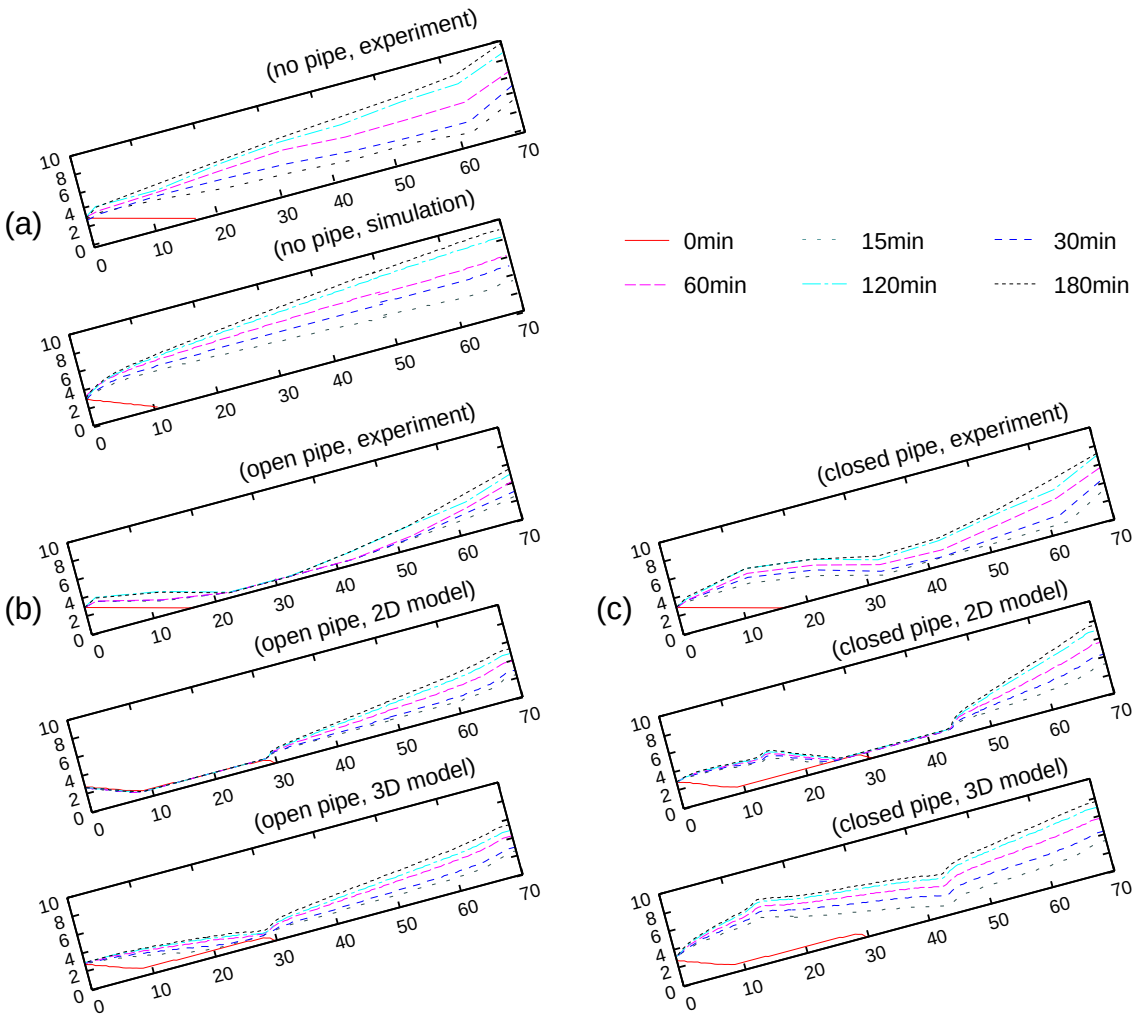

FIGURE 5.5: Comparison of the water surface profile between the simulation results and the expreriment observations. The dimensions are cm in figures. (a) no pipe; (b) open pipe; (c) closed pipe condition

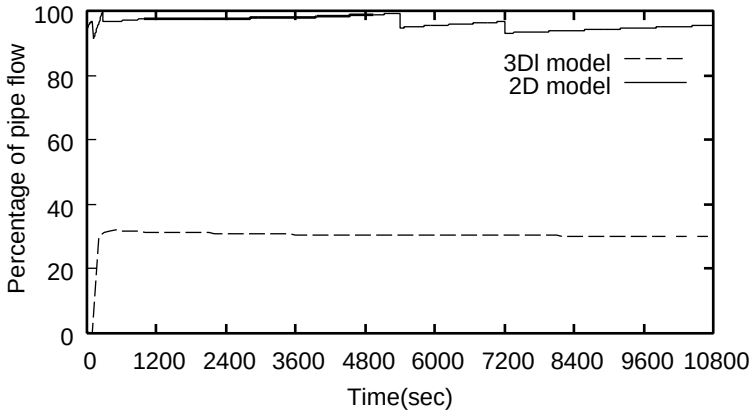

FIGURE 5.6: Percentage of the pipe flow

pipe simulation. Approximately 30% of the water was discharged through the pipe in the three dimensional simulation. On the other hand, almost all of the water was discharged through the pipe in the two dimensional simulation. The value of C used in these simulations was obtaind by calibration using 3D model, which might be the reason of disagreement between experimental observations and results obtained by 2D model. Therefore, we performed the 2D simulations again using different values of C, but we were not able to obtain good agreement of water table between the simulations and the experiment for both open and closed pipe conditions. These facts suggest that the effect of the pipe is overestimated in the two dimensional model.

Furthermore, even in this small experimental flume, the flow is three-dimensional. Fig. 5.7 shows the water flux vector and water surface profile at the - plane. At the upper end of open pipe, water flows into the pipe from the surrounding soil as Fig. 5.7-(a), while water flows out from the pipe to the soil at the lower end of closed pipe as Fig. 5.7-(b). Accordingly, the two dimensional model is not sufficient to deal with the interaction between pipe flow and soil matrix flow and the three dimensional model needs to be used to describe water dynamics in the heterogeneous soil layer.

5.4 Summary

In this chapter a physically-based pipe-matrix subsurface model was developed combining the three-dimensional saturated–unsaturated flow model with the slot model. The slot model calculates pipe flow by using the dynamic flow equation under both partially filled pipe and full pipe conditions. The saturated–unsaturated flow is calculated by using Richards equation. The simulations were cariied out for three different conditions (no pipe, open pipe, and closed pipe) by using the coupled model. The simulations were compared with experimental observations and the result of the 3D model showed reasonable agreement. The 2D model tends to overestimate the effect of the pipe, but the 3D model successfully describes the water dynamics of the pipe and soil matrix. The 3D model has a capability of dealing properly with the interaction between pipe flow and soil matrix flow.

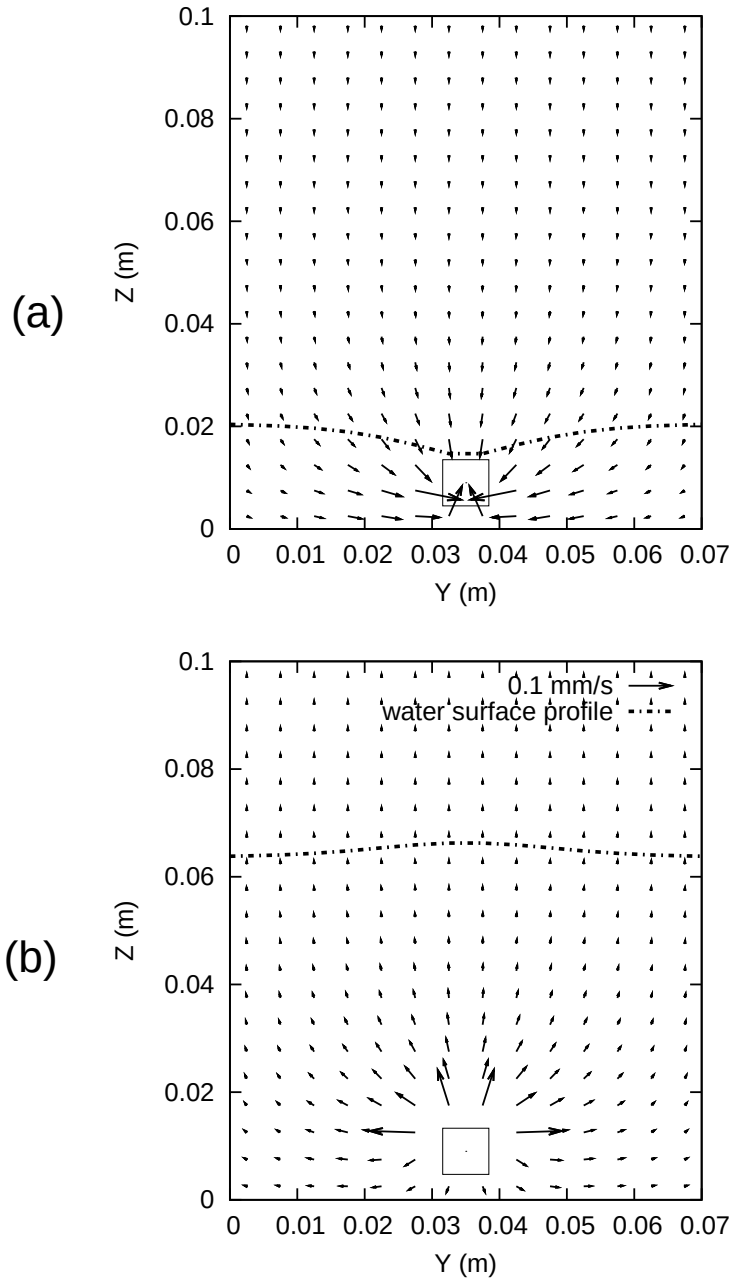

FIGURE 5.7: Water flux vector and water surface profile on $y - z$ plane at the upper end of open pipe; (b) at the lower end of closed pipe

Chapter 6

Assessment of integrated kinematic wave equations for a hillslope runoff modeling

6.1 Introduction

In hydrological modeling, the kinematic wave models have been broadly used for river flood simulations. *TaKasao & Shiiba* [84] proposed an integrated kinematic wave model which can handle two types of flows, matrix flow and overland flow. Their model can be applied to hillslope runoff simulation. *Tachikawa et al.* [83] extended this kinematic wave model and proposed a new integrated kinematic wave model which can deal with three different flows, unsaturated, saturated, and overland flows. These integrated kinematic wave models have been successfully applied to Japanese basins as an element model of distributed hydrological models in many researches. *Hunukumbra* [33] extended this model application and applied the models to several places in the world. He found a tendency that the model performances were good in wet and steep basins such as Japanese basins, while, in dry and mildly sloped basins, the integrated kinematic wave models could not give well agreed results with observation data.

Hunukumbra [33]'s finding is obviously important and critical in distributed hydrological modelings, but it is still unclear which hydrological condition affects the applicability of the integrated kinematic wave model, because only four different places were used in

the study of *Hunukumbra* [33]. To assess a correlation between the model validity and the each hydrological conditions, it is required to analyze the model performance in various places with many events. However, it is difficult to conduct this rearch using observation data because of limited available data and uncontrollable nature conditions. In this chapter, we carried out numerical experiment to assess the validity of the integrated kinematic wave model.

The results simulated by a physically-based two-dimensional (2D) model was considered as a surrogate of observations. The 2D model consists of a 2D subsurface flow model and a 1D overland flow model. Even though 2D Richards equation cannot perfectly simulate a runoff process of real hillslope in all aspect because of many uncertainty of soil properties, structures, unknowness of hydrological processes and etc., the same model concept has been used in several previous researches for simulating hillslope runoff and it successfully reproduces the runoff discharge and the other characteristics of hillslope. For example, *Hopp & McDonnell* [29] performed numerical experiments controlling storm size, slope angle, soil depth and bedrock permeability to study the effect for hillslope runoff using a model which is based on the same concept with the 2D model of this study. *Keim et al.* [38] also performed virtual experiments to investigate the process of evaporation and canopy interception using a model based on the same concept. Hence, the 2D model is considered to be useful to evaluate the effect of several hydrological conditions on runoff discharge.

Under several conditions with controlling slope angle, soil depth, rainfall intensity and initial condition, the simulations were conducted using the 2D model. The integrated kinematic wave model was calibrated to reproduce the results of the 2D model.

6.2 Model description

6.2.1 Integrated kinematic wave model

TaKasao & Shiiba [84] proposed an integrated kinematic wave model which can handle matrix flow and overland flow in one system based on relation between water depth and

discharge as follows:

$$q = \begin{cases} vh, & (h \leq d) \\ vh + \alpha(h - d_s)^m, & (d < h) \end{cases} \tag{6.1}$$

where h is water depth, q is flow discharge, v is velocity of subsurface flow, $\alpha(= \sqrt{i}/n)$ and β are parameters, and the shallow rectangular cross-section was assumed. *Tachikawa et al.* [83] modified this relation and proposed an extended model describing three types of flows. The extended model equation is as follows

$$q = \begin{cases} v_c(h/d_c)^\beta, & 0 \leq h \leq d_c \\ v_c d_c + v_a(h - d_c), & d_c < h \leq d_s \\ v_c d_c + v_a(h - d_c) + \alpha(h - d_s)^m, & d_s < h \end{cases} \tag{6.2}$$

where $v_c(= K_c i)$ is velocity of unsaturated flow and $v_a(= K_a i)$ is velocity of saturated flow. To keep continuity of flow velocity, $K_c = K_a/\beta$ is assumed, where K_c and K_a are hydraulic conductivities of the unsaturated and saturated flows, respectively. Fig. 6.2 shows the concept of this model. There are five parameters (n, k_a, d_s, d_c, β) in this model. If d_c is zero, v_a is v, and d_s is d, Eq. (6.2) is equivalent to Eq. (6.1).

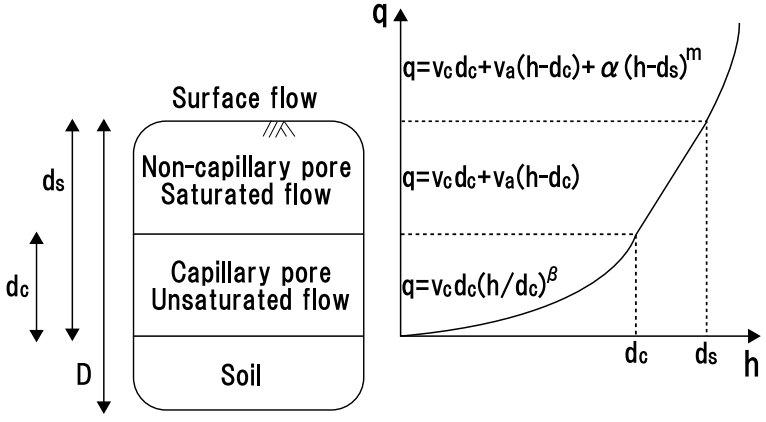

FIGURE 6.1: Concept of integrated kinematic wave model [83].

Combining Eq. (6.2) and the following continuity equation describes slope runoff system.

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = r(t) \tag{6.3}$$

where t is time, x is horizontal dimension, r is rainfall intensity.

6.2.2 2D model

The physically-based 2D model consists of a 2D subsurface flow model and a 1D overland flow model. Subsurface flow is described by 2D Richards equation as follows

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial x}\left\{K\frac{\partial \psi}{\partial x}\right\} + \frac{\partial}{\partial z}\left\{K\left(\frac{\partial \psi}{\partial z} + 1\right)\right\} \tag{6.4}$$

where ψ is the pressure head, θ is the volumetric moisture content, K is hydraulic conductivity, x denotes the horizontal dimension, and z denotes the vertical dimension, assumed to be positive upwards. A coordinate transformation technique was used to express non-orthogonal doamin as Fig. 6.2. The backward Euler time discretization and the Picard iteration method were implemented. Details fo discretization are shown in Chapter 3.

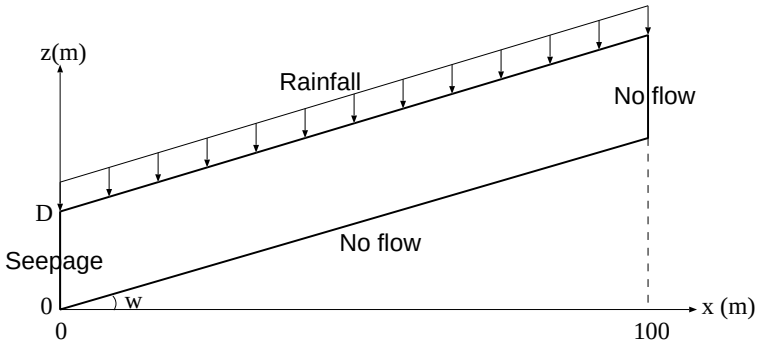

FIGURE 6.2: 2D model, the subsurface flow domain.

The upper (x = 100 m) and bottom ($z - x\sin w$ = 0 m) boundaries were set to be no flow. The lower boundary treated as a seepage face. Seepage face length was controlled automatically according to *Neuman* [64]. At the ground surface, water can enter the soil domain at the rainfall intensity (Neumann boundary) as long as ψ is negative, otherwise, it becomes Dirichlet boundary (ψ = 0). If the infiltration rate excesses the rainfall intensity while the surface node is Dirichlet boundary, it turns Neumann boundary condition again. This control is conducted in iterative procedure.

When the surface condition is saturated, the infiltration rate was calculated by the subsurface flow model. The amount of difference between the rainfall intensity and the calculated infiltration rate is the lateral inflow for the overland flow from soil domain and precipitation. Otherwise, when the surface condition is Neumann boundary, the infiltration rate is same as rainfall intensity and the lateral inflow for the overland flow

would be zero. The overland flow equation is written as follows

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = q_l(t) \tag{6.5}$$

$$q = \alpha h^m \tag{6.6}$$

where q_l is lateral flow rate into surface flow from subsurface flow and precipitation which is calculated by the subsurface flow model. The 2D subsurface flow model and the 1D overland flow model are combined explicitly.

6.3 Numerical experiment

6.3.1 Simulation condition

We considered the results simulated by the 2D model as a surrogate of observations. Under several conditions, the simulations were conducted using the 2D model and the integrated kinematic wave model was calibrated to reproduce the results of the 2D model as better as possible.

A simple slope was considered as Fig. 6.2. The slope length was 100m. The slope angle and soil depth were varied in simulation as Table. 6.1. Rainfall intensity was given as Fig. 6.3 when total rainfall is 10 mm. When total rainfall is 20, 40, 70, and 100 mm, the rainfall intensity was multiplied by 2, 4, 7 and 10, respectively. Two types of initial conditions were considered. A wet initial condition was simulated as soil moisture contents and the corresponding pressure head field after three days drainage from a saturated soil domain without rainfall. A dry initial condition was obtained after a week drainage.

TABLE 6.1: Simulation condition

slope angle	5, 20, 35 degree
soil depth	0.25, 0.5, 1.0, 2.0 m
total rainfall	10, 20, 40, 70, 100 mm
initial condition	wet, dry

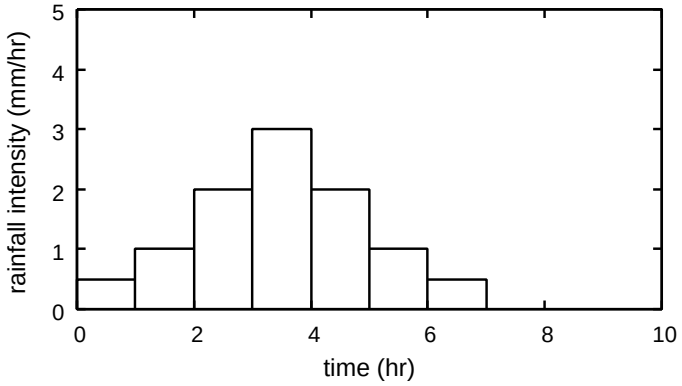

FIGURE 6.3: Rainfall intensity, total rainfall is 10 mm.

6.3.2 Model parameter

The equation of *Van Genuchten* [91] for the soil water retention curve and that of *Mualem* [61] for the unsaturated hydraulic conductivity function were used for the subsurface flow model. The parameter values were set as $\theta_s = 0.475$, $\theta_r = 0.28$, $K_s = 2.5$ m/s, $\alpha = 4$ m^{-1}, $n = 2$ by referring to *Hopp & McDonnell* [29].

As the model of *Tachikawa et al.* [83] includes that of *TaKasao & Shiiba* [84], the former one was used as the integrated kinematic wave model in this study. The five parameters (n, k_a, d_s, d_c, β) of the integrated kinematic wave model were calibrated for each simulations of the 2D model by using the Shuffled Complex Evolution (SCE) algorithm [20].

The Nash-Sutcliffe coefficient (NSE) [63] was used as the objective function of parameter optimization. The NSE is defined as

$$\mathrm{NSE} = 1 - \frac{\sum_{t=1}^{T} \left(q_0^t - q_m^t\right)^2}{\sum_{t=1}^{T} \left(q_0^t - \bar{q}_0\right)^2} \tag{6.7}$$

where q_0^t is observation discharge at time t, q_m^t is simulated discharge at time t, and $\bar{q}_0$ mean value of observed discharge. Nash-Sautcliffe coefficient can range from $-\infty$ to 1. The coefficient of 1 crresponds to a perfect match of simulated discharge to the observed data. NSE of 0 indicates that the simulated discharge is as accurate as the mean of the observed data. If the observed mean is a better predictior than the simulated discharge data, NSE is less than zero.

6.4 Result and discussion

6.4.1 Runoff discharge

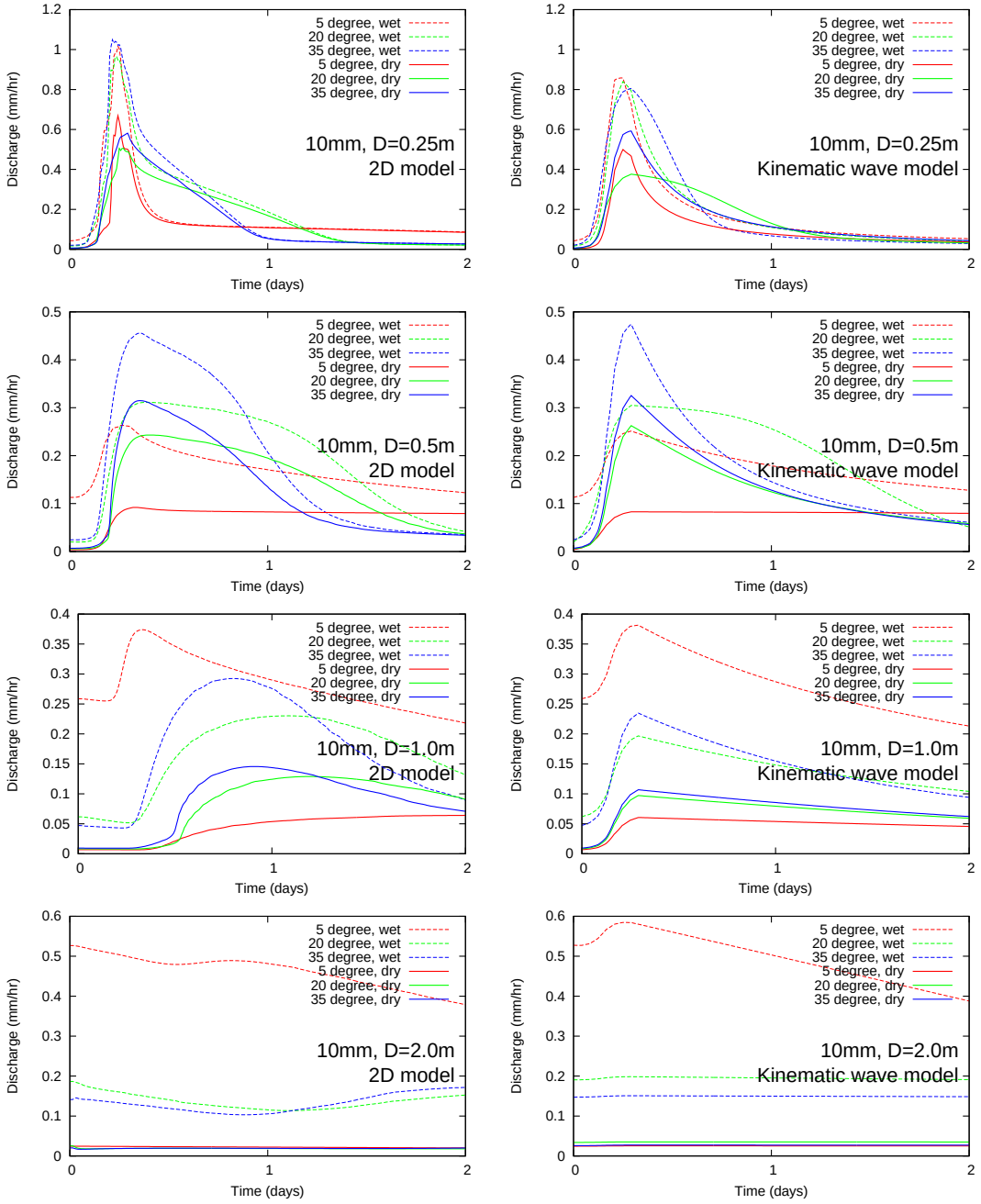

FIGURE 6.4: Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 10 mm.

Figs. 6.4∼6.8 describe runoff discharges simulated by the 2D model and the integrated kinematic wave model. The left column of the figure shows the results of the 2D model and the right column shows those of the integrated kinematic wave model. When

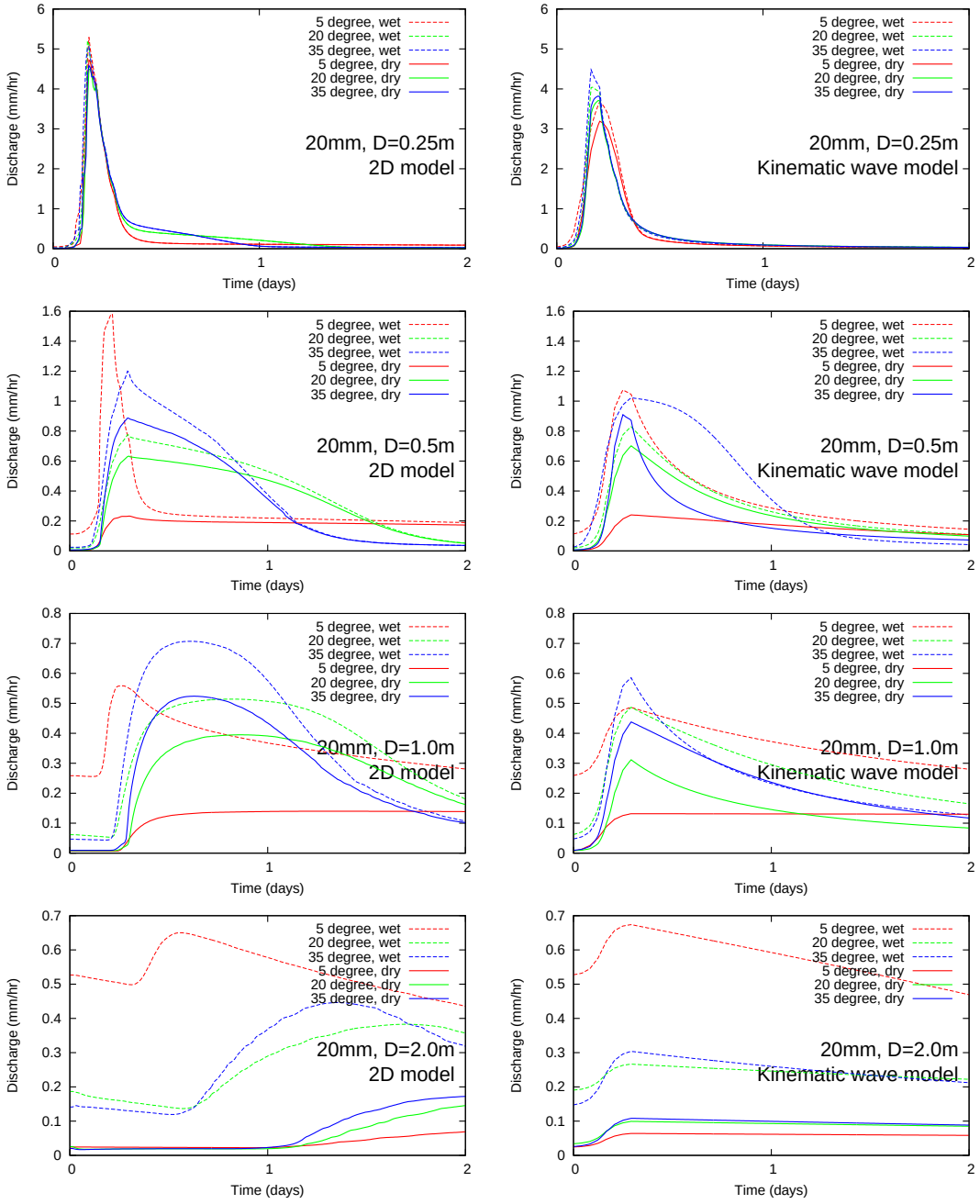

FIGURE 6.5: Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 20 mm.

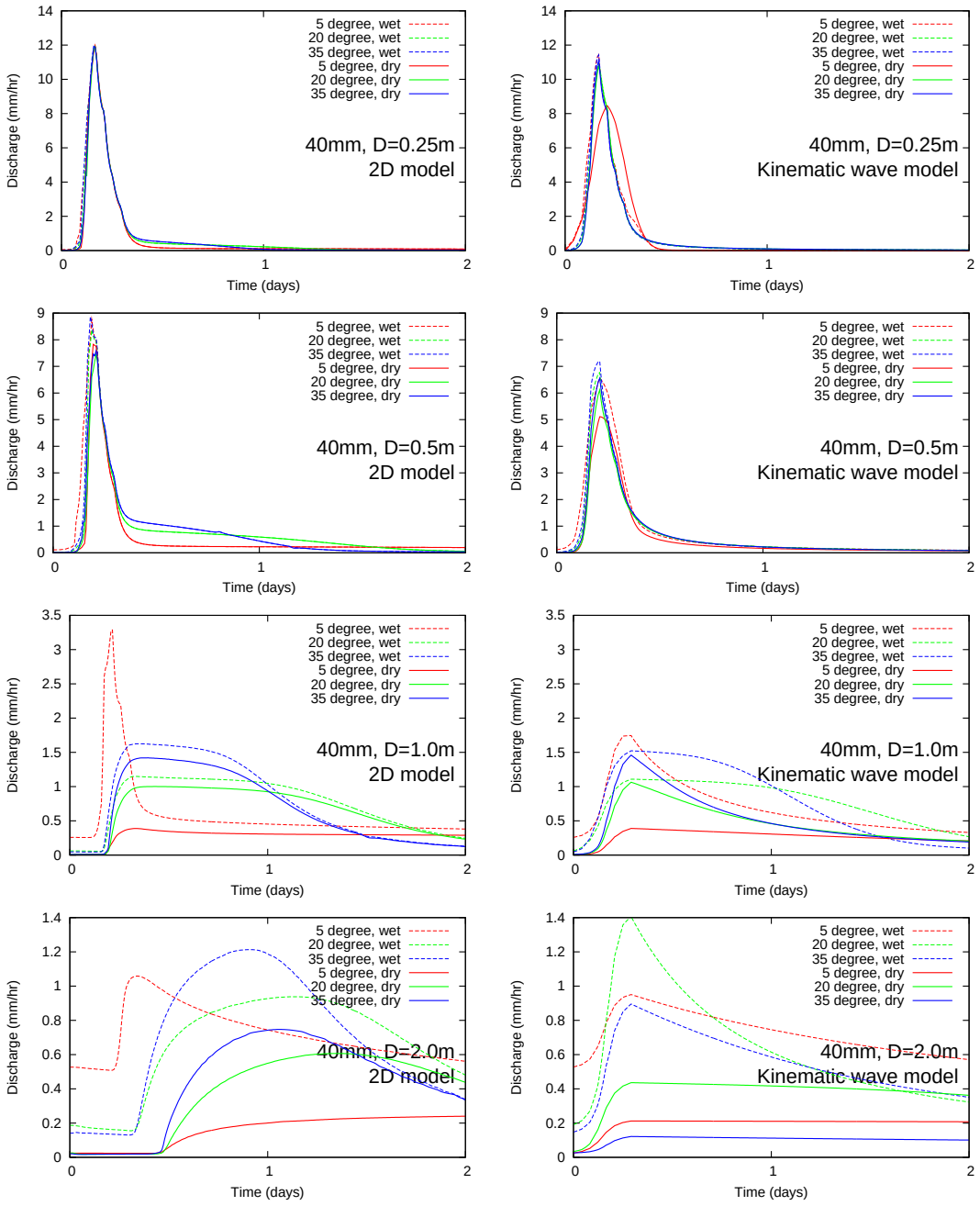

FIGURE 6.6: Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 40 mm.

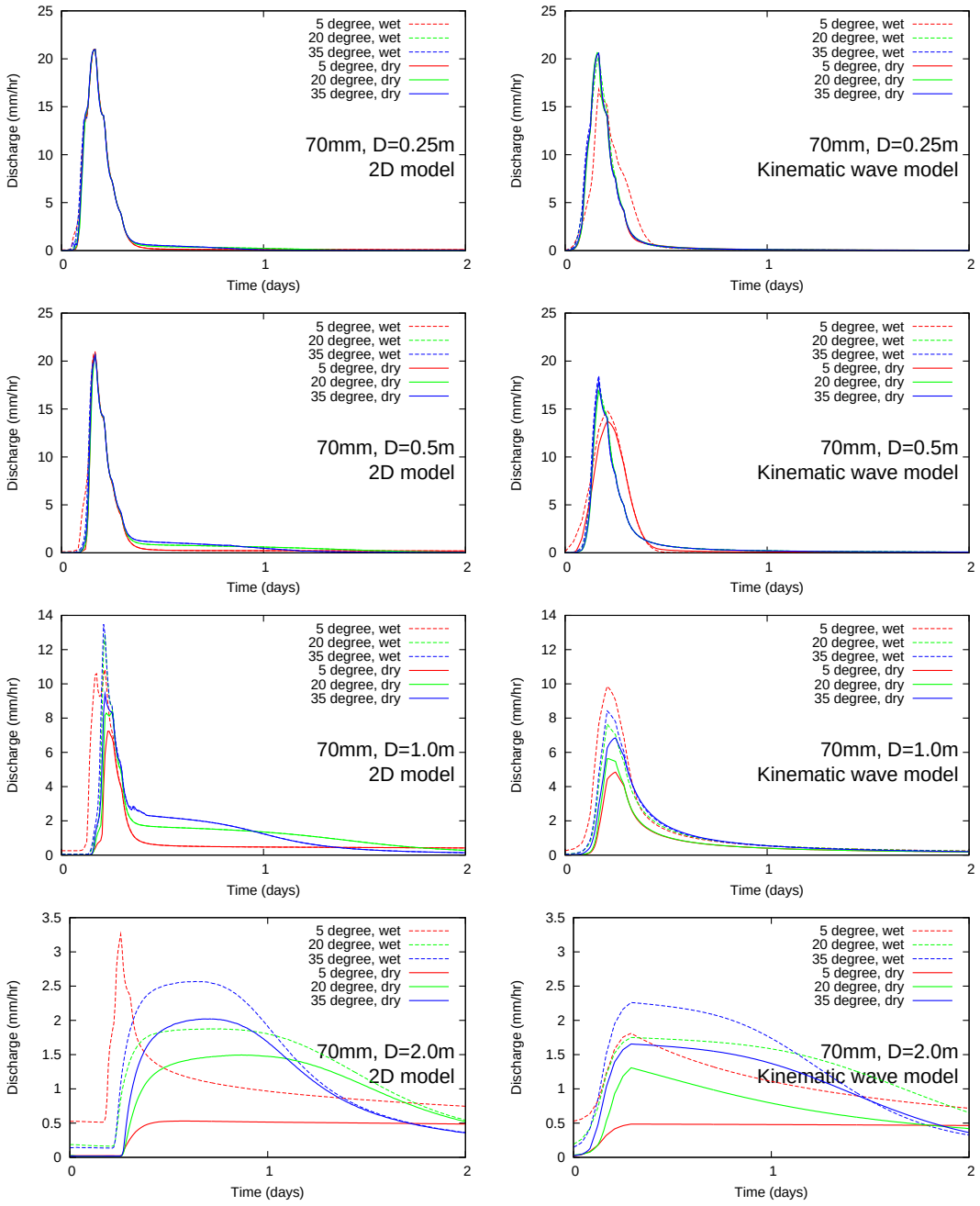

FIGURE 6.7: Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 70 mm.

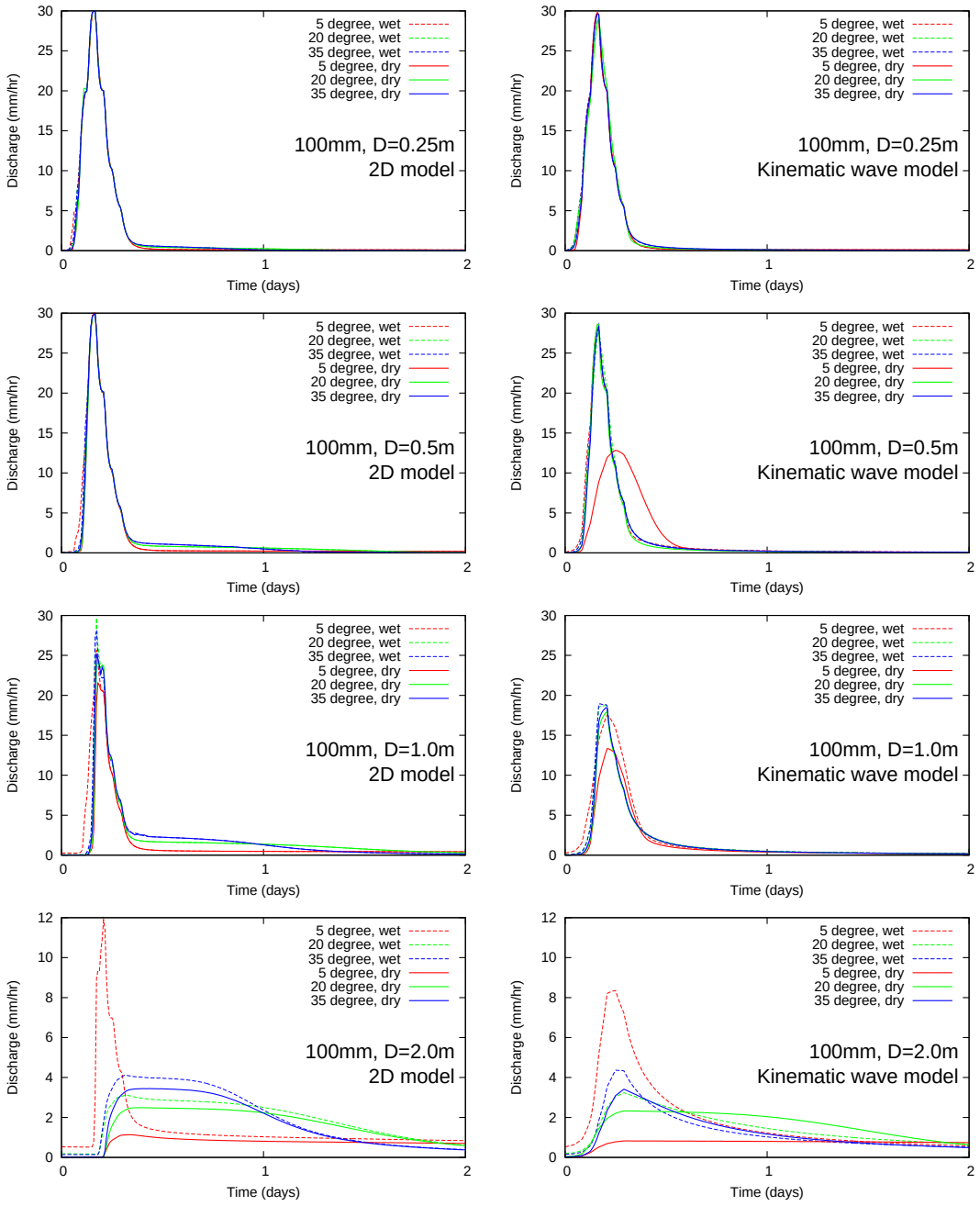

FIGURE 6.8: Hydrographs simulated by 2D model and integrated kinematic wave model when total rainfall was 100 mm.

soil depth is 0.25 m or 0.5 m, the runoff discharges simulated by the two models are well matched in mose cases. On the other hand, the discharge patterns are comparatively not agreed each other when soil depth is 1 m or 2 m. When soil depth is 2 m, the peak time of the integrated kinematic wave model appeared earlier than that of the 2D model when soil depth is 1 m ot 2 m. This time lag is supposed to be arised from vertical infiltration which cannot be treated in the integrated kinematic wave model. Furthermore, in the cases when total rainfall was large and slope was steep, the discharges of the integrated kinematic wave model were agreed well with those of 2D model. Entirely, the overall runoff pattern was well reproduced by the integrated kinematic wave model.

The slope of 5 degree performed by 2D model results in earlier peak flow than the others in some particular cases, e.g. D = 2 m of Fig. 6.4∼6.8. This is caused by the difference of initial condition which of 5 degree slope is wetter that the others. As previously mentioned, the initial conditions were made by simulating three days drainage from a saturated soil domain without rainfall. Hence, the initial conditions of 5 degree slope is generally wetter than the other two conditions.

6.4.2 Nash-Sutcliffe coefficient

Fig. 6.9 shows the NSEs with different soil depths. Obviously, NSEs become worse as soil depth becomes thicker. It could be arised from the effect of vertical infiltration. As soil depth is thicker, vertical infiltration gives larger effect to runoff process. Because the integrated kinematic wave model does not have a mechanism describing vertical infiltration in soil domain, it fails to reproduce the runoff discharge calculated by the 2D model which can express effect of vertical infiltration in thick soil condition. Furthermore, NSEs tend to be higher with wet initial conditions and vice versa with dry initial condition. This reason also might be two dimensional flow effect. As the initial condition is wetter, the soil domain is easily and quickly saturated and the vertical infiltration gives relatively small effect to runoff process. This corresponds with the discussion of Fig. 6.4∼6.8.

Fig. 6.6 shows the NSEs with different slope angels. It seems that there is no apparent correlation between slope angle and validity of the integrated kinematic wave model. However, except considerably wrong results (Nash-Sutcliffe coefficient less than 0.7), reproducibility tends to be better with larger slope angle. This might be arised

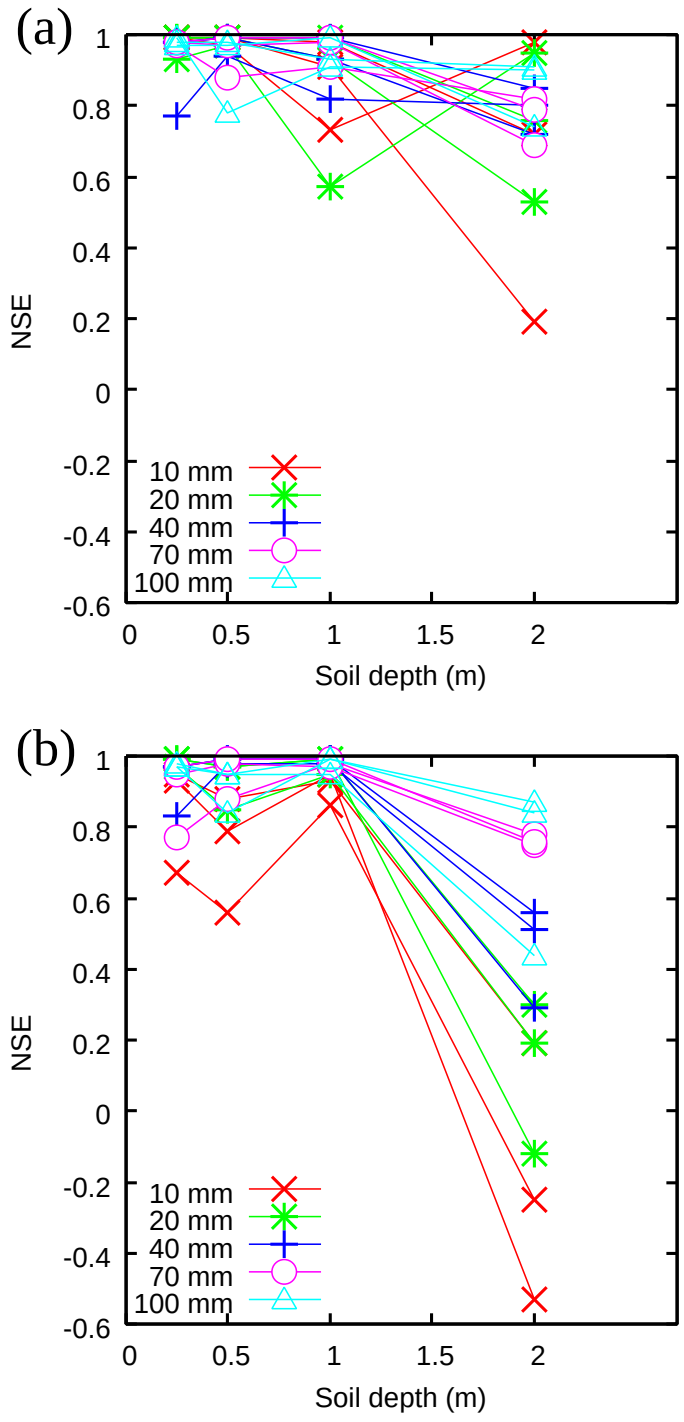

FIGURE 6.9: Nash-Sutcliffe coefficient with different soil depths: (a) wet, (b) dry initial condition.

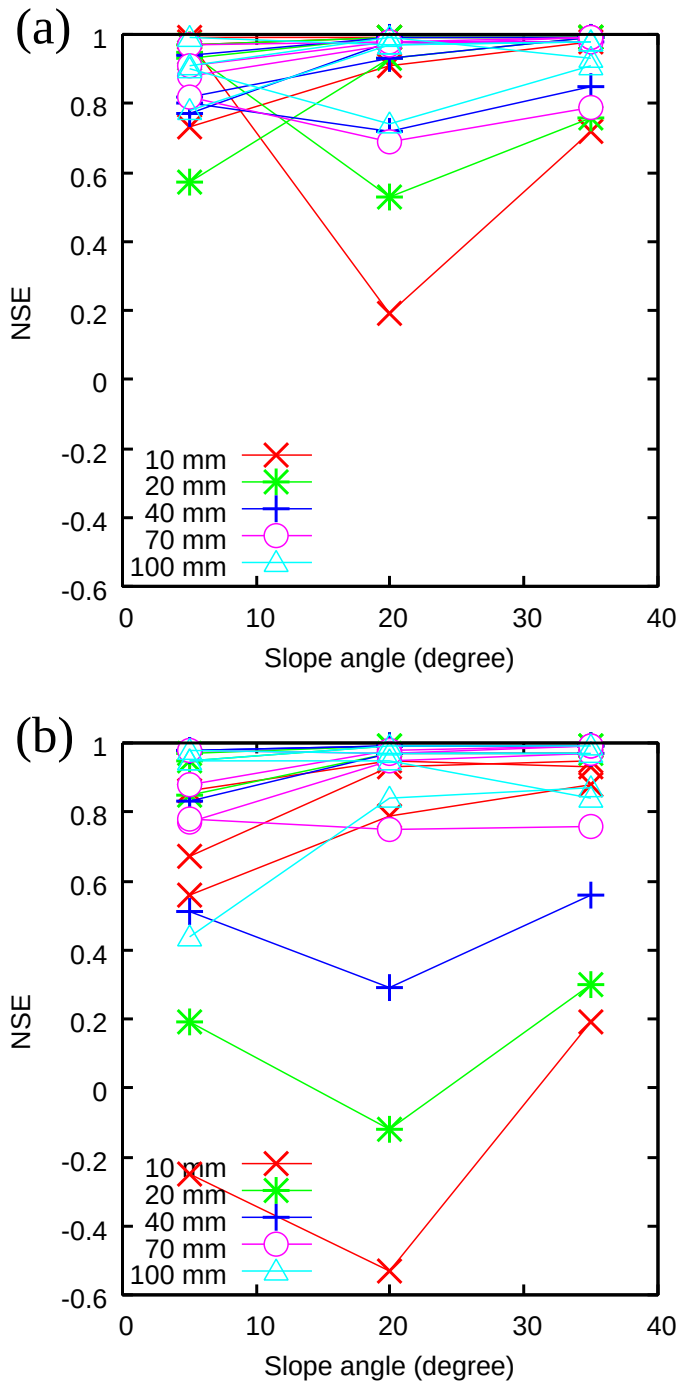

FIGURE 6.10: Nash-Sutcliffe coefficient with different slope angles: (a) wet, (b) dry initial condition.

from that flow condition of the lower part does not affect that of upper part in the kinematic wave model because it assumes supercritical flow.

These two characteristics shown in Fig. 6.9 and 6.10 about validity of the integrated kinematic wave model are well agreed with the research of *Hunukumbra* [33]. He found that the performance of the integrated kinematic wave model was good in wet and steep basins while it becomes worse in dry and mildly sloped basins.

Another point is that the validity is better with high rainfall intensity in Fig. 6.9 and 6.10. The reason is supposed to be same as that of the initial condition effect. In high rainfall intensity, vertical infiltration gives less effect to overall runoff process.

6.4.3 Discharge components

As previously mentioned, if $d_c = 0$ in the model proposed by *Tachikawa et al.* [83], the model results in that proposed by *TaKasao & Shiiba* [84]. Fig. 6.11 shows separate components of discharges of three models with 100 mm total rainfall, $D = 0.25$ m, 5 degree slope angle and wet initial condition. It can be found that the 2D model and the model proposed by *Tachikawa et al.* [83] give different discharge components even though the overall discharges are well agreed each other. The two integrated kinematic wave models gave different results and the hydrograph simulated by *TaKasao & Shiiba* [84]'s model is closer to the 2D model's result. However, as *TaKasao & Shiiba* [84]'s model can be considered as a variant of *Tachikawa et al.* [83]'s model, the result simulated by *TaKasao & Shiiba* [84]'s model can be considered as the result of *Tachikawa et al.* [83]'s model. This is a typical parameter equifinality problem. It means that another information is required to reproduce discharge components using the integrated kinematic wave model, e.g. specific parameter range or discharge rate of specific component.

6.5 Summary

To assess the validity of the integrated kinematic wave model, we carried out numerical experiments using the physically-based 2D model and the integrated kinematic wave model. The results simulated by the 2D model were assumed to be a surrogate of observations and the kinematic wave model was calibrated to reproduce the result of

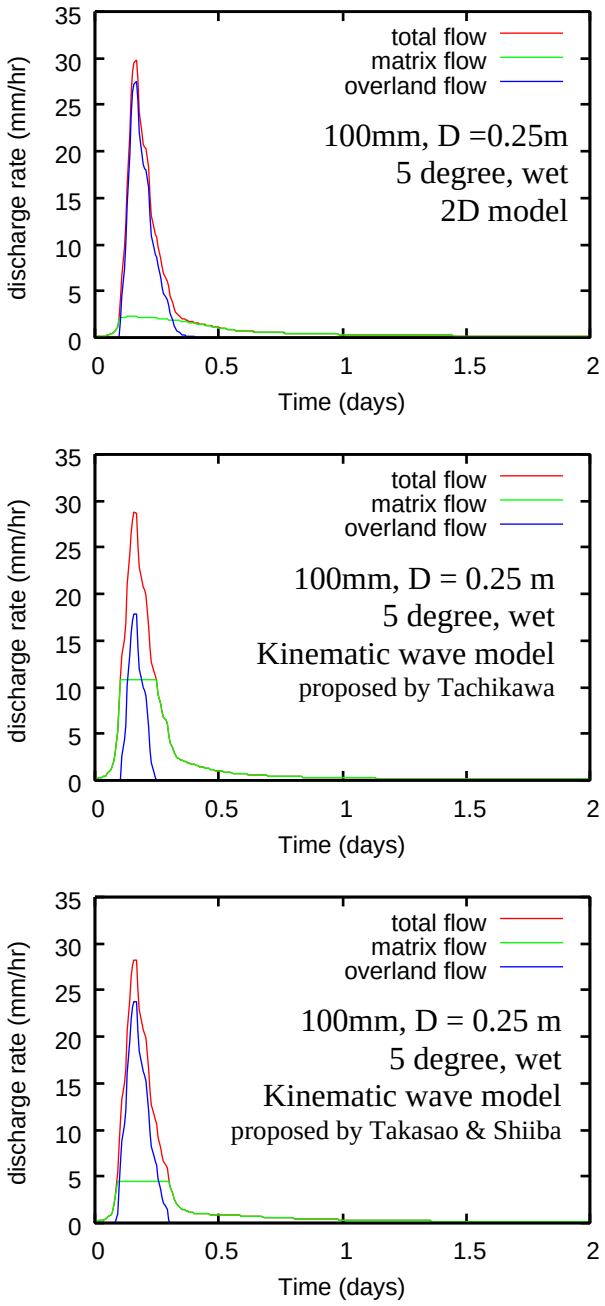

FIGURE 6.11: Matrix and overland flows simulated by the 2D model and the integrated kinematic wave model proposed by *Tachikawa et al.* [83] and *TaKasao & Shiiba* [84] (100 mm, $D = 0.25$ m, 5 degree, wet).

the 2D model. It was supposed that rainfall intensity, soil depth, slope angle, and initial condition could effect to the validity of the integrated kinematic wave model and the numerical experiments were conducted with changing these factors. According to the results of the numerical experiments, in condition of higher rainfall intensity, thinner soil depth, steeper slope angle, and wet initial condition, the integrated kinematic wave model could give good agreement with the 2D model. This analysis corresponds with the study of *Hunukumbra* [33]. Including vertical infiltration mechanism, the integrated kinematic wave model is expected to give better results for dry and mildly sloped basins.

Another important finding in this chapter is that if the integrated kinematic wave model perfectly reproduces the overall discharge from hillslope, it does not guarantee the good agreement with respect to discharge components. To obtain reliable result of separated discharge components using the integrated kinematic wave model, the other information which limits the range of model parameter should be needed in terms of hydrological characteristics of the study basins.

Chapter 7

Conclusions

This thesis has focused on mainly four objectives as follows:

1. To reduce computational costs of the saturated–unsaturated flow model.
2. To extend FDM for simulating the saturated–unsaturated flow by using the coordinate transformation method.
3. To compare the iteration methods of the saturated–unsaturated flow model.
4. To present the applications of the saturated–unsaturated flow model.

All of above objectives are involved with the numerical modeling of the saturated–unsaturated flow. The aim of the research on the saturated–unsaturated flow model is to use the model as a tool to understand the rainfall-runoff processes inside of a hillslope. Numerical modeling of saturated–unsaturated flow through porous media is an important research topic and involved in various branches of water resources engineering, agricultural engineering, chemical contaminant tracing, and rainfall-runoff modeling. Especially in hydrological engineering, the saturated–unsaturated subsurface flow model has been used for researching new findings obtained by observation studies. For example, to more deeply understand the hydrological processes of a hillslope including the preferential flow, the saturated–unsaturated subsurface flow model has been widely used in many recent researches. The saturated–unsaturated flow model is more physically-based than the conventional hydrological models such as the reservoir, tank,

and integrated kinematic wave models and can give us more detailed information about rainfall-runoff processes inside of a hillslope.

However, the saturated–unsaturated flow model consumes large amount of computer resources as compared to the conventional hydrological models. Rapid developments in computer technology have alleviated the computational resource problems but a three-dimensional subsurface flow simulation, particularly a simulation in a wide region with a relatively fine grid resolution, still requires a large amount of computer resources. This thesis has tried to improve this problem using the IADI method.

Many numerical models including FDM, FEM, FVM, mixed FEM and Eulerian-Lagrangian have been developed for simulating saturated–unsaturated flow in the past three decades. Among of those models, this thesis researched on the numerical methods of FDM. FDM is one of the most widely used models for simulating saturated–unsaturated flow and has advantages in terms of ease of coding and understanding owing to its simplicity of discretization as compared to the other models. However, it has been often pointed out that FDM do not accurately represent all geometrically complex flow domains. This thesis has tried to improve this problem using the coordinate transformation method. Then, to clear which iteration method is the most efficient under particular conditions, three different iteration methods were compared throughout test simulations. Further, two application cases of using the saturated–unsaturated flow model as a tool to study a subsurface flow of a mountain slope.

In **Chapter 2**, a new IADI algorithm for simulating saturated–unsaturated flow was developed. The IADI scheme has advantages in terms of simplicity and computational cost (on a per iteration basis) because only tridiagonal linear systems are involved in the calculation procedure. However, the saturated–unsaturated subsurface flow model using the conventional IADI method resulted in numerical instabilities and convergence difficulties. Moreover, the conventional IADI scheme could not be applied to 3D problems. To overcome this problems, AIADI scheme was derived based on the Douglas-Rachford ADI algorithm. AIADI scheme is a perturbation form of the backward Euler difference equation linearized by the Picard method. AIADI scheme successfully performed 2D and 3D test problems and showed better stabilities than the conventional IADI scheme. However, the AIADI scheme generated comparatively large mass balance error for simulating saturated flow compared with the Picard iteration scheme. This mass balance

error is arised from the additional terms, which are added to make the AIADI scheme stable for simulating saturated flow. But AIADI scheme was faster than the Euler backward implicit scheme linearized by the Picard iteration method in our test cases. Therefore, users can choice between the implicit scheme conserving the accuracy of mass balance and the AIADI scheme consuming less CPU resources and having a relatively ease of implementation.

To alleviate the restriction of FDM on curvilinear flow domain, the coordinate transformation method was applied to FDM in **Chapter 3**. Because of highly nonlinearity, implicit temporal discretization and iterative procedures are generally needed for numerical stability in saturated–unsaturated flow modeling. Coordinate transformation represents diffusion with cross-derivative terms. The transformed equation requires a 19-point stencil instead of a 7-point stencil, which is required by a conventional FDM. Forming 19-point stencil matrix make the numerical method extremely complex and less attractive. To avoid those difficulties, 7-point stencil model was developed in this thesis. In this approach, the cross-derivative terms were evaluated at the previous iterative level, and the other terms were evaluated at the current iterative level. Therefore, the 7-point stencil is calculated implicitly by the iterative calculation, as in the case of the conventional finite-difference model. The performances of 19-point stencil model and 7-point stencil model were investigated through 2D infiltration problems. The results of two models were well agreed each other. Except the cases when highly skewed grid was used, 7-point stencil model showed better performance than 19-point stencil model whereas 19-point stencil model was more efficient than 7-point stencil model when highly skewed grid was used. To verify the results and compare the performance of the proposed model, the test simulations were performed by the proposed model and HYDRUS. The proposed model successfully performed 2D and 3D test simulations with curvilinear flow domain. Moreover, the proposed model exhibited smaller mass balance error, the same order of relative error, and a faster convergence speed than HYDRUS in our test simulations. The conventional FDM with orthogonal and high-resolution grid was also compared with the proposed model. Even using low-resolution grid, the proposed model successfully simulated the rainfall-runoff system of a curvilinear slope. The results of the proposed model with lower-resolution and t he conventional FDM with orthogonal and high-resolution were well agreed. As a results of different resolutions required by two models, the proposed model consumed much less computational resources than

the conventional FDM. This implies that the proposed model may save considerable computational time as compared to the conventional FDM, e.g., in the case of iterative parameter estimation or Monte Carlo exercise, which usually requires hundreds or thousands of runs to arrive at an ideal parameter set or an objective function.

The Picard method was used as the iteration method for nonlinear system in chapter 3. In **Chapter 4**, the performances of different iteration methods were compared. The Picard and Newton iteration methods are the most common approaches for Richards' equation. The advantages and disadvantages of the Picard and Newton methods are well known. The Picard method is simple to implement and cost-efficient (per iteration basis). However it converges slowly than the Newton method. On the other hand, the Newton method is comparatively complex to implement and consumes more CPU resources per iteration than the Picard method. But it converges fastly. The comparison between two methods in FEM for saturated–unsaturated flow was already well evaluated. However, it is expected that two iteration methods shows different behavior in FDM using the coordinate transformation method because the iteration is progressed in partially implicit manner in the proposed model presented in Chapter 3. To avoid complexity of forming 19-point stencil matrix, the cross-derivative terms were evaluated explicitly in iteration procedure. But there is another way to avoid complexity of programming besides 7-point stencil strategy. The Newton-Krylov method requires calculating the matrix-vector product instead of forming 19-stencil matrix directly. The matrix-vector product can be approximated by taking differences of the original nonlinear function in the procedure of Krylov iteration. But additional cost is expected for calculating the original nonlinear function at every Krylov iteration step. Hence, there might be a trade-off between the Newton-Krylov method and 7-point stencil Newton iteration method using partially implicit manner. The performances of three iteration methods, the 7-point stencil Newton iteration, the 7-point stencil Picard iteration, and the Newton-Krylov method were investigated through three test simulations. The Newton-Krylov method converged more fastly than the other two methods but consumed more than 5 times CPU resources per iteration than the other two methods. When mesh skewness and anisotropy of flow domain are extremely high, the Newton-Krylov method consumed less CPU resources than the other two iteration methods. Generally, when mesh skewness and anisotropy of flow domain are not high, the 7-point stencil methods required the least CPU resources among the three methods in our test simulations.

There were some cases when the Picard method consumed less CPU resources than the Newton method. But in those cases, the differences of CPU costs consumed by two methods were not significant except the cases when both mesh skewness and anisotropy of flow domain are high.

In **Chapters 5**, the physically-based pipe-matrix coupled model was developed. The saturated–unsaturated subsurface flow model was combined with the slot model in an iterative manner. The pipe flow has a significant effect on hydrological processes of the hillslope and the pipe flow mechanisms have complex flow dynamics and interaction with water in the surrounding soil. The last two decades have seen the development of a several models that considers pipe flow. However, a few studies have proposed the physically-base models considering detailed interaction between soil matrix flow and pipe flow at the hillslope scale. Therefore, the physically-based model which can simulate unsteady-state pipe-matrix flow was developed. To verify the developed model, the simulations of no pipe, open pipe, and closed pipe cases were performed by the 2D and 3D models. The simulation results were compared with the experimental observation data. The water surface profiles obtained by the 3D model were agreed well with the observation data whereas 2D model cannot reproduce the observation data with one parameter set. It was found that 2D model tended to overestimate the effect of pipe flow compared with 3D model. Therefore, to represent more exact hydrological processes of hillslopes including the interaction between the pipe flow and matrix, 3D model is desirable.

In **Chapters 6**, the integrated kinematic wave model for a hillslope modeling was assessed. Usually the kinematic wave models have been broadly used for river flood simulations. *TaKasao & Shiiba* [84] and *Tachikawa et al.* [83] proposed the integrated kinematic wave model which can handle both subsurface flow and overland flow in one system. Their models can be applied to hillslope runoff simulation and have been successfully applied to several Japanese basins, which were comparatively wet and steep basin. However, the results simulated by the integrated kinematic wave model were not agreed well with the observation data in dry and mildly sloped basins. This finding was obviously important, but it was still unclear which hydrological condition affected the applicability of the integrated kinematic wave model, because a few cases have been reported. However, it is difficult to conduct the simulation in various places and compare the simulations results with the observation data because of limited available data and

uncontrollable nature conditions. Therefore, numerical experiments were carried out to assess the validity of the integrated kinematic wave model. The results simulated by a physically-based 2D model were considered as a surrogate of observations. The 2D model consists of a 2D saturated–unsaturated subsurface flow model and a 1D overland flow model. Under several conditions with controlling slope angle, soil depth, rainfall intensity and initial condition, the simulations were conducted using the 2D model. The integrated kinematic wave model was calibrated to reproduce the results of the 2D model. According to the results of the numerical experiments, in condition of higher rainfall intensity, thinner soil depth, steeper slope angle, and wet initial condition, the integrated kinematic wave model could give good agreement with the 2D model. Especially when soil depth is thick, the integrated kinematic wave model poorly reproduced the results of 2D model because the integrated kinematic wave model cannot treat the vertical infiltration. Therefore, including vertical infiltration mechanism inside of the model, the integrated kinematic wave model is expected to give better results for dry and mildly sloped basins.

The numerical methods for simulating only one-phase flow in porous media have been presented in this thesis. However, the focusses of the recent researches are not only on water flow but also on solute or heat transport in porous media, e.g., chemical contamination tracing, analysis of groundwater residence time, interaction between freshwater and saltwater, and etc. The presented model in this thesis can be extended for solute and heat transport in porous media by applying solute and heat transport governing equations to the presented model.

Bibliography

[1] An, H., Ichikawa, Y., Tachikawa, Y. and Shiiba, M. (2008), Developing a three-dimensional coupled model of pipe-matrix subsurface flow, *Hydrological Research Letter 2*, 52-55.

[2] An, H., Ichikawa, Y., Tachikawa, Y. and Shiiba, M. (2010), Three-dimensional finite-difference saturated–unsaturated flow modeling with non-orthogonal grids, using a coordinate transformation method, *Annual Journal of Hydraulic Engineering, JSCE, 54*, 505-510.

[3] An, H., Ichikawa, Y., Tachikawa, Y. and Shiiba, M. (2010), Validity assessment of integrated kinematic wave equations for hillslope rainfall-runoff modeling, *Water Resources Research, 46*, W11521, doi:10.1029/2009WR009024.

[4] Bause, M., Knabner, P., and Bergamaschi, L. (2004), Computation of variably saturated subsurface flow by adaptive mixed hybrid finite element methods, *Advances in Water Resources, 27*, 145-144.

[5] Bergamaschi, L. and Putti, M. (1999), Mixed finite elements and Newton-type linearizations for the solution for the unsaturated flow equation, *International Journal for Numerical Methods in Engineering, 45*, 1025-1046.

[6] Billstein, M., Svensson, U. and Johansson, N. (1999), Development and validation of a numerical model of flow through embarkment dams - comparisons with experimental data and analytical solutions, *Transport in Porous Media, 35*, 395-406.

[7] Broadbridge, P. and White, I. (1988), Constant rate rainfall infiltration: A versatile nonlinear model, 1, *Water Resources Research, 24*, 145-154.

[8] Brown, P.N. and Saad, Y. (1990), Hybrid Krylov methods for nonlinear Newton-Krylov algorithms, *SIAM Journal on scientific and statistical computing, 11*, 450-481.

[9] Carsel, R.F. and Parrish, R.S. (1988), Developing joint probability distributions of soil water retention characteristics, *Water Resources Research, 24*, 755-769.

[10] Celia, M.A., Bouloutas, E.T., and Zarba, R.L. (1990), A general mass-conservative numerical solution for the unsaturated flow equation, *Water Resources Research, 26(7)*, 1483-1496.

[11] Chan, T.F. and Jackson, K.R. (1984), Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms, *SIAM Journal on scientific and statistical computing, 5*, 533-542.

[12] Chapman, T.G. and Ong, G. (2006), A new equation for shallow groundwater flow over a curved impermeable boundary: Numerical solutions and laboratory tests, *Water Resources Research, 42*, W03427, doi:10.1029/2005WR004437.

[13] Chen, X. (2004), A Cartesian method for fitting the bathymetry and tracking the dynamic position of the shoreline in a three-dimensional, hydrodynamic model, *Journal of computational physics, 200*, 749-768.

[14] Chung, T.J. (2002), Computational fluid dynamics. Cambridge University Press.

[15] Clement, T.P., Wise, W. R., and Molz, F.J. (1994), A physically based, two-dimensional, finite-difference algorithm for modeling variably saturated flow, *Journal of Hydrology, 161*, 69-91.

[16] Cooley, R.L. (1971), A finite difference method for unsteady flow in variably saturated porous media: application to a single pumping well, *Water Resources Research, 7*, 1607-1625.

[17] Dane, J.H. and Mathis, F.H. (1981), An adaptive finite difference scheme for the one-dimensional water flow equation, *Soil Science Society of America, 45*, 1048-1054.

[18] Dogan, A. and Motz, L.H. (2005), Saturated-unsaturated 3D groundwater model. I: development, *Journal of Hydrologic Engineering, 10(6)*, 492-504.

[19] Douglas, J., and Rachford, H.H. (1956), On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society, 82(2)*, 421–439.

[20] Duan, Q., Sorooshian, S. and Gupta, V.K. (1992), Effective and efficient global optimization for conceptual rainfall-runoff models, *Water Resources Research, 28(4)*, 1015–1031.

[21] Farthing, M.W., Kees, C.E., and Miller, C.A. (2003), Mixed finite element methods and higher order temporal approximations for variably saturated groundwater flow, *Advances in Water Resources, 25*, 85-101.

[22] Fassino, C. and Manzini, G. (1999), Fast-secant algorithms for the non-linear Richards equation, *Communications in Numerical Methods in Engineering, 14*, 921-930.

[23] Freeze, R.A. (1971), Three dimensional transient, saturated-unsaturated flow in a groundwater basin, *Water Resources Research, 7*, 153-171.

[24] Forsyth, P.A., Wu, Y.S., and Pruess, K. (1995), Robust numerical methods for saturated-unsaturated flow with dry initial conditions in heterogeneous media, *Advances in Water Resources, 18*, 25-38.

[25] Gerke, H.H. and Van Genuchten, M. (1993), A dual-porosity model for simulating the preferential movement of water and solutes in structured porous media, *Water Resources Research, 29*, 305-319.

[26] Gwo, J.P., Jardine, P.M., Wilson, G.V., and Teh, G.T. (1995), A multiple-pore-region concept to modeling mass transfer in subsurface media, *Journal of Hydrology, 164*, 217-237.

[27] Haverkamp, R. and Vauclin, M. (1981), A comparative study of three forms of the Richard equation used for predicting one-dimensional infiltration in unsaturated soil, *Soil Science Society of America, 45*, 13-20.

[28] Hogarth, W.L., Parlange, J.Y. (2000), Application and improvement of a recent approximate analytical solution of Richards' equation, *Water Resources Research, 36*, 1965-1968.

[29] Hopp, L. and McDonnell, J.J. (2009), Connectivity of hill slope scale: identifying between interactions storm size, bedrock permiability, slope angle and soil depth, *Journal of Hydrology, 376*, 378–391.

[30] Hsu, M., Teng, W., and Lai, C. (1998) Numerical simulation of supercritical shock wave in channel contraction, *Computers & Fluids, 27*, 347-365.

[31] Huang, K., Zhang, R., and Van Genuchten, M. (1994), Eulerian-Lagrangian approach with an adaptively corrected method of characteristics to simulate variably saturated flow, *Water Resources Research, 30*, 499-507.

[32] Huang, K., Mohantry, B.P., and Van Genuchten, M. (1996), A new convergence criterion for the modified Picard iteration method to solve the variably saturated flow equation, *Journal of Hydrology, 178*, 69-91.

[33] Hunukumbura, J.M.P.B.: Distributed hydrological model transferability across basins with different physio-climatic characteristics, Phd thesis, Kyoto University, 2009.

[34] Jie, Y., Jie, G., Mao, Z. and Li, G. (2004), Seepage analysis based on boundary-fitted coordinate transformation method, *Computers and Geotechnics, 31*, 279-283.

[35] Jarvis, N.J. (1998), Modeling the impact of preferential flow on nonpoint source pollution. In: Selim, H.M., Ma, L. (Eds.), Physical nonequilibrium in soils: Modeling and Application, Ann Arbor Press, Chelsea, MI, 763-774.

[36] Jones, J.E. and Woodward C.S. (2001), Newton-Krylov-Multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems, *Advances in Water Resources, 24*, 763-774.

[37] Kavetski, D., Binning, P., and Sloan, S.W. (2002), Noniterative time stepping schemes with adaptive truncation error control for the solution of Richards equation, *Water Resources Research, 38(10)*, 1211.

[38] Keim, R.F., Tromp-van Meerveld H.J., and McDonnell, J.J. (2006), A virtual experiment on the effects of evaporation and intensity smoothing by canopy interception on subsurface stormflow generation. *Journal of Hydrology, 327*, 352-364.

[39] Kelley, C.T. (1995), Iterative methods for linear and nonlinear equations, SIAM, Philadelphia.

[40] Kershaw D.S. (1981), Differencing of the diffusion equation in Lagrangian hydrodynamic codes. *Journal of Computational Physics, 39*, 375-95.

[41] Kinouchi, T., Kanda, M. and Hino, M. (1991), Numerical simulation of infiltration and solute transport in an s-shaped model basin by a boundary-fitted grid system, *Journal of Hydrology, 122*, 373-406.

[42] Kitahara, H. (1992), Property of soil pipe flow in the forest, *Journal of Japan society of Hydrology & water Resources, 5*, 15-25(In Japanese).

[43] Knoll, D.A. and Keyes, D.E. (2004), Jacobian-free Newton-Krylov methods: a survey of approaches and applications , *Journal of Computational Physics, 193*, 357-397.

[44] Kogaki, T. (1999), Finite difference schemes in generalized coordinates and their application to large eddy simulation. Ph.D. thesis, University of Tokyo. (In Japanese)

[45] Koo, M. and Leap, D.I. (1998), Modeling three-dimensional groundwater flows by the body-fitted coordinate (BCF) method: I. stationary boundary problems, *Transport in Porous Media, 30*, 217-239.

[46] Koo, M. and Leap, D.I. (1998), Modeling three-dimensional groundwater flows by the body-fitted coordinate (BCF) method: II. free and moving boundary problems, *Transport in Porous Media, 30*, 345-362.

[47] Kosugi, K. (1994), Three-parameter lognormal distribution model for soil water retention. *Water Resources Research, 30*, 891-901.

[48] Kosugi, K. (1996), Lognormal distribution model for unsaturated soil hydraulic properties. *Water Resources Research, 32*, 2697-2703.

[49] Kosugi, K., Uchida, T., and Mizuyama, T. (2004), Numerical calculation of soil pipe flow and its effect on water dynamics in a slope, *Hydrological Processes, 18*, 777-789.

[50] Kotakemori, H., Hasegawa, H., and Nishida, A. (2005), Performance evaluation of a parallel iterative method library using OpenMP, *In Proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region,* 432-436, Beijing, November, 2005.

[51] Lakner, M., Plazl, I. (2008), The finite differeces method for solving systems on irregular shapes, *Computers and Chemical Engineering, 32*, 2891-2896.

[52] Lehmann, F. and Ackerer, Ph. (1998), Comparison of iterative methods for improved solutions of the fluid flow equation in partially saturated porous media, *Transport in Porous Media, 31*, 275-292.

[53] Li, M., Cheng, H., and Yeh, G. (2000), Solving 3D subsurface flow and transport with adaptive multigrid, *Journal of Hydologic Engineering, 5*, 74-81.

[54] Liang, W., Kosugi, K., and Mizuyama, T. (2009), A three-dimensional model of the effect of stemflow on soil water dynamics around a tree on a hillslope, *Journal of Hydrology, 366*, 62-75.

[55] Maliska, C.R. and Raithby, G.D. (1984), A method for computing three dimensional flows using non-orthogonal boundary-fitted coordinates, *International Journal for Numerical Methods in Fluids, 4*, 519-537.

[56] Mansell, R.S., Liwang, Ma., Ahuja, L.R., and Bloom, S.A. (2002), Adaptive grid refinement in numerical models for water flow and chemical transport in soil: a review, *Vadose Zone, 1*, 222-238.

[57] Manzini, G., and Ferraris, S. (2004), Mass-conservative finite volume methods on 2-D unstructured grids for the Richards' equation, *Advances in Water Resources, 27*, 1199-1215.

[58] Mastin, C.W. (1998), Truncation error on structured grids. In: Tomson J.F., Soni B.K., Weatherill N.P., editors. Handbook of Grid Generation. CRC Press LLC, Florida, USA, 32-1–10.

[59] Menziani, M., Pugnaghi, S., and Vincenzi, S. (2007), Analytical solutions of the linearized Richards equation for discrete arbitrary initial and boundary conditions, *Journal of Hydrology, 332*, 214-225.

[60] Morinishi, Y., Lund, T.S., Vasilyev, O.V., and Moin, P. (1998), Fully conservative higher order finite difference schemes for incompressible flow, *Journal of computational physics, 143*, 90-124.

[61] Mualem, Y. (1976), A new model for predicting hydraulic conductivity of unsaturated porous media, *Water Resources Research, 12*, 513-522.

[62] Murray, R.J. and Reason C.J.C. (2001), A curvilinear version of the Bryan-Cox-Semtner Ocean Model and Its Representation of the Artic Circulation, *Journal of computational physics, 171*, 1-46.

[63] Nash, J.E. and J.V., Sutcliffe (1970), River flows forecasting through conceptual models, part 1 a discussion of principles, *Journal of Hydrology, 27(3)*, 282–290.

[64] Neuman, S.P. (1973), Saturated-unsaturated seepage by finite elements, *Journal of Hydraulics Division, ASCE,* HY12, 2233-2250.

[65] Paniconi, C., Aldama, A.A., and Wood, E.F. (1991), Numerical evaluation of iterative and noniterative methods for the solution of the nonlinear Richards equation, *Water Resources Research, 27(6)*, 1147-1163.

[66] Paniconi, C. and Putti, M. (1994), A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems, *Water Resources Research, 30(12)*, 3357-3374.

[67] Parlange, J. (1972), Theory of water movement in soils, 6, Effect of water depth over soil, *Soil Science Society of America, 113*, 308-312.

[68] Parissopoulos, G.A., and Wheater, H.S. (1988), On numerical errors associated with the iterative alternating direction implicit (IADI) finite difference solution of the two dimensional transient saturated-unsaturated flow (Richards) equation. *Hydrological Processes, 2*, 187–201.

[69] Peaceman, D.W., and Rachford, H.H. (1955), The numerical solution of parabolic and elliptic differential equations, *Journal of the Society for industrial and Applied Mathematics, 3(1)*, 28–41.

[70] Peric, M. (1994), A finite volume method for the prediction of three-dimensional fluid flow in complex ducts, Ph.D. thesis, University of London, Imperical College.

[71] Perrens, S.J., and Watson, K.K. (1977), Numerical analysis of two-dimensional infiltration and redistribution, *Water Resources Research, 13*, 781–790.

[72] Philip, J.R. (1957), The theory of infiltration, 1, The infiltration equation and its solution, *Soil Science Society of America, 83*, 345-357.

[73] Pruess, K. and Wang, J.S.Y. (1987), Numerical modeling of isothermal and non-isothermal flow in unsaturated fractured rock–a review, *Geophysics Monograph, 42*, 11–22.

[74] Rathfelder, K., and Abriola, L.M. (1994), Mass conservative numerical solutions of the head-based Richards equation, *Water Resources Research, 30(9)*, 2579–2586.

[75] Richards, L.A. (1931), Capillary conduction of liquids of through porous medium, *Physics*, *1*, 318–333.

[76] Rubin, J. (1968), Theoretical analysis of two-dimensional transient flow of water in unsaturated and partly unsaturated soil, *Soil Science Society of America, 32*, 610–615.

[77] Ruhaak, W., Rath, V., Wolf, A. and Clauser, C. (2008), 3D finite volume groundwater and heat transport modeling with non-orthogonal grids, using a coordinate transformation method, *Advances in Water Resources, 31*, 513-524.

[78] Saad, Y. (1996), Iterative methods for sparse linear systems, PWS Publishing Company, Boston.

[79] Saad, Y., Schultz, M.H. (1986), a generalized minimal residual algorithm for solving nonsymetric linear systems, *SIAM Journal on scientific and statistical computing, 7*, 856–869.

[80] Shiraki, K. (1998), A new method for simulating water flow in hillslope soils using coarse space grids, *Journal of Japan society of Hydrology & water Resources, 11*, 586-598(In Japanese).

[81] Šimůnek, J., Sejna, M., Van Genuchten, M.T. (1999), HYDRUS-2D:simulating water flow and solute transport in two-dimensional variably saturated media, Tech, Res., IGWMC, Golden, CO, USA.

[82] Šimůnek, J., Nick, J.J., Van Genuchten, M.T., and Gärdenäs, A. (2003), Review and comparison of models for describing non-equilibrium and preferential flow and transport in the vadose zone, *Journal of Hydrology, 272*, 14-35.

[83] Tachikawa, T., Nagatani, G. and Takara, K. (2004), Developement of stage-discharge relationship equation incorporating saturated-unsaturated flow mechanism, *Annual Journal of Hydraulic Engineering, JSCE, 48*, 7–12 (in Japanese).

[84] Takasao, T. and Shiiba, M. (1988), Incorporation of the effect of concentration of flow into the kinematic wave equations and its applications to runoff system lumping, *Journal of Hydrology, 102*, 301–322.

[85] Tani, M. (2008), Analysis of runoff-storage relationships to evaluate the runoff-buffering potential of a sloping permeable domain, *Journal of Hydrology, 360*, 132-146.

[86] Terajima, T., Sakamoto, T., Shirai, T. (2000), Structure and flow phases in soil pipes developing in forested hillslopes underlain by a quaternary sand-gravel formation, Hokkaido, northern main island in Japan, *Hydrological Processes, 14*, 713–726.

[87] Thompson, J.F., Soni, B.K., and Weatherill, N.P. (1999), Handbook of grid generation, USA: CRC Press LLC.

[88] Tocci, M.D., Kelley, C.T., and Miller, C.T. (1997), Accurate and economical solution of the pressure-head form of Richards' equation by the method of lines, *Advances in Water Resources, 20*, 1–14.

[89] Tsutsumi, D., Sidle, R.C., and Kosugi, K. (2005), Development of a simple lateral preferential flow model with steady state application in hillslope soils. *Water Resources Research, 58*, 20–30.

[90] Uchida, T., Kosugi, K., and Ohte, N. (1995), A experimental study on the influence of pipe flow on water surface profile in slope. *Transactions of the Japanese Ferestry Society, 106*, 505–508 (In Japanese).

[91] Van Genuchten, M. A. (1980), Closed-form equation for predicting the hydraulic conductivity of unsaturated soils, *Soil Science Society of America, 44*, 892-898.

[92] Vauclin, M., Khanji, D., and Vachaud, G. (1979), Experimental and numerical study of a transient, two-dimensional unsaturated-saturated water table recharge problem, *Water Resources Research, 15(5)*, 1089–2001.

[93] Watanabe, M., Etoh, T., and Murota, A. (1989), A method of numerical analysis of sewage transient flow with lateral pipes. *Journal of Hydraulic, Coastal and Environmental Engineering, 411*, 81–90 (In Japanese).

[94] Weeks, S.W., Sander, G.C., Braddock, R.D. and Matthews, C.J. (2004), Saturated and unsaturated water flow in inclined porous media, *Environmental Modeling and Assessment, 9*, 91-102.

[95] Wesseling, P. (2001), Principle of computational fluid dynamics, Berlin: Springer.

[96] Zang, Y., Street, R.L., and Koseff, J.R. (1994), A non-staggered grid, fractional step method for time-dependent incompressible Navier-Stokes equations in curvilinear coordinate, *Journal of Computational Physics, 114*, 18-33.