Vrije Universiteit Brussel
Faculty of Applied Sciences
Laboratory of Hydrology

MODELING THREE-DIMENSIONAL GROUNDWATER FLOW AND TRANSPORT BY HEXAHEDRAL FINITE ELEMENTS

By A
Mohammed Adil SBAIA

Thesis submitted in fulfillment of the requirements for the award of the degree of Doctor of Philosophy in
Applied Sciences

September, 1999 A
‘Say ’O my lord! advance me in knowledge’

The Holy Quran, chapter 16, verse 114.
Dedication

To my lovely mother, and all my teachers.
Author’s Thesis Committee:

**Promoter:** Prof. F. De Smedt (*Free University of Brussels*)

**Members of the Jury:**

- Prof. R. Ababou (*University of Toulouse*)
- Prof. W. Bauwens (*Free University of Brussels*)
- Prof. A. Larabi (*MohammedyVy University of Rabat*)
- Prof. J. Vereecken (*Free University of Brussels*)
- Prof. K. Walraevens (*University of Ghent*)
- Prof. A. Van der Beken (*Free University of Brussels*)

Member!
Member!
Member!
Vice-Chairman!
Member!
Chairman!
Acknowledgments


My! deep! gratitude! goes! to! my! promoter! Prof.! Dr.! ir.! F.! De! Smedt,! chairman! of! the! Inter! University! Program! of! Water! Resources! Engineering! (IUPWARE),! for! his! unconditional! contribution! to! the! success! of! this! thesis,! by! giving! moral! financial! and! human! support.! I'm! indebted! to! his! wide! scientific! skills,! talent,! kindness,! and! patience.! He! remains! to! me! the! first! source! of! inspiration! during! this! study! leave,! by! sharing! his! knowledge! with! me,! all! his! former! students! and! co-workers! !

I'm! very! thankful! to! Prof.! Dr.! ir.! A.! van! der! Bekken,! director! of! the! Laboratory! of! Hydrology! of! the! positive! support! and! encouragements! throughout! my! Ph.D.! roads! !

Extended! thanks! to! Prof.! Dr.! ir.! A.! Larabli! from! M.! V.! University,! Ecole! Mohamid! Ingénieurs,! Hydrogeology! Section,! Rabat,! Morocco! As! a! former! Ph.D.! at! the! Laboratory! of! Hydrology! he! inspired! my! skills! at! many! stages! and! was! always! available! and! eager! to! help! ! I! appreciated! gratefully! his! sense! of! criticism! on! the! contents! of! a! primary! version! of! this! thesis! dissertation! which! he! kindly! accept! to! revise! !

My! thanks! to! all! the! members! of! the! Jury! for! accepting! to! be! a! part! of! the! author's! thesis! committee! and! for! providing! many! suggestions! and! useful! comments! which! surely! rises! the! standard! of! this! thesis! ! Any! remaining! mistakes! are! my! own! responsibility! !
I’m so afraid to forget somebody’s name! of all IUPWARE! technical! and! scientific! staff,! and! Ph.D.’s! I have! meet! since! my! arrival! in! July! 1995! Thanks! to! all! of! you! guys! for! your! help! and! making! the! working! environment! as! pleasant! as! possible.

Finally, my heartfelt thanks! goes! to! all! members! of! my! family! for! their! love! and! support! My! parents! moral! principles! and! philosophy! was! the! first! catalyst! to! end! up! four! years! of! hard! work! Staying! abroad! in! difficult! conditions! while! away! days! (and! nights)! looking! for! a! particular! problem! solution! frustrated! over! extended! periods! of! time! combined! with! lack! of! sleep! too! much! coffee! and! heat burn! pizza! was! the! price! to! pay! for! this! Ph.D.! degree.

M.AA. SBAI
Brussels, September 1999
Abstract

Modeling Three-dimensional Groundwater Flow and Transport by Hexahedral Finite Elements

By
Mohammedy Adily SBAIy

Doctor of Philosophy in Applied Sciences
Free University of Brussels
Faculty of Applied Sciences

This research work deals with three-dimensional modeling of groundwater flow and solute transport problems in groundwater aquifer systems with several complexities, heterogeneities, and variable conditions. In the field, Finite element methods are used throughout to solve a range of different problems, using in particular the Galerkin weighted residual approach based on trilinear hexahedral elements.

Special emphasis is made on transient and non-linear groundwater flow problems with moving interfaces, such as the water table and the freshwater-saltwater sharp interface. A generalized Fast Updating Procedure technique is developed for these situations, which presents a number of advantages over existing solutions. In comparison to classical computational techniques, it leads to rapid and efficient solutions. One of the important contributions is the automatic construction of the generic soils characteristic curves, which are dynamically dependent upon the overall system status. Several test examples are successfully worked out for validating this technique in different aquifer configurations, and under different initial and boundary conditions. These test cases show that the proposed method is cheap, numerically stable and accurate. Numerical stability is guaranteed through a developed solver, which is obtained by using a state of the art
methods! for! robust! preconditioning! and! efficient! numerical! implementation. ! The! accuracy! is! demonstrated! by! comparison! against! analytical! approaches! and! experimental! solutions. !

The! usefulness! of! the! method! is! clearly! shown! by! the! application! of! the! 3-D! sharp! interface! finite! element! model! "GEO-SWIM"! to! the! coastal! aquifer! system! of! Martil! in! the! north! of! Morocco. ! Several! efficient! runs! are! made! leading! to! a! calibrated! management! model! for! the! study! area! giving! a! clear! picture! of! the! salinization! risk! in! the! aquifer! due! to! saltwater! encroachment. !

Three-dimensional! modeling! of! solute! transport! problems! in! groundwater! aquifer! systems! is! equally! investigated. ! It! is! concluded! that! the! standard! Galerkin! finite! element! method! is! computationally! intensive! since! the! obtained! system! of! numerical! equations! is! very! large! sparse! none! symmetric! and! usually! difficult! to! solve! with! standard! iterative! techniques. ! Hence! preconditioning! is! necessary! to! improve! the! convergence! behavior! of! ill-conditioned! systems. ! In! this! work! we! propose! an! M! matrix! type! of! transformation! on! the! general! transport! matrix! which! guarantees! the! existence! of! the! preconditioning! schemes! and! hence! improves! the! overall! solvers! performance! and! robustness. ! The! usefulness! of! the! method! is! demonstrated! by! solving! several! test! examples! with! different! complexities! including! hypothetical! and! field! applications! in! Belgium. ! Different! solvers! are! tested! as! the! minimal! residual! method! and! the! stabilized! biconjugate! gradient! method! in! combination! with! different! preconditioning! schemes! as! diagonal! scaling! and! incomplete! factorization. ! It! is! concluded! that! M! matrix! pre- conditioning! is! very! simple! to! implement! and! proves! to! be! very! efficient! and! robust. !

An! effort! is! put! on! packaging! the! computer! programs! by! giving! modern! visual! support! to! many! modules! Therefore! several! GUI! programs! are! provided! as! complementary! tools! to! support! the! developed! models! enabling! their! friendly! use! and! the! possibility! for! future! extensions.!
Contents

List of Figures xvi

List of Tables xivii

List of Mathematical Notations and Symbols xxiii

List of Abbreviations xxviii

13 Introduction 13

1.1 General y. .................................................. y 1
1.2 Problem Definition y ........................................ y 2 y
1.3 Scope and Objectives y y ................................. y 6 y
1.4 Organization y ................................................ y 7 y

23 An Introduction to Computational Hydrogeology 113

2.1 Introduction y ................................................. y 11
2.2 What is Computational Hydrogeology y y ................ y 12 y
2.3 Classification of Groundwater Flow and Transport Models y.y.y.y 13y
  2.3.1 Physical Models y y ........................................ y 14 y
  2.3.2 Analytical Models ........................................ y 15 y
  2.3.3 Numerical Models ........................................ y 15 y
2.4 The Computer Modeling Paradigm y .......................... y 16 y
  2.4.1 Conceptual Model Development y ....................... y 16 y
2.4.2 y Model Selection y y ................................. y 17 y
2.4.3 y Mesh Generation y ................................. y 18 y
2.4.4 y Boundary Conditions y .............................. y 20 y
2.4.5 y Model Parameters y ................................. y 21 y
2.4.6 y Model Run y ................................. y 21 y
2.4.7 y Calibration and Sensitivity y Analysis y y ................................. y 22 y
2.4.8 y Interactive Scientific Visualization y y ................................. y 22 y
2.4.9 y Verification and Prediction y ................................. y 23 y
2.5 y Structured Meshes Generation y ................................. y 23 y
2.5.1 Bound y Fitted Meshes y ................................. y 23 y
2.5.2 y Multiblock Boundary y Fitted Grids y ................................. y 25 y
2.6 y The Finite Element Method y ................................. y 26 y
2.6.1 General y ................................. y 26 y
2.6.2 y Basic Concepts y ................................. y 27 y
2.6.3 y The Galerkin Weighted Residual Method y ................................. y 28 y
2.6.4 y Basis and Weighting Functions y ................................. y 29 y
2.6.5 y Numerical Integration y ................................. y 29 y
2.7 y Advanced Finite Element Methods y ................................. y 31 y
2.7.1 Mixed Finite Element Method y ................................. y 31 y
2.7.2 y Control Volume Finite Element Method y ................................. y 32 y
2.8 y Iterative Methods y ................................. y 33 y
2.8.1 Iterative Methods y versus Direct Methods y y ................................. y 33 y
2.8.2 y A Basic Iterative Method y ................................. y 35 y
2.8.3 y Linear Symmetric and Nonsymmetric Systems y ................................. y 36 y

Numerical Formulation of Mathematical Groundwater Flow Mod-3
els

3.1 y Introduction y ................................. y 37 y
3.2 y Governing Equations for 3-D Groundwater Flow y ................................. y 38 y
3.2.1 Basic Equations y .............................. y 38 y
3.2.2 ySaturated Groundwater Flow y .......................... y 39 y
3.2.3 yUnsaturated Groundwater Flow y .......................... y 40 y
3.2.4 ySteady State Groundwater Flow y .......................... y 41 y
3.2.5 yUnconfined Groundwater Flow y .......................... y 42 y
3.2.6 yConstitutive Relationships y .............................. y 42 y
3.2.7 yBoundar y and Initial Conditions .......................... y 45 y
3.3 y Governing Equations for Saltwater Intrusion y .......................... y 47 y
3.3.1 Basics y ........................................ y 47 y
3.3.2 y They Multiphase Sharp Interface Approach y.y.y .......................... y 48 y
3.3.3 y A Simplified Approach y .................................. y 50 y
3.4 y Application of The Finite Element Method y.y.y .......................... y 51 y
3.4.1 Galerkin Spatial Approximation y .......................... y 51 y
3.4.2 y Finite DifferenceyApproachyimensionalyTimey.y.y.y .......................... y 52 y
3.4.3 y Numerically Implement by Boundary Conditionsy.y.y .......................... y 54 y
3.5 y Finite Element Matrix Anal sis y .............................. y 57 y
3.5.1 y Properties of the General Matrix .......................... y 58 y
3.5.2 y Sparse Matrix Storage Scheme y .......................... y 59 y
3.6 y Solution Strategies y ........................................ y 61 y
3.6.1 y The Moving Mesh Method y .............................. y 62 y
3.6.2 y The Fixed Mesh Method y .............................. y 63 y

4 Modeling3-D3Transient3-Variably3-Saturated3-Groundwater3-Flow3

with3-Moving3-Interfaces3 653

4.1 Introduction y ........................................ y 65 y
4.2 y The FUP Numerical Technique y .......................... y 68 y
4.2.1 Determination of Idealized Relative Hydraulic Conductivity 68 y
4.2.2 y Idealized Water Retention Curve y .......................... y 70 y
4.3 y Numerical Solver y ........................................ y 75 y
5 A 3-D Sharp Interface Approach for Modeling Seawater Intrusion into Coastal Aquifers

5.1 Introduction y 1053
5.2 y Conceptual Model y 105
5.3 y The Generalized FUP Approach y 109
5.3.1 Relative Hydraulic Conductivity y 110
5.3.2 y Water Retention Curve y 110
5.3.3 y Water Solution Procedure y 111
5.3.4 y Other Features of the Proposed Approach y 112
5.4 Validation and Application Examples y 113
5.4.1 Seawater Intrusion in a Confined Aquifer y 114
5.4.2 y Seawater Intrusion in an Unconfined Aquifer y 116
5.4.3 y Plan Seawater Intrusion on Confined Artificially Recharged System y 121
5.4.4 y Saltwater Intrusion in a Multilayered Aquifer System y 126
5.4.5 y Moving Saltwater Intrusion into an 3-D Laboratory Sandy Box Model y 128
5.5 Model Application to Seawater Intrusion in a Martily Aquifer y.y.y y 130 y
  5.5.1 General Situation and Background y ............................. y 30 y
  5.5.2 y Data Analysis y ............................................. y 32 y
  5.5.3 y Construction of the Conceptual Model y ........................... y 34 y
  5.5.4 y Model Application and Results y ................................ y 38 y
  5.6 Summary y ............................................................. y 42 y

6 Finite Element Modeling of Three-Dimensional Transport Using M-matrix Preconditioners and Nonsymmetric Solvers 1453
  6.1 Introduction y ......................................................... y 45 y
  6.2 Theory y ............................................................... y 47 y
    6.2.1 Governing Equations y ......................................... y 147 y
    6.2.2 y Initial and Boundary Conditions y ............................. y 49 y
  6.3 Numerical Model y .................................................... y 50 y
    6.3.1 The Conforming Finite Element Method y ......................... y 50 y
    6.3.2 y Iterative Solvers y ............................................ y 51 y
    6.3.3 y Preconditioning Methods y ................................... y 54 y
  6.4 Results and Discussion y ............................................. y 57 y
    6.4.1 y Test Problem 1: y Continuous Point Injection y Uniform y
      Flow Field y ........................................................ y 58 y
    6.4.2 y Test Problem 2: y Steady State Transport y Radial Flow y
      Field with Counter Dispersion y .................................. y 60 y
    6.4.3 y Test Problem 3: y Transient Transport y Radial Velocity y
      Field y .................................................................. y 63 y
    6.4.4 y Test Problem 4: First Field Example y ............................ y 65 y
    6.4.5 y Test Problem 5: Second Field Example y ......................... y 67 y
  6.5 Summary y ................................................................. y 71 y

7 Software Development and GUI for Models Support 173
7.1 Introduction ............................................................. 773
7.2 yGeneral Overview .................................................... 774
7.3 yGEO-SWIM Architecture .......................................... 775
    7.3.1 Design Goals .................................................. 775
    7.3.2 yStructure .................................................... 777
7.4 yVisualization Tools ................................................. 779

83 Conclusions and Recommendations ............................... 1853
  8.1 Conclusions ....................................................... 1855
    8.1.1 Variable Saturated Groundwater Flow Problems ............ 1855
    8.1.2 ySaltwater Intrusion Problems ................................ 1877
    8.1.3 ySolute Transport Problems ................................ 1888
  8.2 yRecommendations ................................................ 1899

References ................................................................. 193

Appendices ................................................................. 2053

A3 Analytical Solutions For Transient Seepage ..................... 2053
  A.1 Linearization Techniques ........................................ 2055
  A.2 yPolubarinova-Kochina’s series functions ..................... 2066

B3 Mueller’s Method ..................................................... 2093
  B.1 Synopsis .......................................................... 2099
  B.2 yDescription ..................................................... 2099
  B.3 yAlgorithm ....................................................... 2101
  B.4 yConvergence criteria ........................................... 211
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1A</td>
<td>Different sources of groundwater contamination (Fetter, 1998).</td>
<td>A 3</td>
</tr>
<tr>
<td>2.1A</td>
<td>Definition of computational hydrogeology.</td>
<td>A13</td>
</tr>
<tr>
<td>2.2A</td>
<td>Simplified flow chart representation of the computer modeling paradigm.</td>
<td>A17</td>
</tr>
<tr>
<td>2.3A</td>
<td>Structured mesh (left) and unstructured mesh (right) for the same physical domain.</td>
<td>A19</td>
</tr>
<tr>
<td>2.4A</td>
<td>Example of a single multiblock grid component.</td>
<td>A25</td>
</tr>
<tr>
<td>2.5A</td>
<td>Hexahedral finite element.</td>
<td>A28</td>
</tr>
<tr>
<td>2.6A</td>
<td>Isoparametric hexahedral element.</td>
<td>A30</td>
</tr>
<tr>
<td>2.7A</td>
<td>Control volumes versus finite element cells for 2-D triangulations.</td>
<td>A33</td>
</tr>
<tr>
<td>3.1A</td>
<td>Representative unsaturated hydraulic conductivity curves for given soil types.</td>
<td>A43</td>
</tr>
<tr>
<td>3.2A</td>
<td>Hysteresis effects in the wetting and drying cycles of the unsaturated hydraulic conductivity (After Fetter, 1998).</td>
<td>A44</td>
</tr>
<tr>
<td>3.3A</td>
<td>Schematic representation of the saltwater intrusion sharp interface approach.</td>
<td>A48</td>
</tr>
<tr>
<td>3.4A</td>
<td>Three-dimensional control volume contribution of the element at the node i.w.</td>
<td>A53</td>
</tr>
<tr>
<td>3.5A</td>
<td>Example of a partially penetrating pumping well withdrawing water over a portion of the filtered part.</td>
<td>A63</td>
</tr>
</tbody>
</table>
4.1 (a) Location of flow at turbulent between two nodes, wand (b) idealized water relative hydraulic conductivity curve versus nodal pressure heads (note that \( k_{ij} = \varepsilon \) only if both nodes are unsaturated). A69 A

4.2A Water retention curves for specific soil types, and errors in the aer retention capacity tangent approximation (modified from Istock, 1989). A71 A

4.3A Illustrative example of mapping saturated, unsaturated and partially saturated nodes from a partially saturated element. A73 A

4.4A Idealized (a) water retention curve, and (b) analytic differentiation of water slope tangent at nonlinear first iterate. A74 A

4.5A Schematic view of a draining vertical soil column. A83 A

4.6A Water level drawdown \( \zeta(t) \); computed results with the FUP were plotted versus analytical solution. A84 A

4.7A Computed sharp water ater table height \( \zeta(t) \) using the moving mesh technique versus analytical results. A85 A

4.8A Water content profiles at specific target times, w dots indicate nodal positions. A86 A

4.9A Schematic view of a soil column draining through a leaky outlet. A88 A

4.10 Discharge rate versus head drop relationship, numerical versus analytical results. A89 A

4.11 Water level drawdown \( \zeta(t) \), FUP and moving mesh numerical methods are compared to the analytical solution. A90 A

4.12 Water content profiles at target times. A92 A

4.13 Schematic view of test problem 3. A93 A

4.14 Finite element mesh for test problem 3. A94 A

4.15 Comparison of moving numerical and analytical water tables, contour water head with an interval of 0.05 m were also plotted. A95 A

4.16 Configuration of the laboratory model showing the finite element mesh, the photographed sections, and the core dam location. A98 A
4.17 Comparison of predicted (continuous lines) and observed free surface during rise and steady state for the homogeneous dam: \( w(F) \) vs. section; \((B_1,B_2)\) Back sections; \((S)\) side section. 

4.18 Comparison of predicted (continuous lines) and observed free surface during drawdown for the homogeneous dam: \( w(F) \) vs. section; \((B_1,B_2)\) Back sections; \((S)\) side section. 

4.19 Comparison of predicted (continuous lines) and observed free surface during rise and steady state for the heterogeneous dam: \( w(F) \) vs. section; \((B_{14},B_2)\) Back sections; \((S)\) side section. 

4.20 Comparison of predicted (continuous lines) and observed free surface during drawdown for the heterogeneous dam: \( w(F) \) vs. section; \((B_{14},B_2)\) Back sections; \((S)\) side section. 

5.1A Potential pressure conditions along the free boundaries; in the unsaturated, fresh and saltwater zones. 

5.2A Generalized (a) water retention curve, and (b) analytic differentiation of the slope tangents at the nonlinear first iterate. 


5.4A Comparison of Glover's analytical solution and numerical results. 

5.5A Zoom window showing the fresh-saltwater interface position in the groundwater heads (X and Z axis have the same scale). 

5.6A Convergence rate history of the modified Picard nonlinear iteration solver for test problem 1. 

5.7A Van der Veer's analytical interface problem (1977). 

5.8A Comparison between numerical and Van der Veer's analytical solution. 

5.9A Freshwater potential heads distribution for the first run. 

5.10 Freshwater potential heads distribution for the second run. 

5.11 Convergence rate history of the modified Picard nonlinear iteration solver for test problem 2 (runs 1 and 2).
5.12 Definition of the Hunt’s analytical interface problem (1985); (a) plan view, and (b) well cross-section profile. .......................... 123 A

5.13 Comparison between analytical and numerical results for the Hunt’s test problem at the aquifer bottom. Results are plotted together with the contour lines of potential heads. .......................... 125 A

5.14 Cross-section view of the computed interface and freshwater potential heads along the injection well. .......................... 125 A

5.15 A zoom view on the interface, crossing potential isolines are shown along with the outflow to the sea face. .......................... 126 A

5.16 Schematic representation and parameter values of the fourth test example (Huyakorn et al., 1996). .......................... 127 A

5.17 Comparison of the analytical and numerical solutions computed by Huyakorn et al. (1996). .......................... 127 A

5.18 Comparison of the analytical and numerical solutions computed with the GEO-SWIM code, for the fourth test problem. .......................... 128 A

5.19 Descriptive view of the experimental and box model used by Sugio and Rahim (1992). .......................... 129 A

5.20 Finite element mesh used in Sugio’s laboratory sand box model validation. .......................... 130 A

5.21 Numerical solutions versus laboratory experiments for (a) Front section, (b) back section, and (c) bottom section. .......................... 131 A

5.22 General geographic situation map of Martil aquifer. .......................... 132 A

5.23 Study area and locations of cross-sections of interest. .......................... 133 A

5.24 Interpolated contour maps of (A) phreatic aquifer, (C) aquitard and (D) confined aquifer respectively. .......................... 134 A

5.25 S-N and W-E cross-sections showing the conceptual model, showing the finite element mesh and soil types distribution. .......................... 135 A
5.26 Comparison between (a) computed and (b) observed (in 1966, before 
heavypumping)steady-state groundwater potentials presented as meters above sea level. 137 A
5.27 Shape and extent of the natural 3-D fresh-saltwater interface . . . 139 A
5.28 W-E cross-sections showing the initial groundwater heads and fresh-water 
saltwater interface position. The vertical scale magnification factor equals 100 for cross-sections C-C' and E-E', and 66.7 for F-F')w . . . 140 A
5.29 Computed moving front positions of the sharp fresh-water-saltwater inter-
face each 8 years from 1974 to 2006. The shaded surface represents the bottom of the lower aquifer. 141 A
5.30 Cross-sectional view of the seawater intrusion; simulated moving in-
terface positions from 1974 to 2006 are plotted. 142 A
5.31 Maximum interface position versus time, the graph shows evidence of overexploitation of the 
aquifer during the nineties. 143 A

6.1 Schematic representation of the test problems; (A) Injection in a uni-
form flow field; (B) Radial injection with equilibrium counter-disper-
sion; (C) Radial injection; (D) Field problem 1; and (E) Field problem 2. 156 A
6.2A Convergence history analysis of test problems (a) 1a and (b) 1b. 159 A
6.3A Comparison of analytical and numerical solutions of normalized con-
centrations versus radial distance for test examples 2a and 2b. 162 A
6.4A Convergence history analysis of test problems (a) 2a and (b) 2b. 163 A
6.5A Comparison of analytical and numerical solutions of normalized con-
centrations versus radial distance for test examples 4a and 4b. 165 A
6.6A Simulated steady-state (a) potentials, water velocities, and (b) water 
concentration iso-surfaces for test problem 4. 166 A
6.7A Convergence history analysis of test problem 4. A
6.8A An illustrative plan view of the computed head and velocity fields and the pollution plume at the base of the lower aquifer unit. A

7.1A GEO-SWIM modules. A
7.2A Visual GEO-SWIM interface: A sample front section view, mesh and soil types are overlayed. A
7.3A Flow conditioner package: Upper list shows boundary conditions for the processed case study. A
7.4A Options of GEO-SWIM solver package. A
7.5A Executable program paths interface. A
7.6A A Tab view of the GEO-SWIM to Tecplot converter GUI package. A
List of Tables

4.1A Unpreconditionedyconjugategradientiterativealgorithm(Barretty et al., y1994).y ............................................. A76 A
4.2A Preconditionedyconjugategradientiterativealgorithm(Barretty et al., y1994).y ............................................. A77 A
4.3A ComparisonoftheyFUPyandymovingmeshsolversyefficiency for ytestyproblem1.y ............................................. A87 A
4.4A ComparisonoftheyFUPyandymovingmeshsolversyefficiency for ytestyproblem2.y ............................................. A91 A
4.5A FUPysolverperformanceyforystestyproblem3.y .......... A96 A

5.1A Glover’stestyproblemphysicalyandcomputationalymodelyandparameter yvalues.y ............................................. 115 A
5.2A Physicalyandcomputationalymodelyandparametervaluesyoftheythirdy testyproblem.y ............................................. 124 A
5.3A Informationyponythicknessyoftheymodelyandlayers.y ........ 135 A
5.4A Modelycalibratedparameteryunderysteadystateconditions.y .......... 142 A

6.1A TheypreconditionedyMinimumResidualyiterativeymethod(Barretty et al., y1994).y ............................................. 152 A
6.2A TheypreconditionedyBiCGSTAByiterativeymethod(Barretty et al., y 1994).y ............................................. 153 A
6.3A Numericalyfeaturesyoftheytestyproblems.y ................. 157 A
6.4A Physical parameters of the test problems y .......................... 158 A
6.5A Solver performances for the first test problem y .......................... 161 A
6.6A Solver performances for the test problem y2b y .......................... 164 A
6.7A Solver performances for the forth test problem y .......................... 168 A
6.8A Solver performances for the fifth test problem y .......................... 170 A

A.1A Coefficients of series used in the Polubarinova-Kochina's analytical solution y .......................... 207 A
# List of Mathematical Notations and Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>,</td>
</tr>
<tr>
<td>$\psi_a$</td>
<td>Air-entry! pressure! head! value! [L]!</td>
</tr>
<tr>
<td>$b_i, b_j$</td>
<td>Basis! functions!</td>
</tr>
<tr>
<td>$\mathbf{B}, \mathbf{B}^*$</td>
<td>Boundary! conditions! vector!</td>
</tr>
<tr>
<td>$C!$</td>
<td>Concentration! of! dissolved! matter! [ML$^{-3}]$.</td>
</tr>
<tr>
<td>$\Omega_i$</td>
<td>Control! volume! of! node! [L$^3$.</td>
</tr>
<tr>
<td>$c!$</td>
<td>Normalized! concentration! [dimensionless].</td>
</tr>
<tr>
<td>$C_D$</td>
<td>Source! distribution! function! [ML$^{-3}]$.</td>
</tr>
<tr>
<td>$\mathbf{C_y}$</td>
<td>Unknown! concentration! vector! [ML$^{-3}]$.</td>
</tr>
<tr>
<td>$C_w$</td>
<td>Water! capacity! [L$^{-1}]$.</td>
</tr>
<tr>
<td>$\nabla$</td>
<td>Del! operator!</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Density! difference! ratio! [dimensionless].</td>
</tr>
<tr>
<td>$\mathbf{Dy}$</td>
<td>Diagonal! matrix!</td>
</tr>
<tr>
<td>$dV$</td>
<td>Differential! volume!</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Dynamic! viscosity! of! water! [ML$^{-1}$T$^{-1}]$.</td>
</tr>
<tr>
<td>$\mathbb{D}$</td>
<td>Hydrodynamic! dispersion! tensor! [L$^2$T$^{-1}]$.</td>
</tr>
<tr>
<td>$erfc$</td>
<td>Complementary! error! function!</td>
</tr>
<tr>
<td>$erf$</td>
<td>Error! function!</td>
</tr>
<tr>
<td>$exp$</td>
<td>Exponential! function!</td>
</tr>
<tr>
<td>$e_{max}^m$</td>
<td>Maximum! error! at! iteration! level! $m$</td>
</tr>
<tr>
<td>$\Gamma_i$</td>
<td>Finite! element! patch! of! node! i! in! the! surface! xy! plane!</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>First-order! decay! coefficient!</td>
</tr>
<tr>
<td>$\forall$</td>
<td>For! all! values! of!</td>
</tr>
<tr>
<td>$\rho_f$</td>
<td>Freshwater! density! [ML$^{-3}]$.</td>
</tr>
<tr>
<td>$g$</td>
<td>Acceleration! of! gravity! constant! ($= 9.81 m^2/s$).</td>
</tr>
</tbody>
</table>
\[(x, y, z)\] A Global!coordinate!system!/[L]!
\(G, G^*\) Global!flow!conductance!matrix!
\(G_{ij}, G_{ii}\) Global!flow!conductance!matrix!entries!
\(G^T\) Transpose!of!matrix!\(Gy\)
\(\tilde{h}\) Approximate!groundwater!potential!/[L]!
\(h_s\) Equivalent!saltwater!potential!/[L]!
\(h\) Groundwater!head!/[L]!
\(hy\) Unknown!groundwater!head!vector!
\(h_i, h_j\) Unknown!nodal!groundwater!head!at!nodes\!i!and\!j!/[L]!
\(Iy\) Identity!matrix!
\(\eta(t)A\) Interface!depth!below!datum!level!at!time\!t!/[L]!
\(i_{t_{max}}\) Maximum!number!of!linear!PCG!Iterations!
\([J]A\) Jacobian!matrix!
\(K_d\) Adsorption!distribution!coefficient!/[L^3 M^{-1}]!
\(K_y\) Hydraulic!conductivity!tensor!/[LT^{-1}]!
\(k_r\) Relative!hydraulic!conductivity!tensor!/[LT^{-1}]!
\(K_y\) Saturated!hydraulic!conductivity!tensor!/[LT^{-1}]!
\(k_y\) Unsaturated!permeability!tensor!/[L^2]!
\(\delta_{ij}\) Kronecker!symbol!
\(K'(G, r^0)A\) Krylov!subspace!of!dimension!\!i!for\!\(Gy\) and\!\(r^0\)
\(L(.).\) Differential!operator!
\((\xi, \eta, \zeta)\) Local!finite!element!coordinate!system!/[L]!
\(\alpha_L\) Longitudinal!dispersivity!/[L]!
\(Ly\) Lower!triangular!matrix!
\(My\) Mass!transport!matrix!
\(\omega_{\text{max}}\) Maximum!underrelaxation!factor!
\(\omega_{\text{min}}\) Minimum!underrelaxation!factor!
\(n_c\) Effective!porosity!/[dimensionless]!
\(ln(.)\) A Natural!logarithm!function!
\(nn\) Number!of!nodes!
\( \mathbf{n} \) Outward unit vector perpendicular to curve tangent
\( n \) Porosity [dimensionless]
\( \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \) Partial derivatives with respect to global coordinates
\( \frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \zeta} \) Partial derivatives with respect to local coordinates
\( P_i \) Polynomial of degree \( i \)
\( \mathbf{P} \) Preconditioning matrix
\( \psi \) Pressure head [L]
\( p \) Water pressure [ML\(^{-1}\)T\(^{-2}\)]
\( q_n \) Prescribed Neuman or Cauchy flux type [LT\(^{-1}\)]
\( q \) Specific discharge rate [LT\(^{-1}\)]
\( Q \) Total inflow/outflow [L\(^3\)T\(^{-1}\)]
\( \mathbf{Q} \) Transport matrix
\( R \) Internal flow source/sink term [T\(^{-1}\)]
\( \mathbf{r} \) Residual vector
\( R \) Retardation factor [dimensionless]
\( R_c \) Solute transport source/sink [MT\(^{-1}\)L\(^{-3}\)]
\( S_C \) Cauchy type concentration boundary
\( S_{ii} \) Diagonal term of the storage matrix
\( S_1 \) Dirichlet-type boundary
\( S_D \) Dirichlet type concentration boundary
\( S_p \) Elastic storage coefficient [dimensionless]
\( S_2 \) Neuman-type boundary
\( S_N \) Neuman type concentration boundary
\( \rho_s \) Saltwater density [ML\(^{-3}\)]
\( \theta_s \) Saturated water content [dimensionless]
\( \varepsilon \) Small positive number
\( S_s \) Specific storage coefficient [L\(^{-1}\)]
\( S_e \) Storage coefficient of element \( e \)
\( \mathbf{S} \) Storage matrix
\( S_w \) Water saturation [dimensionless]
\[\sum\] Sigma!sum\!notation!

\(T_{\text{max}}\) Maximum!calculation!time![T]!

\(t_{\text{tol}}\) Prescribed!groundwater!head!tolerance![L]!

\(t_p\) Target!time!value![T]!

\(\Delta t\) Time!step![T]!

\(t\) Time!variable![T]!

\(T\) Transmissivity![L^2T^{-1}]!

\(\alpha_{TH}\) Transverse!horizontal!dispersivity![L]!

\(\alpha_{TV}\) Transverse!vertical!dispersivity![L]!

\(\omega^m\) Underrelaxation!factor!at!iteration!level!m

\(\textbf{U}_{\text{y}}\) Upper!triangular!matrix!

\(\Delta t_{\text{user}}\) User-specified!time!step![T]!

\(V_{i}^c\) Control!volume!contribution!of!element!\(e\) at!node!\(i\) [L^3]

\(V_F\) Freshwater!zone!volume![L^3]!

\(V_{\text{y}}\) Groundwater!seepage!velocity![L^2T^{-1}]!

\(V_{S}\) Saltwater!zone!volume![L^3]!

\(V\) Space!volume![L^3]!

\(z\) Elevation!head![L]!

\(\theta\) Volumetric!water!content![dimensionless]!

\(\rho\) Water!density![ML^{-3}]!
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D, 2-D, 3-DM</td>
<td>One, two, and three-dimensionalM</td>
</tr>
<tr>
<td>ANSIM</td>
<td>American National Standards InstituteM</td>
</tr>
<tr>
<td>APIM</td>
<td>Application Programming InterfaceM</td>
</tr>
<tr>
<td>BCM</td>
<td>Boundary conditionsM</td>
</tr>
<tr>
<td>BICGM</td>
<td>Biconjugate Gradient SolverM</td>
</tr>
<tr>
<td>BI-CGSTABM</td>
<td>Biconjugate Gradient Stabilized SolverM</td>
</tr>
<tr>
<td>BI-CGSTAB (l)M</td>
<td>Biconjugate Gradient Stabilized Solver of order lM</td>
</tr>
<tr>
<td>CADM</td>
<td>Computer Aided DesignM</td>
</tr>
<tr>
<td>CF90M</td>
<td>Cray Fortran 90 compiler systemM</td>
</tr>
<tr>
<td>CGM</td>
<td>Conjugate Gradient SolverM</td>
</tr>
<tr>
<td>CG-likeM</td>
<td>Conjugate Gradient-like SolverM</td>
</tr>
<tr>
<td>CGSM</td>
<td>Conjugate Gradient Squared SolverM</td>
</tr>
<tr>
<td>CPUM</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CrM</td>
<td>Courant numberM</td>
</tr>
<tr>
<td>Cray J916/8-1024M</td>
<td>Super Computer architectureM</td>
</tr>
<tr>
<td>CSRM</td>
<td>Compressed\texttt{Sparse Row} matrix storage format</td>
</tr>
<tr>
<td>CVFEM</td>
<td>Control Volume Finite Element</td>
</tr>
<tr>
<td>DCSRM</td>
<td>Diagonal Compressed\texttt{Sparse Row} matrix storage format</td>
</tr>
<tr>
<td>DSM</td>
<td>Diagonal Scaling preconditionerM</td>
</tr>
<tr>
<td>DXFM</td>
<td>Data Exchange file Format</td>
</tr>
<tr>
<td>FDM</td>
<td>Finite DifferencesM</td>
</tr>
<tr>
<td>FEM</td>
<td>Finite ElementsM</td>
</tr>
<tr>
<td>FDM</td>
<td>Finite Difference MethodM</td>
</tr>
<tr>
<td>FEM</td>
<td>Finite Element MethodM</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>FEMWATERM</td>
<td>Finite Element Ground Water flow model</td>
</tr>
<tr>
<td>FUPM</td>
<td>Fast Updating Procedure</td>
</tr>
<tr>
<td>GEO-PROFM</td>
<td>Geohydrological Professional Groundwater Flow model</td>
</tr>
<tr>
<td>GEO-SWIM</td>
<td>Geohydrological Saltwater Intrusion Model</td>
</tr>
<tr>
<td>GHM</td>
<td>Ghyben-Herzberg Model</td>
</tr>
<tr>
<td>GISIM</td>
<td>Geographical Information System</td>
</tr>
<tr>
<td>GMRESM</td>
<td>Generalized Minimal Residual Solver</td>
</tr>
<tr>
<td>GUIM</td>
<td>Graphic User Interface</td>
</tr>
<tr>
<td>HISM</td>
<td>Hydrogeological Information System</td>
</tr>
<tr>
<td>IF0M</td>
<td>Incomplete Factorization Preconditioner Model</td>
</tr>
<tr>
<td>IF0-M</td>
<td>Modified Incomplete Factorization Preconditioner of order 0z</td>
</tr>
<tr>
<td>IFDM</td>
<td>Integrated Finite Difference Method</td>
</tr>
<tr>
<td>MB</td>
<td>Mega Bytesz</td>
</tr>
<tr>
<td>MBE</td>
<td>Mass Balance Errorz</td>
</tr>
<tr>
<td>MCSR</td>
<td>Modified Compressed Sparse Row matrix storage formatz</td>
</tr>
<tr>
<td>MFEM</td>
<td>Mixed Finite Elementz</td>
</tr>
<tr>
<td>MIF0</td>
<td>Modified Incomplete Factorization preconditioner of order 0z</td>
</tr>
<tr>
<td>MIF0-M</td>
<td>Modified Incomplete Factorization Preconditioner of order 0z</td>
</tr>
<tr>
<td>MIPS</td>
<td>Multiple Instruction Processor Systemz</td>
</tr>
<tr>
<td>MOCM</td>
<td>Modified Method of Characteristicsz</td>
</tr>
<tr>
<td>MODFLOW</td>
<td>Modular groundwater flow modelz</td>
</tr>
<tr>
<td>MR</td>
<td>Minimal Residual solverz</td>
</tr>
<tr>
<td>MT3D</td>
<td>Modular Transport Three-Dimensional finite difference modelz</td>
</tr>
<tr>
<td>NIM</td>
<td>Number of IterationsM</td>
</tr>
<tr>
<td>PCM</td>
<td>Personal ComputerM</td>
</tr>
<tr>
<td>PCGM</td>
<td>Preconditioned Conjugate GradientM</td>
</tr>
<tr>
<td>PDEM</td>
<td>Partial Differential EquationM</td>
</tr>
</tbody>
</table>
Pe! Peclet Number!

Quasi-3D! Quasi Three-Dimensional!

RAM! Random Access Memory!

USGS! United States Geological Survey!
Chapter 1

Introduction

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 General</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Problem Definition</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope and Objectives</td>
<td>6</td>
</tr>
<tr>
<td>1.4 Organization</td>
<td>7</td>
</tr>
</tbody>
</table>

1.1 General

In the past few decades, modeling has become an important and powerful tool in many branches of science. Models allow engineers and scientists a way to test hypotheses in a manner that is nondestructive to the actual problem. Computer modeling has become a necessity and a meaningful way of improving our quality of life, and the experts at the academic and laboratory environment level.

The astonishing development of computer hardware and software technologies, and the significant increase of computational power, has contributed widely in solving complex engineering problems in several fields. Although, it seems that
there will be no limits for such developments, it turns out that at the same time A needs A for A future man civilizations will put more A demands, and A more delicate A om A putational A problems to be manageable by digital or A and A 'interactive' computers. A Numerical modelers A and engineers are therefore always in a challenge to design A and implement A better A strategies, in terms of both A cost A and A efficiency. A Which, at the same time may prove very useful for an improved understanding of the A world's A surrounding physical phenomena. A

1.2y Problem Definitiony

Protection of freshwater resources against contamination and toxic pollution is A of A abroad A interest A for A the A community, including modelers, practitioners A ers, A technical A experts, and A decision A makers A and A managers. A Different A sources A of A groundwater contamination are man made actions modifying the natural envi A ronment process, as illustrated in Fig. 1.1. Mathematical A modeling A plays an ever A increasing A role A in the A quantitative analysis A of A the actual behavior A of A groundwater A in terms of quantity and quality, and in the design of efficient protection and A remediation scenarios. A

Mathematical models are classified in two broad categories: stochastic and de A terministic. Stochastic models are useful when our total ignorance of the actual A situation reach a sufficient level to assume a 'statistically' random realizations A of the unknown problem parameters. A Deterministic models are by far the most A common A as today. A Systems A of A partial A differential A equations A that A relate A pa A rameters A such A as A potential A head, water A flow A and concentration A of A dissolved A matter A are commonly utilized by this type of numerical models. A

The types A of A methods commonly used for numerical A modeling are A finite A difference methods (FDM), and finite element methods (FEM). The system of equ A A tions that results A is solved A simultaneously A to determine necessary A flow parameters A
Figure 1.1: Different sources of groundwater contamination (Fetter, 1998).

At specific nodes, in this manner, the hydraulic characteristics are found at specific points throughout the system. Finite element methods, unlike finite difference methods which require orthogonal grid elements, utilize elements of various size and shape, enabling these methods to better handle irregular shapes, and complex domains with various heterogeneities, and curvilinear boundaries, allowing also for better implementation of characterized time-varying conditions. Despite these FEM advantages, FDM is still competitive, and somewhat more stable. In dealing with practical field applications, where the simulation cost becomes an important objective of the project itself, which explains the encountered success and widespread use of these kind of models.

These conclusions become more evident in case of fully three dimensional models. A three dimensional model does not only permit the simulated independent variable to be calculated everywhere in the real world physical domain, but it removes many underlying assumptions when formulating the original model. Examples of these assumptions are the well-known Dupuit horizontal flow in aquifers and vertical flow in aquitards, leading to quasi-3D models; and the Baden-Ghyben (1888) and Herzberg (1901) approximation for the salt-freshwater interface near
the sea coast. These assumptions have been proven to be physically unrealistic for a full range of situations (Bear, 1972).

Moreover, for transient groundwater flow problems, the PDE’s are more delicate to solve, since four independent variables are involved (space coordinates and time). In unconfined aquifers involving a variably saturated flow, a traditional model cannot encompass many additional parameters related to the soil characteristics, most of these parameters are found using some mathematical niceties, for which it is often hard to find a physically based interpretation. These limitations shorten the applicability of this kind of models, especially for groundwater systems with a high variability of soils and heterogeneities. Unsaturated systems often involve variable and moving boundaries in time, like the water table de-limiting the saturated and variably saturated zones, the salt-freshwater interface separating the assumed immiscible freshwater and saltwater zones.

Saltwater Intrusion, or encroachment, defined by Freeze and Cherry (1979) as the migration of saltwater into freshwater aquifers under the influence of groundwater development, becomes a problem in coastal areas where freshwater aquifers are hydraulically connected with seawater. When large amounts of freshwater are withdrawn from these aquifers, hydraulic gradients encourage the flow of seawater toward the pumped wells. A common goal of these models is to predict and characterize the movement of the transition zone in the aquifer where freshwater and saltwater meet. Another purpose of modeling is to predict the behavior, degree, and extent of mixing that occurs in this transition region. These are two simple examples of how models are used to predict the future conditions of situations that may actually take many years to occur.

Mathematical groundwater models of this class are complex, and the existing numerical approaches are not applicable for field applications of large dimensions requiring high resolution. Most of the existing three-dimensional saltwater intrusion models are miscible density dependent solute transport models, which face
data availability problems and a need of large amount of computer hardware re-A
sources. Two-phase sharp interface models are 2-D or at most quasi-3D, becauseA
the traditional approach requiring the solution of coupled system of freshwaterA
and saltwater governing equations is very demanding. Hence, there is still prac-A
tical limitations. An application of these models to field conditions. Moreover,
boundary conditions relevant to the saltwater phase are not always easily acces-A
sible. Known in case of moving interfaces. To our present knowledge, there is noA
three-dimensional model based on a 'cheap' embedded sharp interface approachA
available.

Three dimensional modeling of solute transport is readily accessible today. ButA
still, some computational limitations are existing. For enhancing, the scope ofA
the numerical methods known to be standard. An use for this type of prob-A
lems. Indeed, finite element discretizations lead to large, sparse matrices whichA
are often ill-conditioned and hence an iterative solution method even when it isA
robust may fail to provide an acceptable solution. For transport problems theseA
difficulties are augmented by the possibility of other typical errors arising in theA
computational process, such as numerical dispersion, numerical oscillations andA
overshoots leading to measuring concentrations fronts. Different strategies haveA
been adopted in the past decades to exclude or minimize at best these numericalA
errors. We provide here another methodology based on a robust numerical pre-A
conditioning of the FE algebraic equations, based on physical interpretation ofA
the different terms in the numerical formulation leading to an anticipated massA
conservation of the numerical process.

For most modeling applications, the time and effort spent on pre-processingA
and post-processing data far exceed that spent on other project activities. ThisA
work can be tedious for three dimensional simulations which require more inputA
data and more nicely presented results in easily understood graphic form. ThisA
explains the new trends towards standardization of graphic interface tools, andA
the motivation for developing comprehensive visual interfaces for a more interactivity and friendly use.

1.3y Scope and Objectives

This work focuses on contributing to computationally efficient three dimensional groundwater flow and solute transport models and additional set of tools based on new techniques and methodologies, which enhance significantly standard concepts known at present. Special emphasis is made on transient and nonlinear problems, which are more difficult to investigate and model effectively. Before a detailed exposition of the main objectives taken into account in this research, we still separate the investigated problems into three related categories: (i) transient and variably saturated groundwater flow with a moving water table, (ii) the same but including a moving fresh-saltwater interface, and (iii) solute transport problems.

Respectively, the objectives for each of these three main topics are as follows:

1. Develop a robust and efficient time dependent solver, based on linearized equations. The trade-off between the global cost in memory storage, CPU requirements, and the general solution accuracy should be quite reasonable;

2. The linearization process should be mass conservative, smooth, and efficient;

3. Provide built-in generic representation of the soils characteristic curves, to relax the numerical solution procedure. It should use a minimum set of soil physical parameters;

4. Testing of the codes efficiency and robustness under various platforms and with different test examples, to ensure numerical accuracy;

5. Application of the computer programs to study the actual behavior, and future management plans of selected field sites. These include an aquifer...
system from Morocco for seawater intrusion analysis, and two field sites from Belgium subject to groundwater pollution.

6. Development of user interface program packages to be used as a front end to the simulation codes.

One of the important addressed goals in the development stage was the code portability issue, such that the model will run on any hardware provided that a Fortran compiler is available. To achieve this goal, we developed a Fortran 90A compatible code with hardware independent compiler directives. The code has been tested by running test examples including academic, hypothetical and field scenarios on the following platforms:

- Cray J916/8-1024 supercomputer (UNIX-based UNICOS 9.0.2.1 and CF90A Fortran compiling systems)
- Silicon Graphics Challenge-L workstation (UNIX-based IRIX 5.3 and MIPS A Fortran90 compiler)
- Sun UltraSparc workstation (UNIX-based SOLARIS 5.2.1 and SPAR-A Cworks Pro Fortran compiler)
- Personal computers with Intel Pentium processors (MS DOS-Win95/Win98/NT), provided that a 32-bit compiler is available.

1.4y Organization

The dissertation text is organized in eight chapters including the introductory and concluding remarks chapters. Separate chapters discussing the results of the application of the methods and concepts described in other chapters are not given. Therefore, all things related to a specific topic, including the theoretical...
developments, applications and discussions are put together. An one consistent chapter.

Chapter 2 is an introduction to several aspects of computational hydrogeology, including the model classification, and the computational techniques such as the finite element method, and their practical application. Throughout the text, the corresponding literature is reviewed. This is not intended to give a complete review of the numerical methods used for modeling groundwater flow and solute transport in aquifer systems. It only introduces the reader to this field through numerous hints and by providing standard references. State of the art methods for each problem are described in the literature reviews given at the beginning of the respective chapter.

In chapter 3, the review is extended to more specific aspects of three-dimensional modeling of groundwater flow in aquifer systems. We start by a general discussion on several forms of the groundwater flow governing equations. Afterwards, a detailed exposition of the finite element matrix equations derivation is given, this constitutes the core of the chapter, and sets up the necessary foundations for numerical models. Several other aspects are discussed, such as the numerical implementation of boundary conditions and the sparse matrix storage scheme. Once the algebraic FE equations are derived, several solution methods and strategies are presented in the last paragraph, aiming to introduce the reader to the concepts and methodologies used in the next chapter.

Chapter 4 focuses on a variably saturated groundwater flow with a moving water table profile. A general discussion on the subject is followed by the conceptual development of the FUP technique, the different steps involved in an application, and how difficulties associated with several non-linearities are avoided. The assessment of the inner linear numerical solver and preconditioner are discussed in detail, and their use is justified. Other steps and procedures involving the construction of the non-linear solver and time average are presented. The FUPA based model is verified using four test examples. The first example is quite sim-
ple and describes the drainage through a vertical soil column. A second example of a basically similar to the first except that a leaky soil is placed at the exit of the soil column, retarding the vertical flow. A third example is important for the transient drainage studies in water table aquifers. A last validation sample is a most interesting aspect of this study, it shows clearly the performances and capabilities of the 3-D model. This study compares the model predictions against the laboratory experiments designed and performed by Baseghi and Desai (1987, 1990), in a 3-D earth model with several complex configurations.

Generalization of the FUP approach for problems with freshwater-seawater moving interfaces is discussed in Chapter 5. A conceptual problem correctly formulated here is more complicated problems is formulated. This is smoothly adapted from the developments already performed in Chapter 4. Here again, a series of test problems are relevant to check the numerical solution accuracy. A five examples are considered, including (i) confined and (ii) unconfined aquifers; (iii) seawater intrusion control with recharging wells; (iv) saltwater in-a-trusion in a multilayer aquifer system; and (v) a moving sharp interface as a 3-D laboratory sand box model as designed by Sugio and Rahim (1992), which is the most challenging problem. Except for the fifth test all numerical solutions are compared with available analytical solutions, additionally for the fourth test the results are compared to the numerical solution computed by Huyakorn et al. (1996). A satisfactory agreement is noticed and his overall comparative study is last paragraph is devoted to the model application to seawater intrusion in a Martil aquifer system in Morocco, for studying the actual behavior of the sharp interface, and establishing future risk to salinization scenarios.

Chapter 6 deals with numerical aspects of three-dimensional modeling of soil-lute transport problems, using the common-place Galerkin finite element technique. A model governing equations are presented, followed by their matrix FEA equivalent form. Limitations related to this approach are shown and explained. An alternative through the use of a robust numerical method is proposed. This is...
volves using the art of solvers for a particular class of symmetric systems arising from the FE discretization of the global system, namely the conjugate gradient stabilized method (Van der Vorst, 1992). This is followed by applying the best preconditioner for a particular situation, meaning the multiplicity of choice from a range of existing ones. Motivations for preferring attractive schemes are explained and reinforced by giving examples from the state of the art literature. The performances of possible mixtures of the selected solvers and preconditioners are compared based on CPU’s, the convergence behavior, and accuracy. Five problems are studied including two field situations from Belgium.

Chapter 7 describes how several of the developed models are implemented from software point of view. Thus, a brief explanation of many computer packages and additional tools is given. One should keep in mind that, this chapter is not intended to be a user manual for the software, neither a technical report explaining the development process more related to software engineering design in computer science. These detailed aspects are specifically hidden.

Finally, in chapter 8, the overall conclusions are discussed and ideas for future research are formulated.
Chapter 2G

An Introduction to

Computational Hydrogeology

Contents

2.1 Introduction .............................................. 11
2.2 What is Computational Hydrogeology? ............. 12
2.3 Classification of Groundwater Flow and Transport
   Models ..................................................... 13
2.4 The Computer Modeling Paradigm .................... 16
2.5 Structured Meshes Generation ......................... 23
2.6 The Finite Element Method ............................. 26
2.7 Advanced Finite Element Methods .................... 31
2.8 Iterative Methods ........................................ 33

2.1 Introduction

In this chapter, we attempt to present a short and concise coverage of key aspects of modeling groundwater flow and solute transport in aquifer systems. An effort
is made to mix theoretical and practical aspects together, such that it will be A
fruitful from the developer and practitioner points of view. A

We start by a short discussion on the terminology, explaining the difference A
between A subsurface hydrology modeling A and A computational hydrogeology A, which A
is An A more A general A and appealing A definition. A Next, we discuss A different A kind A of A
groundwater A flow A and A solute A transport A models. A Section A.4 describes A the A different A
steps usually involved A during A classical modeling approach, starting from the A
cceptual A model A development to A building A a reliable A predictive A model. A Emphasis A
is made on practical aspects and precautions to take in this A
process. A Section A.5 gives A brief introduction A to A structured mesh A generation A in A simple A domains, and A
the extension to A complex A domains using A multiblock A methods (Ho-Le, 1988). A The A
next section sets up A the A basics A of A the A standard A Galerkin A finite A element A method,
and A its A theory A useful throughout A the A upcoming A chapters. A Section A.7 introduces A
advanced A finite A element A methods, and A other A discretization A methods A becoming A
common A place A in A the A groundwater A community. A A Particular A features A of A these A
alternative A methods are briefly described A and A commented. A Finally, we give A
a brief A overview A of A iterative A methods, which are A the A heart A of A any numerical A solver. A

2.2y What is A computational A hydrogeology A?

Computational A hydrogeology is an emerging multi-disciplinary scientific discipline. A
It has A roots in broad branches of science as A illustrated in A Fig. A.2.1. These A are, A
Groundwater A hydrology, A applied A mathematics A and A computational A methods, and A A
computational A science. A

Additional A fields resulting from other interactions between these areas (see A
Fig. A.2.1) have A specific A orientations. A Nevertheless, they A have A several A aspects A common, and A a sharp A distinction A cannot A be A made. It is clear however, that com-A
putational A hydrogeology A has A much A broader A significance A than A modeling, because A
usually a sound understanding and A detailed know-how in several other fields are A
2.3 Classification of Groundwater Flow and Transport Models

Groundwater and transport models can be categorized into three broad categories (Anderson and Woessner, 1992; Kresic, 1997) numerical, physical, and analytical. Of these three, numerical models are the most commonly used today, with...
the availability of high-speed computers that can solve many systems of equations in a short amount of time. Physical models and unique applications as visual aids allow the actual problem to be scaled down to a size that is manageable and controllable. Analytical models involve solving equations where a definite closed answer is reached at the end of calculations, offering ease of calculation and a simplified version of the real problem. Each of these three types of models will be discussed further below.

2.3.1 Physical Models

Physical models consist of miniature physical analogs of the geology and/or hydraulogy of the situation being studied. Custodio (1987) describes physical models as an analog which is scaled down from the prototype, and in which every pro-to-type element is reproduced, differing only in size. Physical models often come into use in situations where numerical and analog models are inappropriate, due to insufficient historic and hydrogeologic data. Physical models have the advantage of providing a means of visually understanding the problem being studied. One simple type of physical model is a sand-box type model. A container is filled with a porous media, such as sand or glass beads, and the movement of fluids through the media is observed. Another type is the ion-motion analog. In this type of model, the movement of ions, under an electrical gradient, through an electrolytic solution is used to model the movement of fluids through porous media. By introducing other electrically charged probes into the system, hydraulic phenomena such as impermeable layers and pumping wells can be simulated in the system. Another commonly used physical model is the Hele-Shaw Analog (Bear and Dagan, 1964; Segol, 1994). This model is used to represent two-dimensional flow in groundwater systems and consists of two clear plates placed close together with a porous medium between. A flow of fluid between the two plates under a different hydraulic gradient is observed and studied.
2.3 Classification of Groundwater Flow and Transport Models

2.3.2y Analytical Models

The first analytical models that accurately represented hydrogeologic conditions appeared in the 1960's. Analytical models are similar to numerical models, except that the equations involved can be solved exactly at any point of the space flow domain, without the use of approximation methods. In order to arrive at equations that provide an exact solution, many simplifying assumptions must generally be made. Therefore, these models are not suited for systems that involve complex flows and geometries. For this reason, analytical models have a limited use in groundwater flow and transport modeling. However, when analytical models are suitable, they provide solutions that are relatively simple to calculate and understand.

2.3.3y Numerical Models

Numerical models consist of mathematical algorithms that represent the hydraulic and/or chemical aspects of the situation being studied. Systems of partial differential equations that relate parameters such as head, concentration and water flow are commonly utilized by this type of model. A study involving a numerical modeling of groundwater flow started in the early 1960's, most of which these formulations solve the two-dimensional, depth integrated equations using comprehensive approximations, such as Dupuit-Forchheimer for saturated flows, and Ghyben-Herzberg for coastal aquifers. A finite difference is the method of choice during this era; a numerical solution of equations with a range of thousand unknowns was difficult to obtain. During the 1970's, advances were made towards for applications of the finite element method to groundwater models, while in the 1980's many algorithms have been detailed and made much more powerful and efficient. The 1990's era is marked by the development and evolution of three-dimensional models to simulate realistic flow and transport phenomena as they would occur in reality, thus most research efforts are focussing on improving the
existing algorithms, or their adaptation for greater efficiency and robustness.

2.4. TheyComputeModelingParadigm

For every modeling work, many steps are involved to build up a complete re-potent model of the situation being studied. Usually, many stages of the model setup follow the logic presented in Fig. A2. At these steps are grouped into three categories, pre-processing, model run or processing involving application of the numerical method, and the model embedded numerical solver, and finally post-processing tasks related, which are useful for results verification and visualization. A strong link exists between different tasks during the model calibration and verification stage, which is illustrated by the dashed lines in Fig. A2. At these generic model construction steps are discussed in detail below.

2.4.1 ConceptualModelDevelopment

The development of a model concept stands the origin of any modeling effort, and plays a key role in the success of the following steps. A carefully designed and understood conceptual model can save months of an analyst's work. A thorough analysis of the global hydrogeological situation, the flow direction and its seasonal variation, the system communication with other water resources, etc., are acquired at this level. All the information is presented in a form of simplified maps and cross-sections of the aquifer, though a better management (in space and time) is performed by storing and organizing these data in a computerized database, linked with a Geographic Information System (GIS) or at the best a custom GIS or a Hydrogeological Information System (HIS), managing all the data in easily graphic understood form.

It is always a common practice to revise the conceptual model design during the calibration process. As an example, model results may indicate that a negh-
boring hydrogeological unit having lateral contact with the initial model, should be a part of the final model.

2.4.2 Modeling Selection

The selection of a computer code is certainly one of the most challenging tasks. Within the objective to simulate the concept developed in the previous phase, a good knowledge and sufficient level of expertise are needed to choose the most appropriate model for a given situation. However, it should be mentioned that besides the technical requirements, other socio-economic decisive factors...
take much importance. Special care should be taken for understanding the code functionality, and the limitations and approximations used in the numerical approach. An example as the case of modeling saltwater encroachment for aquifer systems, where decision ambiguity exists.

**Miscible Density Dependent Sharp Interface**

Two conceptually different approaches governing the saltwater intrusion, namely the sharp interface and the density dependent flow coupled with miscible salt transport, are widely accepted by the groundwater community (Segol et al., 1975; Galeati et al., 1992; Xue et al., 1995; Yeh et al., 1997). However, in many situations decisive factors determining the choice of the best approach and eventually the numerical code as well, should be defined precisely. For instance, in a situation where regional analysis are required in an advective-dominated flow field, the use of a fully coupled miscible and density dependent transport model will just increase the simulation cost, and make the problem more complex, whereas this level of complexity is not desired.

Nowadays, several computer codes are either freely available, or are packaged and distributed by several vendors with other utilities and auxiliary programs, especially the post-processors. Most of these programs are different distributions of the USGS MODFLOW code (McDonald and Harbaugh, 1988; Harbaugh and McDonald, 1996). An example of a finite element code put in the public domain as AFEMWATER (Yeh and Ward, 1981; Yeh et al., 1997), capable of solving coupled 3-D groundwater flow, and transport. A detailed review of this computer codes is beyond the scope of this work.

### 2.4.3y Meshy Generation

To perform simulations of the partial differential equations governing groundwater flow and solute transport systems on a computer, these continuum equations...
need to be discretized, resulting in a finite number of points in space (and time) at which variables such as groundwater head, velocity, density, and solute concentration are calculated. The usual methods of discretization, finite differences, finite volumes and finite elements, use neighboring points to calculate derivatives, and so there is the concept of a mesh or grid on which the computation is performed. There are two mesh types, characterized by the connectivity of the points. *Structured meshes* have a regular connectivity, which means that each point has the same number of neighbors (for some grids a small number of points will have a different number of neighbors). *Unstructured meshes* have irregular connectivity: each point can have a different number of neighbors. Fig. 2.3 gives an example of each type of grid. In some cases part of the grid is structured and part unstructured.

![Structured Mesh](image1.png) ![Unstructured mesh](image2.png)

Figure 2.3: Structured mesh (left) and unstructured mesh (right) for the same physical domain.

For all types of meshes, there are certain characteristics that we want to control:

- The local density of points: high density gives more accuracy, but computation takes longer. This leads to adaptive meshing methods;
The smoothness of the point distribution: Large variations in grid density or shape can cause numerical diffusion or dispersion. This can lead to inaccurate results or instability.

- The shape of the grid volumes: when the finite element method uses quadrilateral elements the maximum angle must be bounded strictly below to enable convergence of the method as the element size is reduced.

For simple domains, the choice between regular or irregular meshes is governed mainly by the discretization method. However, for complex domains, irregular mesh generation at least for triangular or tetrahedral elements can be fully automatic and fast. Regular mesh generation requires the domain to be split up into simple blocks which are then meshed automatically.

### 2.4.4y Boundary Conditions

Boundary conditions express the link between the conceptual model and its surrounding environment, including external stresses and water flow at domain boundaries. It is important to address specifically each condition and mate the needed related parameters. These are given in many forms, e.g., water head hydrographs, varying pumping rates, etc. The modeler has to account for the fact that with uncertainties which are related to poor initial estimates of the parameters, in such case the start up values should be improved in the model calibration process. A difficult and nonlinear ABC, such as drainage, seepage, or evapotranspiration are very particular and require special concern, their implementation is based on various techniques different from another situation to another.

In many finite element models, the conditions are attributed directly to nodes, by means of an integer index or code, and optionally another index indicating the node behavior when it is necessary to distinguish between different situations (e.g., a drainage node is switching from saturated to unsaturated, or the other way).
2.4.5 Modeling Parameters

These are classified into two main categories: time and physical model parameters. Time parameters (for all transient models) are required and are reduced to a bare minimum if only values corresponding to the simulation time, and an array of output time levels are given. However, other input parameters are required. Usually, because each model uses its own time stepping scheme. An example is given in chapter 4A for GEO-PROF (De Smedt, 1995) and GEO-SWIMAS bai and De Smedt, 1997a) models. Another example is the MODFLOW’s time control, which subdivides the maximum simulation time into intervals defined as “stress periods” over which RC A are considered to be constant, afterwards computational steps are internally estimated, and are adjusted at each nonlinear iterate to fit output at target time values.

Other physical parameters for groundwater models are hydraulic conductivities (or transmissivities) of all soil types, porosity, and specific yield. It is important to specify other input parameters within the framework of other computational solvers, like soil curves. Parameters are characteristics of the unsaturated flow capacity, etc...). Solute transport problems need additionally the longitudinal, horizontal, and vertical transverse dispersivities, the diffusion, decay, and sorption coefficients. When the corresponding mechanisms are taken into account, other physical parameters depend on the water phase properties, e.g., density, and viscosity.

2.4.6 Modeling Run

Before any model run, it is worth the time to check the validity of the input, to probe possible errors and incompatibilities. Indeed, for large applications it can save hours of expensive and useless computations. In this context, visual mapping of the distributed input would be an invaluable assisting tool. It is also a good practice to perform a preliminary steady state simulation from bulk data sets, without introducing changing conditions, e.g., recharge and lateral inflows. This

can give an insightful understanding and guidelines for the future work. Afterwards, an introduction of existing conditions in the developed model can be performed smoothly.

### 2.4.7y Calibration and Sensitivity Analysis

Sensitivity analysis of input data sets constitute another step towards completion of the model calibration. However, these two tasks are often combined simultaneously as one. This is the most tedious and time-consuming phase, and could be a real hassle for the modeler to perform a good calibrated model. Therefore, it is always suggested to set up guidelines to better characterize uncertainties, by determining more sensitive model parameters. A model integrated or independent tools able to help in this task are suggested, however, this can be at the best semi-automatic and recent advances show that user-interaction and skills are always needed for a clean interpretation of the results.

### 2.4.8y Interactive Scientific Visualization

In general, scientists and engineers need efficient, reliable and powerful tools to express their ideas, and illustrate their research findings. Scientific visualization techniques play an ever increasing role in today’s simulation projects. Many software packages have reached a high level of usability and versatility, such that their use become easier even to non professionals. This is especially true for interactive 3-D modeling, which required expensive budgets and highly qualified people a few years ago. One should keep in mind that advanced tools enable not only quality presentation of the simulated results. Advantageously, they accelerate the model calibration phase, and are also useful for checking the input field data.
2.4.9y Verification and Prediction

This is the pre-analysis stage of a modeling study, and these two processes are often linked together. Because verification and future predictions require transient conditions, and hence an extended set of model parameters, which is often not included in the calibrated (steady state) model, there will be other uncertainties in the predictive model, caused by the parameters not being calibrated (and hence estimated).

However, a clear distinction must be made between verification of a given model, and predictive scenarios. Verification assumes a previously calibrated model with an extended or different data sets, while predictive models assume an hypothetical situation, which is projected or possibly available in the future. Popular examples are, projection of abstraction well fields, dispersion of a dissolved chemical matter due to waste migration from a disposal site nearby a pumping station. Hence, predictive models are never thoroughly verified.

2.5y Structured Meshes Generation

This section begins with a discussion of boundary-fitted grids and the discretization of PDE's, and then introduces the multiblock concept used for more complicated domains. Thompson et al. (1985), Knupp and Steinberg (1993), are detailed expositions of structured mesh generation.

2.5.1 Boundary-Fitted Meshes

Structured meshes are characterized by regular connectivity, i.e., the points of the grid can be indexed (by 2 indices in 2D, 3 indices in 3D) and the neighbors of each point can be calculated rather than looked up (e.g., the neighbors of a point are at \((i + 1, j), (i - 1, j), \text{ etc.}\)). A mesh on a rectangular domain are trivial to generate (though some care needs to be taken in the discretization...
at convex corners). And structured mesh generation techniques concentrate on a meshing of domains with regular boundaries. Generally, the meshes are generated so that they fit the boundaries, with one coordinate surface forming a part of the boundary. This gives accurate solutions near the boundary and enables the use of fast and accurate solvers.

For groundwater flow these grids allow the easy application of groundwater models, which usually require a grid to be aligned with the boundary. An alternative is to use a rectangular grid which is clipped at the boundary, with local grid refinement near sharp features. An alternative (Cartesian grids). This will reduce the truncation order at the boundary and will require the mesh cells to be clipped at the boundary, increasing the complexity of the solver. Cartesian grid generation is very fast, but it does not appear to be applicable to general situations. The most common method of generating boundary-fitting grids is to have a continuous grid that fits all the boundaries. This affects the use of rectangular computational domains to a physical domain with curved boundaries.

It is difficult to fit complex domains with one mapping from a rectangular computational domain without generating excessively skewed grids. To get around this problem the domain is split up into blocks and each block is gridded, with some continuity requirements at the block interfaces. This is a multiblock. The decomposition of the domain into blocks as usually done manually using CAD techniques and is slow. An alternative to continuous boundary-fitted grids with multiple blocks is to use a boundary fitting grid near each boundary, and simple rectangular grid in the interior, and interpolate between them. These are called overset or chimera grids, as discussed by Chesshire and Henshaw (1990).

This type of grid is easier to generate than a multiblock grid since each grid is local and does not need to match the others. The individual grids will generally be of high quality (low distortion). However, the interpolation can be difficult, especially with more than two grids overlapping, and it increases the solver time. A
The overlapping grids cannot be too different in resolution and this can cause problems with the grid required for advection transport problems. Chimera grids are very useful for moving boundaries, e.g., the water table and the salt-freshwater interface, or multiple boundaries. Most of the grid remains fixed but the interpolation changes as the grids move with the boundaries. Chimera grids also have certain advantages, and the recent work by Chesshire and Henshaw (1994) on conservative interpolation methods have increased their usefulness. However, the bulk of structured grid generation is based on multiblock type grids.

2.5.2y Multiblock Boundary-Fitted Grids

In theory, complex geometries can be mapped to a rectangular region, but this will lead to unacceptable distortion of the grid cells. In practice, the physical region is broken up into pieces that each have a simple mapping from a rectangular grid.

![Example of a single multiblock grid component](image)

These blocks are fitted together with some degree of grid continuity at their interfaces as shown in Fig. 2.4, ranging from none to complete, such that the final mesh looks like a single grid. With sloping or spacing discontinuities. So, the grid generation process splits into two parts, the decomposition of the physical domain into blocks and the gridding of each block. The decomposition process has not yet been fully automated, and requires considerable user interaction for choosing block edges to align with object edges, aiming to produce good meshes.
The meshing of the blocks can proceed automatically, using one of the methods devoted to one single block.

2.6y They Finite Element Method

2.6.1 Generaly

Continuous physical systems, such as the airflow around an aircraft, the stress concentration in a dam, the electric field in an integrated circuit, or the concentration of reactants in a chemical reactor, are generally modeled using PDE's. The quality of a numerically approximated solution of any PDE depends mainly on the geometry, discretization, and the numerical approximation scheme. An important problem is the discretized domain. The links between these two aspects are very strong, and in the context of a given problem the domain discretization and the numerical scheme which is applied should be considered together.

Finite difference and finite element based methods are the most commonly used numerical approximation techniques. Basically, in a discrete number of nodal points, finite difference based methods approximate the function derivatives, while finite element methods approximate the function itself. Therefore, finite difference methods produce solutions at a discrete number of points, while finite element methods yield spatial solutions. Furthermore, the finite element method has following advantages over finite difference methods:

- Aquifers anisotropy and heterogeneity are easily considered in the approximate formulation;
- Irregular boundaries are easily incorporated;
- Less nodal points are needed. They have the same level of accuracy in a region of interest, thus core computer storage and computational time are saved;
2.6 The Finite Element Method

- The integral formulation used in this method permits the flux type of a boundary condition to come out naturally (Yeh and Ward, 1980).

This does not diminish the importance of finite difference related methods as one of the most used numerical methods worldwide. For the interested reader, a comparison between the two methods have been discussed in detail by Anderson and Voessner (1992), and Gray (1984). However, in this study the finite element method is preferred. The theoretical background of the finite element method will be briefly presented further below.

2.6.2 Basic Concepts

The finite element method envisions the flow domain as non-overlapping smaller elements called finite elements. The dependent variable is spatially approximated by an interpolating function which is continuous to a specified order. The elements are defined by a discrete number of nodal points. Each element is identified by its number and local nodes numbers, coded by following a given numbering sequence.

The finite elements used in this study are irregular hexahedrons, defined by eight nodes as shown in Fig. 2.5. The numbering sequence is first anti-clockwise in the base plane and then, starting above the first node, anti-clockwise in the top plane.

Consider the continuum problem governed by the PDE

\[ L(h)\textbf{u} = \textbf{f} \]

(2.1)

where \( L \) is a differential operator and \( \textbf{u} \) is the dependent variable.

The mathematical concept leading to the FE formulation can be accomplished via the use of either a variational or a weighted residual approach. For more details see Zienkiewics and Taylor (1989), and Lapidus and Pinder (1982).
so-called Galerkin weighted residual scheme is employed in the models presented in chapters 4, 5, and 6.

2.6.3y The Galerkin Weighted Residual Method

Among the existing methods, the Galerkin scheme is one of the most popular weighted residual methods becoming a standard use. In general, the method can be achieved following a straightforward procedure:

1. Define the trial solution as a finite series approximation

\[ \hat{h} \approx \sum_{j=1}^{n} b_j h_j \]  

(2.2)

where \( \hat{h} \) is the approximate dependent variable estimate, e.g., groundwater potential, \( h_j \) are the unknown nodal values, \( b_j \) are linearly independent basis functions defined over the entire domain and \( n \) is the total number of nodes over the hole domain. Since \( \hat{h} \) is only an approximated solution,
residuals occur when replacing it in equation (2.1), this results in an error

\[ r = \mathbf{A}^T(\hat{h})\mathbf{A} \neq 0 \quad \text{A} \]

(2.3) A

2.4 Formulate the Integral Equations

The purpose is to minimize the residual over the problem domain, which is accomplished using the set of the \( m \) basis functions \( \mathbf{A}_i \), orthogonal to the residual

\[ \int_V b_i r \, dV = \mathbf{0} \quad \text{A} \quad \text{for} \quad A = 1, 2, ..., m \quad \text{A} 
\]

(2.4) A

or equivalently

\[ \int_V b_i L \left( \sum_j b_j h_j \right) \mathbf{A} \, dV = \mathbf{0} \quad \text{A} \quad \text{for} \quad A = 1, 2, ..., m \quad \text{A} 
\]

(2.5) A

2.6.4y Basis and Weighting Functions

The element basis functions commonly employed take the form of polynomials. Their construction is best accomplished using the local coordinates \( \zeta, \eta, \zeta \). The dependence between the local and the global coordinates will be derived in the next paragraph. In local coordinates, the original hexahedral element is mapped into a cube whose corners are located at \( \zeta = \pm 1, \eta = \pm 1, \zeta = \pm 1 \), as shown in Fig. A.6.4y.

The eight basis functions for trilinear hexahedral elements are obtained as a product of three orthogonal Lagrange polynomials in three dimensional isoparametric coordinates

\[ b_i(\xi, \eta, \zeta) = \frac{1}{8A} \left( 1 + \xi \eta_i \right) \left( 1 + \eta \eta_i \right) \left( 1 + \zeta \zeta_i \right) \quad \text{for} \quad i = 1, 2, ..., 8 \quad \text{A} 
\]

(2.6) A

2.6.5y Numerical Integration

Since the derived algebraic system of finite element equations is expressed on an integral basis, one may encounter difficulties to evaluate integrals with respect
to global coordinates, and because base functions are expressed in local coordi-
ates, a transformation is required. This transformation is obtained via the basis functions as

\[
x = A \sum_{j=1}^{8} x_j b_j(\xi, \eta, \zeta) A \\

y = A \sum_{j=1}^{8} y_j b_j(\xi, \eta, \zeta) A \\

z = A \sum_{j=1}^{8} z_j b_j(\xi, \eta, \zeta) A
\]

(2.7)A

(2.8)A

(2.9)A

Using this transformation, first-order derivatives of the basis functions versus global coordinates can be changed to their equivalent with respect to local coordi-

\[
\left( \begin{array}{c}
\frac{\partial b_i}{\partial x_S} \\
\frac{\partial b_i}{\partial y_S} \\
\frac{\partial b_i}{\partial z_S}
\end{array} \right) = [J]^{-14} \left( \begin{array}{c}
\frac{\partial b_i}{\partial \xi} \\
\frac{\partial b_i}{\partial \eta} \\
\frac{\partial b_i}{\partial \zeta}
\end{array} \right)
\]

(2.10)A
\[ [J] = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{pmatrix} \] (2.11A)

where \([J]\) is the Jacobian of the transformation.

The encountered integrals arising out from the finite element approximation of the groundwater flow equations may take the forms

\[ \Lambda = \int_V F(x, y, z) \mathbf{A} dx \, dy \, dz \] (2.12A)

where \(F(x, y, z)\) is a continuous function over the volume \(V\). Equation (2.12) can be transformed to

\[ \Lambda = \int_{-14}^{+14} \int_{-14}^{+14} \int_{-14}^{+14} F(\xi, \eta, \zeta) \mathbf{A} \, d\xi \, d\eta \, d\zeta \] (2.13A)

where \(\mathbf{A} = \text{det} \, [J]\). \(\Lambda\) can be easily computed by a Gaussian quadrature.

## 2.7y Advanced Finite Element Methods

### 2.7.1 Mixed Finite Element Method

The mixed finite element method (MFEM) was first introduced to the groundwater community by the works of Meissner (1972), and of Douglas et al. (1983). Since then, the method has been extensively studied and compared to more classical techniques (Brezzi and Fortin, 1991; Chavent and Roberts, 1991; Cordes and Putti, 1997; Durlofsky, 1993). However, the state of the art has been attained for 2-D de-a launey unstructured triangulations (Durlofsky, 1993), and fundamental research is still needed to extend it to a full range of 3-D problems.

The MFEM formulation leads to continuous flux approximations (first-order accurate) across the element edges, whereas the conforming finite element method fails (discontinuous fluxes). This is achieved by adding additional degrees of freedom...
freedom at the elements amid-edges. The flow equation is discretized using a known groundwater heads at the nodes, and unknown Darcian velocity vectors at the amid-edges. A typical scheme employed for this type of problem as the Raviart-Thomas (RT0) approach that uses piecewise constant basis functions for the Darcian velocity.

As a consequence, the solution is more accurate for the flow field. Also, there is an expensive tooth for such accuracy; the MFE formulation generates about twice as many degrees of freedom as standard finite elements of the same order.

By generating high accuracy velocity fields, the MFEs as an attractive method for modeling coupled flow and transport problems. In particular, the method has been effectively used in conjunction with the finite volume method for the transport equation (Dulofsky, 1993).

2.7.2y Control Volume Finite Element Method

The essence of the control volume finite element method (CVFE) is to use the finite element basis functions to approximate the groundwater heads at the nodes, whereas the conservation equations are applied to control-volumes (Forsyth, 1989). The control-volumes definition is highly flexible, they may be patches centered around the nodes as illustrated in Fig 2.7, or coincide completely with the finite element cells. For more general triangulations, control volumes are general polygons spanning more or fewer elements and which are not necessarily convex.

Here again, recent works have focused on 2-D triangular elements (even if 3-D extensions are straightforward, at least for tetrahedrals) though there exists some restrictions with respect to the triangulation used (Forsyth, 1989; Jie and Van Quy, 1992; Gottardi and Venutelli, 1993).

The cell-volume CVFE dual discretization approach can be used in order to improve the approximation of fluid velocities, computed at the basis of groundwater-
2.8 Iterative Methods

2.8.1 Iterative Methods Versus Direct Methods

The discretization of PDE’s using numerical approximation methods leads to a system of algebraic equations which is in most cases expressed in matrix form. The problem reduces to solving the given system of equations for unknowns \( n \) is the matrix order.

Regardless of the structure of the system, e.g., linear or nonlinear, dense or sparse, one has to choose between direct or iterative solution methods.

The present state of the art in numerical methods is that direct methods can be used as black boxes. This is by far not the case for iterative methods, at least not if we do not know the specific properties of the matrix of the linear system.
to be solved. And even when a trivial matter is decided when a top the iteration process and to obtain a reasonable estimate of the approximation error in the result.

The simplest direct method is probably the well-known Gaussian elimination procedure, unfortunately this popular method leads to fill-in, which makes the method often expensive. Usually large sparse matrices are related to some grid or network, and it is highly desirable to exploit this ‘nice’ property efficiently.

For large 3-D problems, iterative methods are preferable, for this class of problems Van der Vorst and Chan (1998) estimated the flops count for a direct solution method in the order of $A^3$, and the number of flops for an iterative solver in the order of $A^{3/2}$. Also the requirements for memory space for the iterative methods are typically smaller by orders of magnitude. This is often the argument for the usage of iterative methods in 2D situations, when flop counts for both classes of methods are more or less comparable. Finally, it should be noted that iterative methods can exploit good initial guesses, e.g., in time dependent problems. A preconditioner can often be chosen to adapt to the machine architecture.

The above given arguments are quite nicely illustrated by observations made by Simon (1989). For linear problems with some $A \times 10^{34}$ unknowns, he has estimated the CPU time required by the most economic direct method, as 520,040 years, provided that the computation can be carried out at a speed of 1 TFLOP. On the other hand, he estimates the CPU time for preconditioned conjugate gradients, assuming still a processing speed of 1 TFLOPS, as 575 seconds. Though we should not take this granted that in particular a very sparse preconditioning part can be carried out at that high processing speed (for the direct solver this is more likely), and the CPU values may change, we see that the differences in CPU time requirements are gigantic.
2.8 Iterative Methods

2.8.2y Basic Iterative Method

very basic idea, that leads to many effective iterative solvers, is to split the matrix of a given linear system as the sum of two matrices, one of which leading to a system that can easily be solved. The most simple splitting we can think of is $A = A_y - (I_y - G)$. Given the linear system $G h y = A B$, this splitting leads to the well-known Richardson iteration $A$

$$h^{i+1} = A y (I y - G) h^i = A h^i + r^i$$

Multiplication by $A - G y$ and adding $A y$ gives $A$

$$B y - G h^{i+1} = A y - G h^i - G r^i$$

or $A$

$$r^{i+1} = (I y - G) r^i = (I y - G) r^i + r^0 = \mathbb{P}_{i+1}(G) r^0$$

or, in terms of the error $A$

$$G(h y - h^{i+1}) = \mathbb{P}_{i+1}(G)(h y - h^0) A$$

$$h y - h^{i+1} = \mathbb{A} \mathbb{P}_{i+1}(G)(h y - h^0) A$$

In these expressions $\mathbb{P}_{i+1}$ is a (special) polynomial of degree $A + 1$. Note that $A$

$$\mathbb{P}_{i+1}(0) A = A$$

Results obtained for the standard splitting can be easily generalized to other splittings, since the more general splitting $A y = A y - N y = A y - (M y - G)$ can be rewritten as the standard splitting $A y = A y - (I y - B)$ for the preconditioned matrix $A y = M^{-1} G A$. Other more powerful iteration methods can be viewed as accelerated versions of the basic iteration methods. In the context of these accelerated methods, the matrix splittings become important in another way, since the matrix $M y$ of the splitting is often used as a preconditioning matrix. That is, the iterative method is applied to, e.g., $M^{-1} G y = M^{-1} B A$
For the simple Richardson iteration it follows that

\[ h^{i+1} = A r^{04} + r^{14} + r^{24} + \ldots + r^i = \sum_{j=04}^i (Iy - G)^j A^{04} \]

\[ \mathbb{K}^{i+1}(G; r^0) \] as a subspace of dimension 1, generated by \( A^{04} \) and \( G \) and \( A \) called a Krylov subspace for \( G \) and \( A \). Apparently, the Richardson iteration delivers elements of Krylov subspaces of increasing dimension. Note that the Richardson iteration generates a basis for the Krylov subspace, and this basis can be used to construct other approximations for the solution of \( G y = A y \) as well.

### 2.8.3y Linearly Symmetric and Nonsymmetric Systems

When the system matrix is symmetric and positive definite, which is the case for the conductance matrix derived from the FE discretization of the governing equations for groundwater flow, powerful and efficient preconditioned iterative solvers are available, and becomes widespread. The most attractive feature of many of these solvers is the mathematically guaranteed convergence in an arithmetic number of iterations. This is however, not the case in practice, where several difficulties might be encountered whenever the positive transmissibility (PT) condition is not fulfilled (Putti and Cordes, 1996). We refer to section 4.3 for a more detailed explanation.

For matrices that are not positive definite symmetric the situation can be more problematic. As often difficult, and a proper iterative method or a suitable preconditioner. However, projection type methods, like GMRES, Bi-CG, CGS, and Bi-CGSTAB are used as alternatives, even if extreme care should be taken when choosing a most appropriate solver to a given class of systems. Despite CG-like algorithms for symmetric matrices, the convergence is neither guaranteed for such accelerators. A section 6.3 details these aspects, and discusses several examples of preconditioning showing their strengths and weaknesses.
Chapter 3G

Numerical Formulation of Groundwater Flow Models

Contents

3.1 Introduction .................................................. 37
3.2 Governing Equations for 3-D Groundwater Flow ... 38
3.3 Governing Equations for Saltwater Intrusion ...... 47
3.4 Application of The Finite Element Method ....... 51
3.5 Finite Element Matrix Analysis ....................... 57
3.6 Solution Strategies ......................................... 61

3.1 Introduction

The mathematical formulation of the governing groundwater flow equations and the steps involved in their derivation are highlighted, while almost all of the discussion focuses on the distinction of the different flow situations encountered.
in practice. Of special concern in the next chapters, will be the case of variably saturated flow and saltwater intrusion. Next, conforming Galerkin finite element discretizations of these equations are derived and presented in their general form as used and implemented in the numerical groundwater flow models discussed further. The numerical algorithms used to handle each of the equations are left for the upcoming chapters. So, herein only classical derivations as found in the relevant literature are applied. Interesting issues such as treatment of different boundary conditions from numerical point of view, and different implementation techniques are discussed in more detail. A clear picture of different strategies for solving the finite element equations system is given, in which different methods used during the last decades are compiled, compared and commented.

3.2 Governing Equations for 3-D Groundwater Flow

3.2.1 Basic Equations

The governing equations for variable saturated flow in heterogeneous porous medium are derived based on the mass conservation equation and the generalized Darcy law relating flux to potential gradient.

In general, local mass conservation in compressible and variable saturated porous media is expressed as

$$\frac{\partial}{\partial t} [S_p(p) + \theta(p)] = -\nabla q + R$$  \hspace{1cm} (3.1)

where $p$ is the water pressure $[\text{M/LT}^2]$ relative to atmospheric pressure, negative in the unsaturated zone and positive in the saturated zone, $S_p$ is the elastic storage coefficient due to the combined effect of water and solid porous matrix compressibility, and saturated or unsaturated conditions $[\text{dimensionless}]$, $\theta$ is the volumetric water content $[\text{dimensionless}]$, $q$ is the specific discharge rate
3.2 Governing Equations for 3-D Groundwater Flow

$R$ is the internal source/sink term [$T^{-1}$], and $\nabla$ is the del operator [$L^{-1}$], $(\partial/\partial x, \partial/\partial y, \partial/\partial z)^T$, where $\mathbf{x} = (x, y, z)^T$ is the location vector [$L$].

The generalized Darcy law for variable saturated flow in heterogeneous porous media is given by

$$\mathbf{q} = -\frac{k(p)}{\mu} \left[ \nabla p + \rho g \nabla z \right]$$ (3.2)

where $k$ is the unsaturated permeability tensor [$L^2$], $\mu$ is the dynamic viscosity [$M/LT$], $g$ is the acceleration due to gravity [$L^2T^{-1}$], $\rho$ is the fluid density [$M/L^3$], and the $z$-axis is taken vertical and positive upwards.

Introducing the hydraulic head or groundwater potential $h$, as

$$h = z + \frac{p}{\rho g} = z + \psi$$ (3.3)

Where $\psi$ is the pressure head [$L$]. Darcy’s law can be written in an equivalent form

$$\mathbf{q} = -K(h) \nabla h$$ (3.4)

where $K = k\rho g/\mu$ is the hydraulic conductivity tensor [$LT^{-1}$].

3.2.2 Saturated Groundwater Flow

Under saturated conditions, the porous medium compressibility depends upon the water pressure, or groundwater potential, such that we can assume the elastic storage coefficient, $S_p$, proportional to the pressure head, $p$

$$S_p = S_s p \quad \text{if} \quad p \geq 0$$ (3.5)

where the coefficient of proportionality, $S_s$, is called the specific storage coefficient [$L^{-1}$], depending only upon compressibility characteristics of the porous medium and the fluid, $S_s$ represents the volume of water produced per unit saturated volume of the porous medium, per unit decline of groundwater potential.
Note also that the saturated water content, \( \theta_s \), and the corresponding con-Aductivity \( \mathbf{K}_y \) do not depend on pressure. Inserting equation (3.4) and (3.5) into equation (3.1) yields a linear partial differential equation governing saturated flow in a 3-D heterogeneous porous medium:

\[
S_s \frac{\partial h}{\partial t} = \mathbf{\nabla} (\mathbf{K}_y \mathbf{\nabla} h) + R
\]  

(3.6)

### 3.2.3y UnsaturatedyGroundwateryFlowy

In this case, it is assumed that the storage due to compression of the medium or the fluid can be ignored in comparison to storage resulting from changes in the water content (Freeze and Cherry, 1979), i.e.

\[
S_p = 0 \quad \text{if } A_p < 0
\]  

(3.7)

Inserting equation (3.4) and (3.7) into equation (3.1) yields a nonlinear partial differential equation governing unsaturated flow, namely the \( \mathbf{Richard} \) equation:

\[
\frac{\partial \theta}{\partial t} = \mathbf{\nabla} (\mathbf{K}_y \mathbf{\nabla} h) + R
\]  

(3.8)

Notice that this equation contains the groundwater potential \( h \), and the water content \( \theta \), which are a-priori the unknowns of the problem, therefore additional relationships relating these variables and the hydraulic conductivity should be formulated. Examples of these constitutive relationships are given in Section 3.2.6.

Using the chain rule \( \frac{\partial \theta}{\partial t} = \frac{\partial \theta}{\partial h} \frac{\partial h}{\partial t} \) we obtain the following equation:

\[
C \frac{\partial h}{\partial t} = \mathbf{\nabla} (\mathbf{K}_y \mathbf{\nabla} h) + R
\]  

(3.9)

where \( C(h) \) is the water capacity \([L^{-1}]\), given as:

\[
C = \frac{d\theta}{dh} = \theta_s \frac{dS_w}{dh}
\]  

(3.10)
3.2 Governing Equations for 3-D Groundwater Flow

$C$ represents the amount of water released per unit volume of unsaturated porous medium, per unit decline of the groundwater potential, and $A S_w (=\theta/\theta_s)$ is the saturation of water ($0 \leq S_w \leq 1$) [dimensionless].

In the general case, where the elastic storage coefficient is not neglected, the exact derivation leads to a general storage term in the left-hand side of equation (3.9) which is given by (see Huyakorn and Pinder, 1983)

$$S(h) = A S_w S_s + C$$

(3.11)

Hence, the general equation governing the flow in 3-D variably saturated porous media may be written as

$$S(h) \frac{\partial h}{\partial t} = \nabla \cdot (K \nabla \psi) + R$$

(3.12)

where $K_r$ is the relative hydraulic conductivity tensor [dimensionless]. This form of the governing equation is the so-called $h$-based $\psi$-Richard's equation, while other forms are the $\psi$-based $\psi$-pressure head-based $\psi$-Richard's equations, where the de-pendent variables are substituted for $h$ and $\psi$ respectively. However, the $h$-based form presents some advantages, as it can be used for saturated and unsaturated soils, as well for layered soils.

### 3.2.4y Steady State Groundwater Flow

Another possibility arising from the use of Equations (3.6), (3.9) and (3.12) when the variables become independent of time, the flow equation can be reduced to

$$\nabla \cdot (K \nabla \psi) + R = \theta$$

(3.13)

The nonlinearity in the groundwater flow equation is still not removed, but de-creased by an order of magnitude, since dependencies on the groundwater heads are shortened to $K_r = K_r(h)$ relationship. Water contents are not involved in the steady state groundwater potentials, meaning implicitly that steady saturation profiles have a sharp distribution ($\theta = \theta$ in the saturated zone, and $\theta = \theta$ in the unsaturated zone).
3.2.5y Unconfined Groundwater Flow

In modeling unconfined aquifer systems, care should be taken of the vadose zone above the saturated region. The two regions can be distinguished from physical point of view by the variability of the water content denoted here by \( \theta \) (\( 0 < \theta < n \)), which equals the porosity, \( n \), or saturated water content, \( \theta_s \), in the saturated zone. In between an abrupt water table delimits sharply the vadose and saturated zones. Neuman and Witherspoon (1970) derived the exact mathematical expression for the water table as a boundary condition, given by

\[
(R - \theta_s \frac{\partial \zeta}{\partial t}) \mathbf{n}_i = \sum_{i=1}^{3} (K \nabla h) \cdot \mathbf{n}_i
\]

where \( \mathbf{n}_i \) are components of the outward unit vector, and \( \nabla h \) represents the water table elevation above the datum level. This is a direct consequence of the fact that the water table is a streamline satisfying the kinematic condition \( \frac{dF}{dt} = 0 \), where \( F(x, t) = \xi(x, y, t) \) represents the geometry of the water table. In addition, it is necessary to prescribe the atmospheric pressure, taken as a reference, to the water table boundary, such that:

\[
h(x, t) = \Lambda
\]

3.2.6y Constitutive Relationships

In general, for variably saturated flow, relative hydraulic conductivity and volumetric moisture content depend on the pressure head; these well-known relationships are shown in Figure 3.1, which should be identified in order to solve the mathematical problem. The tremendous variability of soils compounds, their complex behavior under saturation and desaturation processes, and the hysteretic nature of these changes (see Figure 3.2), makes an exact quantitative description of the soil curves very difficult. Practically, these curves are obtained from laboratory experimental analysis, by fitting appropriate parameters of semi-empirical expressions.
3.2 Governing Equations for 3-D Groundwater Flow

Figure 3.1: Representative unsaturated hydraulic conductivity curves for given soil types.

Among the most popular ones, the Van Genuchten and Nielsen (1985) equations are given by

\[
\theta(\psi) = \begin{cases} 
\theta_r + (\theta_s - \theta_r)[1 + \beta]^{-\gamma} & \text{if} \ \psi < 0A \\
\theta_s & \text{if} \ \psi \geq 0A 
\end{cases}
\]  

(3.16)

\[
k_r(\psi) = \begin{cases} 
(1 + \beta)^{-5\gamma/2}[(1 + \beta)^\gamma - \beta^\gamma]^2 & \text{if} \ \psi < 0A \\
1 & \text{if} \ \psi \geq 0A
\end{cases}
\]  

(3.17)

where \( \theta_r = \left( \frac{\psi}{\psi_a} \right)^n \), \( \psi_a (\leq 0) \) is the capillary air-entry pressure head value, \( \psi \) is the pressure head, and \( n \) is a constant, with \( \psi = 1 - \frac{1}{n} \) for \( n \) approximately in the range 1.25 to 6.
Paniconi et al. (1991) have suggested the following substitution

\[
\theta(\psi) = \begin{cases} 
\theta_r + (\theta_s - \theta_r)[1 + \beta]^{-\gamma} & \text{if } \psi < \psi_{04} \\
\theta_r + (\theta_s - \theta_r)[1 + \beta_0]^{-\gamma} + S_s(\psi - \psi_0) & \text{if } \psi \geq \psi_{04}
\end{cases}
\]  

(3.18)7

where \(\psi_{04}\) is a continuity parameter, \(\beta_{04} = M(\psi_0) = \left(\frac{\psi_{04}}{\psi_a}\right)^n\).

Other widely used functions are given by Brooks and Corey (1964):

\[
\theta(\psi) = \begin{cases} 
\theta_r + (\theta_s - \theta_r)A\left(\frac{\psi}{\psi_a}\right)^\lambda & \text{for } \psi \geq \psi_a \\
\theta_s & \text{if } \psi < \psi_a
\end{cases}
\]  

(3.19)7

\[
k_r(\psi) = \begin{cases} 
\left(\frac{\theta - \theta_r}{\theta_s - \theta_r}\right)^m & \text{if } \psi < 0A \\
1 & \text{if } \psi \geq 0A
\end{cases}
\]  

(3.20)7

where \(A\) is a constant, and \(A\) is a conductivity shape parameter.
3.2 Governing Equations for 3-D Groundwater Flow

Fuentes et al. (1992) showed that a combination of the Van Genuchten water retention equation (3.16) and the Brooks and Corey conductivity equation (3.20) yields the most consistent approximation for a large number of soil types encountered in practice. Therefore, we recommend using such a combination for the model developed afterwards. However, a major difficulty in applying such equations on large scale is that 'point' scale measurements are only in situ quantities of the real representative field parameters, which make this approach subject to many uncertainties and inadequate for large scale field studies. Notice that, for each soil type a significant number of related parameters is involved, some dimensionless parameters are usually obtained based on curve fitting tech niques rather than a physical meaning based approach (Haverkamp et al., 1999).

3.2.7y Boundary and Initial Conditions

The specification of appropriate boundary conditions is essential in groundwater modeling. Each set of boundary conditions defines a unique solution of the mathematical problem. These conditions need to be well understood from a physical point of view, and their mathematical formulation and numerical treatment should be worked out in an efficient way.

Conditions at the flow domain boundaries are classified from mathematical point of view either as Dirichlet, Neuman, Cauchy or variable conditions.

Dirichlet Boundary Conditions

Also known as first type boundary condition, because the potential value is prescribed on a given boundary $S_{14}$

$$ h = A_0(x, t) \quad \text{on} \quad S_{14} $$

(3.21)
NeumanyBoundary Conditions

This is a second type boundary condition, involving a prescribed flux normal to the boundary, or a prescribed gradient of the potential \( A \)

\[
q_n(x, t) = A(K_T V h) \mathbf{n} = -A q_0(x, t) \quad \text{on} \quad S_2 \quad (3.22)
\]

where \( \mathbf{n} \) is the outward unit vector normal to boundary \( S_2 \), \( q_0(x, t) \) is considered positive when entering the domain and negative otherwise. Neumann boundary conditions are typically encountered at the boundaries of aquifer systems where either recharge or drainage occurs, or no flow \( q_n = 0 \) at impervious boundaries. Other examples are water divide lines, pumping or recharging wells, infiltration, effective rainfall, water outflow to the sea, and ground water inflow or outflow through a boundary from a part of the aquifer that is not considered in the simulation, etc.

Cauchy Boundary Conditions

The third type of boundary condition, which is also known as head dependent flux boundary condition, involves prescribing the total normal flux due to the gradient in the boundary in response to changes in head within the aquifer adjacent to this boundary.

\[
q_n(x, t) = A \lambda [h_0(x, t) - h] = A q_0(x, t) \quad \text{on} \quad S_3 \quad (3.23)
\]

where \( \lambda \) is a constant. This type of boundary can be illustrated by the upper surface of an aquifer overlain by a semi-confining bed that is in turn overlain by a body of surface water. The flux, \( q_{04} \) across the semi-confining bed entering the aquifer, is given by Darcy's law as

\[
q_{04} = K' \frac{h_{04} - h}{d} = C_{04}(h_{04} - h) A \quad (3.24)
\]

where \( K' \) is the hydraulic conductivity of the semi-confining bed, \( d \) is its thickness, \( C_{04} \) is the specific conductance of the resisting layer, \( h_{04} \) is the head in the surface water body, and \( A \) is the head in the aquifer.
Examples of such a boundary condition are artificial injection of water or pumping of ground water, where the transfer as subjected to a certain resistance, as a sedimented infiltration pond or a clogged well, or a stream with a muddy bed.

**Initial Conditions**

For variable groundwater flow problems initial conditions have to be specified, given by

\[ h = h_0(x) \quad \text{in} \quad V \quad \text{(3.25)} \]

where \( h_0 \) is the prescribed initial value for the groundwater potential, and \( V \) is the region of interest.

### 3.3 Governing Equations for Saltwater Intrusion

#### 3.3.1 Basics

The flow in both saltwater and freshwater zones is modeled via an abrupt interface assumption (Bear and Verruijt, 1987). As shown in Fig. A3.3, this approach is successfully applied in case the transition zone is thin relative to the thickness of the freshwater lens. The exact position of this interface is initially unknown; in fact this is part of the solution, such that the sharp interface constitutes a free nonlinear boundary of the problem.

With the abrupt interface approximation, only the flow equations of both fluids need to be solved, i.e., salt concentrations in the freshwater zone \( V_F \), and in the saltwater zone \( V_S \) are respectively given by

\[
\begin{align*}
C &= 0 \quad \text{in} \quad V_F \\
C &= \mathcal{K}_s \quad \text{in} \quad V_S \\
\end{align*}
\quad \text{(3.26a)}
\]
Notice that consequently the density of each flow phase becomes constant,

$$\begin{cases} 
\rho = \rho_f & \text{inA } V_F \\
\rho = \rho_s & \text{inA } V_S
\end{cases} \quad (3.27a)$$

This is easily deduced from the empirical formula suggesting density to be linearly dependent upon concentration, and given by

$$\rho = \rho_f (1 + \delta c) \quad (3.28)$$

where, $\rho_f$ is taken as a reference density, $c \equiv (C/C_s)$ is a normalized concentration, and $\delta = (\rho_s - \rho_f)/\rho_f$ is the density difference ratio, such that $\rho_s$ is the density at the maximum concentration ($c = 1$).

**3.3.2y The Multiphase Sharp Interface Approach**

Under these assumptions, the flow equations governing the flow in each phase are stated independently as in Bear (1979), Volker (1980) and Huyakorn and Pinder...
(1983)A

\[ S(h^{(f)}) \frac{\partial h^{(f)}}{\partial t} = A \nabla \left( K^{(f)} \nabla h^{(f)} \right) + R^{(f)} \]  
(3.29a)A

\[ S(h^{(s)}) \frac{\partial h^{(s)}}{\partial t} = A \nabla \left( K^{(s)} \nabla h^{(s)} \right) + R^{(s)} \]  
(3.29b)A

where \((f)\) and \((s)\) superscripts on the equation variables denote fresh and saltA water phases respectively. The general storage coefficient for the saltwater phaseA is rigorously acquired. An analysis of special conditions. AFor example, the saltwater phase may overlap with the unsaturated zone, a salt cone of depression may develop in contact with the unsaturated zone, or infiltration of salty water from irrigation parcels need to be simulated.

The coupled equations (3.29a) and (3.29b) are solved for a given problem if unique boundary conditions are specified. These include in particular the fresh-saltwater interface as a physical boundary condition for the two equations simultaneously. AVolker (1980) used a similarity between the water table and the fresh-saltwater interface, leading to a generalized form of equation (3.14), and given by

\[ -\theta_s \frac{\partial \eta}{\partial t} n_{34} = \sum_{i=14}^3 \left( K^{(f)} A \nabla h^{(f)} \right) . n_i \]  
(3.30a)A

\[ -\theta_s \frac{\partial \eta}{\partial t} n_{34} = \sum_{i=14}^3 \left( K^{(s)} A \nabla h^{(s)} \right) . n_i \]  
(3.30b)A

where \(\eta\) is the interface depth below the datum level, given by Hubbert (1940) as

\[ \eta = \frac{A \rho_s}{\rho_s - \rho_f} h^{(s)} - \frac{\rho_f}{\rho_s - \rho_f} h^{(f)} \]  
(3.31)A

This leads to the following equations relating the change of the position of the interface with time to potential gradient (or Darcian velocity) components

\[ \frac{\partial \eta}{\partial t} = \frac{K^{(f)} A \partial h^{(f)} \partial \eta}{\theta_s} \frac{\partial h^{(f)}}{\partial x} + \frac{K^{(f)} A \partial h^{(f)} \partial \eta}{\theta_s} \frac{\partial h^{(f)}}{\partial y} + \frac{K^{(f)} A \partial h^{(f)} \partial \eta}{\theta_s} \frac{\partial h^{(f)}}{\partial z} \]  
(3.32a)A

\[ \frac{\partial \eta}{\partial t} = \frac{K^{(s)} A \partial h^{(s)} \partial \eta}{\theta_s} \frac{\partial h^{(s)}}{\partial x} + \frac{K^{(s)} A \partial h^{(s)} \partial \eta}{\theta_s} \frac{\partial h^{(s)}}{\partial y} + \frac{K^{(s)} A \partial h^{(s)} \partial \eta}{\theta_s} \frac{\partial h^{(s)}}{\partial z} \]  
(3.32b)A
The equation for freshwater potential (3.32a) is derived by several authors, among which Bear (1972), Sugio and Desai (1987) implemented it in a 2-D seawater intrusion finite element model, and Bakker (1998) used it for solving the transient Dupuit interface with the analytic element method.

### 3.3.3y A Simplified Approach

Solving the coupled system of equations (3.29a) and (3.29b) subject to boundary conditions, including (3.32a) and (3.32b), is delicate and expensive. Solving such a problem for a fully three-dimensional flow field will introduce another needed complexity. Instead, a simplified approach is developed to determine the sharp interface position, assuming a quasi-stationary saltwater zone, with a hydrostatic pressure distribution

\[ p = \rho g z \quad (3.33) \]

by using the expression in equation (3.3)\(^A\)

\[ h = \Delta \delta z \quad (3.34) \]

Assuming that the hydraulic head changes in the saltwater zone are small, during the saltwater displacement (Hantush, 1968; Anderson, 1976) it is only necessary to solve Equation (3.29a) for the freshwater heads, and hence the coupling as a removed, which is a very suitable and cheap. Thus, the equation governing the flow for saltwater intrusion in 3-D heterogeneous aquifer systems is the same as equation (3.12), except that an additional unknown free boundary, namely the sharp interface, is to be determined.\(^A\)

Differentiation of equation (3.31) gives

\[ \frac{\partial \eta}{\partial t} = \Delta \frac{\partial h}{\partial t} \quad (3.35) \]

Equation (3.35) shows explicitly that the variations of the interface position are dependent on the density difference and the groundwater heads in the flow field,
while the equation is more suitable for numerical implementation.

3.4y Application of the Finite Element Method

For the sake of simplicity, we shall approximate the steady groundwater flow equation (3.13), since the right-hand side of the transient variably saturated flow equation, which expresses the time dependence, will be approximated later. An analytic technique.

3.4.1 Galerkin's Spatially Approximation

By applying the FEA approach already described in the previous chapter, the groundwater potential is approximated by a finite series as

$$\hat{h} \approx \sum_{j=1}^{nn} b_j(x) \phi_j$$  \hspace{1cm} (3.36a)

where \( \phi_j \) is the nodal value of the groundwater potential, \( b_j(x) \) is a trilinear nodal basis function and \( nn \) is the number of the nodes in the problem domain. Using equation (3.36) in equation (3.13) and applying the orthogonality condition in equation (2.5) results in the following equation

$$\int_V \{ \nabla [K \nabla ( \sum_j b_j \phi_j)] + R \} b_i \, dV = \Phi_i \quad \text{for } i = 1, 2, ..., nn$$  \hspace{1cm} (3.37a)

by applying the Green's theorem on the first term and integrating over all elements and forming a matrix. Over the domain, we obtain the FEA equations in matrix form, which are linear for saturated flow and nonlinear for unsaturated flow

$$[G] \{h\} = \{B\}$$  \hspace{1cm} (3.38a)
where \( \{h\} \) is the unknown vector containing the nodal potentials, \( [G] \) is the global conductance matrix depending on the geometrical and conductive properties of the flow domain and \( A \{B\} \) is a vector containing all boundary conditions, and sources and sinks terms.

In section 3.5 some important properties of the conductance matrix are explored for an efficient use by the solver. The entries of \( [G] \) are given by

\[
G_{ij} = \sum_{e} G_{ij}^{e} = \sum_{e} \int_{V^{e}} \nabla b_{i}^{e} \cdot \mathbf{K} \nabla b_{j}^{e} \, dV^{e}
\]

(3.39)

where the local element contributions, \( G_{ij}^{e} \), are calculated in local coordinates \( (\xi, \eta, \zeta) \) by means of equations (2.7) through (2.13).

The boundary conditions, together with sources and sinks, are incorporated in the entries of vector \( A \{B\} \) given by

\[
B_{i} = A \int_{S_{n}} q_{n} \, dS + A \int_{V} R \, b_{i} \, dV
\]

(3.40)

where \( q_{n} \) is the outer normal flux through the boundary surface, \( S \).

### 3.4.2y Finite Difference Approximation in Time

finite difference method is used for approximating the time derivative. An implicit method or backward difference is adopted in the presented groundwater flow model because it is unconditionally stable and quite resistant to oscillatory nonlinear instability (Huyakorn and Pinder, 1983), even if the method is only first order accurate.

Considering the governing equation (3.12), the matrix system is written in the form

\[
\frac{[S]}{\Delta t} \{h\}^{t+\Delta t} + \{G\} [A \{h\}]^{t+\Delta t} = A \{B\}
\]

(3.41)

where \( [S] \) is the diagonalized storage matrix having the following entries

\[
S_{ij} = \sum_{e} \int_{V^{e}} S^{e} b_{i}^{e} \, dV^{e}
\]

(3.42)
where $\delta_{ij}$ is the Kronecker symbol, and $S^e$ is the storage coefficient of element $A$ as defined in equation (3.11).

Assuming that the element storage quantity $AS^e$ is averaged over each element $A$ volume, and that the value is attributed to the 3-D element centroid, we may express the diagonal matrix storage terms as

$$S_{ii} = \sum_e S^e \int_{V_e} b^e_i dV^e = \sum_e S^e V^e_i$$

(3.43)

where $V^e_i$ is the control volume contribution of element $A$ at node $A$ as shown in Fig. A.4. Hence, the storage term does only increase the diagonal dominance of the global matrix $A$.

![Hexahedral element with control volume](image_url)

Figure 3.4: Three-dimensional control volume contribution of the element $e$ at node $w$

Finally, equation (3.41) can be transformed to a similar form as in equation (3.38), but in this case with a global matrix $[G^+]$ and the boundary vector $[B^+]$.
having respectively the following entries:

\[ G_{ij} = \sum_{e} \int_{V_e} \nabla b_i^e \cdot K \nabla b_j^e \, dV + \frac{S_{ij}}{\Delta t} \]  \hspace{1cm} (3.44) \\

\[ B_{i}^* = \int_{S} g b_i \, dS + \int_{V} R b_i \, dV + \frac{S_{ii}}{\Delta t} h_i^t \]  \hspace{1cm} (3.45) \\

So, in the remainder sections, \( G \) and \( B \) will refer to entries given by equations (3.44) and (3.45) respectively.

### 3.4.3y Numerically Implementing Boundary Conditions

Basically, for a general groundwater flow problem there are different boundary conditions possible, depending upon the type of flow: confined, unconfined, or unsaturated-saturated flow. In the presented FE numerical model, a wide range of boundary conditions were implemented to allow for either natural or artificial stresses that might be encountered in practice. All conditions are attributed to the nodes of the finite element mesh, by means of a boundary condition code.

#### Fixed Potential

In such nodes the potentials are given a fixed value in the left-hand side of equation (3.41). And are no longer calculated by the model. As a consequence, for such nodes located on the boundary, the finite element equations are not needed. A simple substitution of equation (3.21) in the matrix system leads to an unsymmetrical global matrix, which as an undesirable property for solving the matrix system. Therefore, in the remaining equations all known fixed potential terms are moved to the right-hand side of equation (3.41) in order to conserve symmetry, and the equations corresponding to a prescribed potential nodes are simply skipped in the computational routines (Larabi and De Smedt, 1994), this method is efficient because it diminishes the dimension of the number of unknowns of the matrix system. It is also robust, because prescribed potential values are preserved at their initial fixed values.
Fixed FluxoryFlowy

Here, the flux or flow rate of inflow or outflow of water in a node or at a series of nodes is fixed in the right-hand side of the matrix system. Hence, boundary conditions prescribed at Neuman boundaries are explicit, except for nodes with a prescribed flux value, and for which the nodal surface normal to the flux vector must be calculated. This is practically feasible for vertical fluxes, as recharge or seepage, otherwise this becomes tricky or difficult to implement. Furthermore, the nodal horizontal surface areas are needed in other computational finite element routines, such that the cost of this implementation is reduced.

Leakagey Fluxory Flowy

In nodes of mixed type boundary conditions as described previously in section 3.2.7, the resisting layer characteristics are described by one parameter, which can be considered as a global conductance, \( AC_{04} \) or as a nodal specific conductance, \( C_{04} \) depending on which condition (flux or flow) is prescribed, respectively. If the conductance is large, \( h \) will be nearly equal to \( h_{04} \) but in the opposite case no much flow is possible and the potential will be different from \( u_0 \). Nodes at boundary \( S_{34} \) are explicitly included in the matrix system, and diagonal dominance of the general matrix will increase at rows corresponding to this condition, i.e. global conductance terms are added to some of the matrix diagonal entries.

Seepage Facey

This condition applies to the case of a seepage face. An outflow can occur under atmospheric conditions, this means zero pressure, or \( Q \) equal to \( A \), but no inflow is possible. This is expressed as

\[
\begin{align*}
  h &= A & \text{if } Q < 0 \\
  Q &= A & \text{otherwise}
\end{align*}
\]

(3.46)
The position of this boundary is known, but its extent is initially unknown. To handle the complexity of such situation, an iteratively based procedure is implemented in the computer model which determines nodal points of this kind, this is an improved version of the Neuman’s procedure (1973). All the nodes which can possibly be on the seepage face are treated initially as prescribed potential boundaries, with the potential equal to the elevation. After every iteration step, the flux values of the nodes are checked, and if an inflowing flux is encountered, this node is treated as an impervious boundary in the next iteration step. On the other hand, if a positive value of pressure is encountered at a boundary node in the unsaturated zone, such node is treated in the next iteration step respectively as node located on a seepage face node boundary.

**Outflow Sea Face**

This condition applies at boundaries of either confined or unconfined aquifers, having a physical contact with a sea and through which the freshwater outflow to the sea as possible. This condition is similar to the seepage face condition except that here the density gradient of salt and freshwater needs to be taken into account, it is given by

\[
\begin{align*}
    h &= A - \delta z & \text{if } Q < 0 \\
    Q &= A & \text{otherwise}
\end{align*}
\]  

(3.47)

where \( A \) is the elevation referenced to the sea water level, and \( A \) is the freshwater potential at the outflow nodes. All the nodes which can possibly be on the outflow face are treated initially as prescribed potential boundaries, with the potential equal to the sea water level corrected for density difference, and during the iteration process if a negative outflow flux is encountered at a boundary node in the saltwater zone, in the next iteration step the flow rate at such nodes will be put equal to zero.
Variable Boundary Conditions

Because boundary conditions are subject to sharp temporal changes, and to the large variability of the medium properties which influence in turn the flow field, variable conditions give powerful capabilities to simulate situations as they would occur naturally. Examples are, horizontal infiltration where the flow rate specified at the air surface infiltrates an unsaturated zone above the water table surface; abstraction wells becoming dry at unsaturated layers where water can be pumped; and drainage systems, where the collected water is removed by an overflow system set at a given elevation. Such boundary conditions (infiltration, drainage or abstraction) are efficiently implemented and given special attention in the groundwater model routines as well.

Special nodes are also involved in many situations, such as isolated nodes which are completely surrounded by empty elements, i.e. where no airfoil type is considered. These nodes are excluded from the system of equations and computational routines, and no flow can be calculated at their locations. Typical examples are man-made holes such as mining excavations, galleries, and drainage systems.

3.5y Finite Element Matrix Analysis

Repeatedly generated systems of FE equations are ideally solved by an iterative solver for large dimensions. But, the choice of this solver depends mainly on the particular properties of the global matrix which should be used efficiently to gain in memory use and CPU consumption. Investigation of FE matrix properties also clarifies many numerical issues related to the solvers efficiency and the preconditioners existence.
3.5.1 Properties of the General Matrix

It is of great importance to investigate the shape of the conductance matrix resulting from the use of a hexahedral finite elements. Indeed, the matrix has some properties that make preconditioning possible and enhance the solver performance. The matrix satisfies the following conditions:

1. The matrix is sparse, because \( G_{ij} \) is equal to zero if \( i \neq j \) are nodes of the same element.

2. The matrix is symmetric, as can be concluded from equations (3.39 and 3.44).

3. The matrix is positive semi-definite (Axelsson and Barker, 1984).

A desired property for the conductance matrix is the so-called M-matrix property, which means that the following conditions have to be satisfied (Axelsson and Barker, 1984):

- \( G_{ii} > 0 \) for \( i = 1, 2, ..., n \)
- \( G_{ij} \leq 0 \) for \( i \neq j \)
- \( G^{-1} \geq 0.7 \)

In case of symmetrical matrices these conditions are less restrictive, and are equivalent to the following (Gustafson, 1984):

- The matrix is positive definite.
- \( G_{ij} \leq 0 \) for \( i \neq j \)

From equation (3.44) it is clear that,

\[
G_{ii} = \int_{V} \nabla b_{i} \mathbf{K} \nabla b_{i} \, dV + \sum_{e} \int_{V_{e}} S_{e}^{c} b_{i}^{c} \, dV^{e}
\]  
(3.48)
is always positive, and using the basis functions properties
\[
\sum_{j=14}^{nn} b_j = A1A \tag{3.49a}
\]
\[
\sum_{j=14}^{nn} \nabla b_j = 0 \tag{3.49b}
\]
it follows that,
\[
\sum_{j=14}^{nn} G_{ij} = A \sum_{e} \int_{V^e} S^e b_i^e dV^e \tag{3.50a}
\]
\[
G_{ii} = A \sum_{j \neq i} G_{ij} + A \sum_{e} \int_{V^e} S^e b_i^e dV^e \tag{3.51a}
\]
Substituting equation 3.48 to \(G_{ii}\) yields
\[
\sum_{j \neq i} G_{ij} = A \int_{V} K_y(\nabla b_i)^2 dV < 0A \tag{3.52a}
\]
This implies that some but not necessarily all of the off-diagonal terms are negative. For trilinear hexahedral elements, Larabi and De Smedt (1994) showed that the conductance matrix satisfies an M-matrix property if and only if all finite elements are cubes.

First and second type boundary conditions do not affect the general structure of \(A\). Third type boundary conditions only increase the diagonal dominance of \(A\) some rows.

### 3.5.2y Sparse Matrix Storage Scheme

Conventional array storage of the conductance matrix composed of \(24\) elements, requires usually more core computer storage than the hardware can handle. Therefore, use is made of the symmetry and sparsity of the matrix and only non-zero entries in the lower triangular part of the matrix are stored via an indexing algorithm that keeps element positions within the original matrix. Several compressed storage schemes have been developed for sparse matrices (Saad, 1994),
with the aim of gain in efficiency both in terms of memory utilization and arithmetic operations. It seems that the Compressed Sparse Row (CSR) format, and its variants as the Modified CSR (MCSR) and Diagonal CSR (DCSR) are the most popular because they are implemented in many computer packages. Similar schemes referred as the forward and backward structures, have been described by Nawalany (1986), and Zijl and Nawalany (1993).

Herein, a DCSR variant is used, and which is described as follows:

A compact row-wise real vector \( \mathbf{g} \) is used to represent the conductance matrix \( \mathbf{G} \), all non-zero entries existing in subsequent rows along the lower triangular submatrix of \( \mathbf{G} \). A direct relationship can be established between the dimensions of \( \mathbf{G} \) and \( \mathbf{g} \), respectively denoted by \( m \) and \( n_y \), as

\[
n_y = \sum_{i=14}^{m} k_i \tag{3.53}A
\]

where \( k_i \) is the number of non-zero elements in the \( i \)th row of the lower triangular part of \( \mathbf{G} \).

Two integer pointer vectors \( \mathbf{pc} \) and \( \mathbf{pd} \) are used to store respectively the numbers of the columns of subsequent non-zero elements, and the positions of the diagonal elements, of \( \mathbf{G} \) in vector \( \mathbf{g} \), such that

\[
pd_j = \sum_{i=14}^{j} k_i \tag{3.54}A
\]

\[
G_{i,pc_j} = \mathbf{g}_j \tag{3.55}A
\]

Since a two-way correspondence between \( \mathbf{G} \) and its compressed representation \( \mathbf{g} \), \( \mathbf{pc}, \mathbf{pd} \) is needed, this has to be clarified and established. A non-zero element \( G_{ij} \) \( (i \leq j) \) is retrieved as follows:

- For \( A = 1 \) and \( A = 1 \) \( G_{114} = \mathbf{g}(1)A \)
3.6 Solution Strategies

- For $\forall i > 1$, a search is performed on the elements of $pc_j$ to determine the element that is equal to a given $\delta_j$. The search can be limited to the range

$$k \in (pd_{i-1}, pd_i), \text{ when } pc_k = \delta_j, \text{ we find } \mathbf{G}_{ij} = \delta_k$$

Reciprocally, to find the element $\mathbf{G}_{ij}$ corresponding to a given element $\delta_k$, its column number is directly obtained from $pc_k$, and its row number is the maximum $i$, such that $pd_i < k.A$

3.6y Solution Strategies

The finite element system is usually solved by mesh-free iterative techniques, since the number of unknowns involved may be very large. The choice of a particular solver must suit the special system properties to gain in efficiency and robustness, the shape of the matrix stays the most decisive factor, i.e., symmetric or unsymmetric, dense or sparse, banded or random, etc. More discussions on the solver to be used within the computer packages being under study will be given in the next chapter.

Besides the numerical solver, several strategies exist for solving unsaturated flow problems in unconfined or multilayer aquifers, the choice of a given method involves a number of underlying approximations and limitations, but the application goals and interests remain the general guidelines for such compromise. For instance, the moving mesh procedure is quite efficient, practical, and economical for prediction of the water table fluctuations in a regional groundwater aquifer system. In contrast, for seasonal variations study of the saturation soil profiles involving infiltration and ponding, the unsaturated zone plays a key role, and the more general variably saturated groundwater flow formulation including the unsaturated soils parameters is more appropriate.
3.6.1 TheyMovingyMeshMethod

This technique involves an adaptive finite element mesh fitting the geometry of the external flow domain boundaries. The most naturally encountered free and moving boundaries in groundwater flow problems being the water table and the fresh-saltwater interface. The technique has been successfully used by several groundwater flow modelers worldwide (France, 1974; Desai et al., 1983; Bear and Verruijt, 1987; Larabi and De Smedt, 1993; Crowe et al., 1998). A nice feature of the method is that soil unsaturated properties are not needed, such that we may escape from the use of the water retention and the relative hydraulic conductivity curves in the model, which will decrease the problem in online-earity. However, a price is paid for such approximations, because under certain conditions unacceptable errors could be introduced to the solution, or convergence difficulties are present. To better illustrate these limitations, the following two descriptive examples are given.

1. yAFirstExample

Recharge from effective rainfall in an homogeneous shallow unconfined aquifer is prescribed at the most upper layer, which as the first layer to be adjusted during the solution procedure. Retardation affects an unsaturated zone where an important issue especially at the percolation zone as being thick. A This scenario not being taken into account in the moving mesh based simulators, the numerical solution is inaccurate in consequence, especially for time dependent scenarios involving a number of consecutive series of humid and dry periods.

2. yASecondExample

Partially penetrating well pumping a fixed amount of water flow, Q₀, is placed in an unconfined aquifer as illustrated in Fig. 3.5. At the top section of the well a filter of length Lₐ, is placed at a given depth, dₐ, from the soil surface. A
the water level at the well, $h_w$, is beneath the top filter section, the distributed pumping rate over the moving pumping nodes at the well face should be updated in parallel, this amount of water depends on the length of the saturated part of the filter, this adaptive procedure at the boundary nodes is tricky to implement. The problem still remains when the well becomes completely dry, because a user intervention is still required.

![Diagram of a partially penetrating pumping well](image)

**Figure 3.5:** Example of a partially penetrating pumping well withdrawing water over a portion of the filtered part.

Hence, unexpected problems may occur with the moving mesh strategy. The modeler experience plays a very important role here, since the interpretation of the model obtained in many situations should be performed with care.

### 3.6.2y TheyFixedMeshyMethod

Another possibility consists on adopting a fixed or ‘rigid’ network of elements, invariant in space and time. This concept is more widely used for modeling groundwater flow problems in unconfined aquifers (Cooley, 1983; Huyakorn et al., 1986; Paniconi et al., 1996), but in all these models the traditional approach
consider the characteristic soil curves to resolve the non-linearities in the model for the saturated zone. Drawbacks related to such approaches are classified into two main points:

1. Numerical instabilities: due to the irregular shape of the characteristic soil curves, and the difficulties surrounding an accurate representation of the capacitance terms. These effects are best described in the next chapter.

2. Inefficiency: most of computer codes using a compact matrix based solver, need to compute all matrix entries terms for the fixed mesh size elements, which need excessive computer time to achieve an accurate solution. These kind of problems for large 3-D problems has been the domain of high-end supercomputers and mainframe workstations for a long time ago. Even if several ports are existing nowadays for cheap desktop computers, CPU cost is still very high. Larabi and De Smedt (1997) have showed that the matrix entries depends upon the soil medium properties and the elements geometry which remains constant along the iterative process, they demonstrate that important computer time saving is achieved when keeping the fixed contribution the same all the time, so the relative hydraulic conductivity is the unique parameter which has to be adapted. However, this previous study was limited to steady state conditions and will be further extended for transient problems.
Chapter 4G

Modeling 3-D Transient Variably Saturated Groundwater Flow with Moving Interfaces

Contents

4.1 Introduction ................................................. 65
4.2 The FUP Numerical Technique ......................... 68
4.3 Numerical Solver ........................................... 75
4.4 Model Validation and Applications .................... 82
4.5 Summary .................................................... 103

4.1 Introduction

The problem of unconfined groundwater seepage is of a great interest in many fields such as in hydrogeology, civil and agricultural engineering, and hydrology. Practical applications are for example, seepage flow in earth dams for stability.
analysis; prediction of water table levels in a phreatic aquifer bounding an adjacent water body (river, canal, lake, stream, reservoir) whose water level fluctuates with time; performance of trenches which intercept contaminated groundwater; a bank storage due to fluctuations of water levels in rivers.

The solution of these problems is often complicated owing to the occurrence of a free or moving water table and seepage face which are unknown a priori, and should therefore be determined as part of the solution procedure. Analytical solutions for such problems are derived for two-dimensional groundwater flow (in vertical cross-section) under the Dupuit-Forchheimer assumption, which neglects the vertical flow component. Classic solutions for initial and boundary value problems of this kind are found in the works of Harr (1962), Polubarinova-Kochina (1962), Bear (1972, 1979) and Bear and Verduijn (1987); these solutions are limited to simple situations where hydraulic properties are uniform and main geometries are regular. These limitations have led to the development of numerical techniques using the FDM (Rubin, 1968; Freeze, 1971). However, it is not possible to construct a finite difference grid which fits exactly the curved water table position. The IFDM brings a further improvement, this method can handle easily domains of complex shape by constructing an irregular fitted net of elements, but the drawback of the method is that the elements should satisfy a given number of orthogonality conditions, which restricts its flexibility. In contrast, the FEM is more flexible in handling such difficulties, which explains its popular use and implementation in many groundwater flow numerical codes, as those presented by Neuman (1973), Yeh and Ward (1980), Cooley (1983), Huyakorn et al. (1984), Paniconi and Putti (1996), and Larabi and De Smedt (1997).

However, it is our point of view that more research is needed to implement a powerful and cost-effective solver for three-dimensional modeling with the standard finite element method, because of the highly involved cost and constructing and solving the algebraic numerical systems of equations, which becomes...
cumbersome for transient problems as the iterative solution procedure involves repeatedly generating equations for the desired solution. A solver should exploit efficiently the particular properties of the global FE matrix as discussed in Chapter 3. Furthermore, non-linearities inherent to the water table and seepage face it: a systematic approach, may lead to numerical instabilities. Another major difficulty which is typically encountered in solving groundwater seepage flow in a variably saturated flow domain, arises from the strongly nonlinear behavior of the Richards’ governing flow equation. Also, numerical approximation of the change in slope tangent of the water retention curve stemming from the matrix capacitance term is a complicated issue as reported by Paniconi and Putti (1996).

In this chapter, the developed numerical approach and its basic concepts which are used to solve the approximate FE matrix equations system derived in the previous chapter are introduced for solving problems with a moving free surface. The developed technique is formally called the Fast Iterative Procedure (FUP). Special consideration is made on computational efficiency in terms of CPU run-time and convergence speed. Care is taken of controlling numerical stability by effective handling of nonlinearities resulting from difficulties related to an approximation of the capacitance matrix coefficients, the chord slope, or from the occurrence of nonlinear boundary conditions (time dependent fixed heads, drainage, seepage, abstraction, etc.). Other essential issues for guaranteeing numerical stability will be discussed, such as the performance of the nonlinear iterative solver, the existence of linearized preconditioned conjugate gradient scheme for hexahedral finite elements, and the time stepping scheme.

Several test examples are studied to demonstrate the numerical technique capability of predicting accurate results efficiently. Comparisons are made against available analytical solutions, and other numerical schemes such as the moving mesh method. A comprehensive validation of the model predictions is achieved, by comparing to laboratory experimental measurements of free surface flow in an earth model of irregular shape (Baseghi and Desai, 1987). This approach...
lem is rather complex because it allows for real three-dimensional flow and for heterogeneous dam materials.

4.2y They FUPy Numerically Technique

This section will focus on the development of the FUP. The method is computationally fast because it avoids systematic reconstruction of the hole set of the FEA equations, and does not require recalculating the global matrix entries as given in Equation 3.44. This ‘fast’ FEA matrix reconstruction is performed in two steps. First, the relative hydraulic conductivity values in the conductance matrix terms are updated on an iterative basis, and second, the mass storage entries are approximated automatically from an idealized water retention curve as will be explained in the next two sub-sections. The two steps involve changes in the water suction potential values, and hence updating of the flow field.

4.2.1 Determination of Idealized Relative Hydraulic Conductivity

In the conductance matrix entries in Equation 3.39, two contributions can be recognized, the basis functions derivatives referring to the geometry of the finite elements, while $K_y$ refers to the hydraulic properties of the medium with respect to water flow. It follows that the coefficients depend upon the position of the water table. In the case where the nodes remain fixed, only the effective conductivity will be variable, as some nodes will be situated in the unsaturated zone. $G_{ij}$ can be approximated as proposed by Larabi and De Smedt (1997), as follows:

$$ G_{ij} \approx k_{ij} \int_V \nabla b_i K_y \nabla b_j dV = A^{ij} G_{ij}^s $$

(this means actually, that we assume that the relative hydraulic conductivity between nodes is independent of their positions in space and time, and will depend only on the water status of the region in between as shown in Fig. 4.1(a)).
Figure 4.1: (a) Location of water table between two nodes, and (b) Idealized relative hydraulic conductivity curve versus nodal pressure heads (note that $k_{ij} = 1$ only if both nodes are unsaturated).

Notice that the saturated conductance coefficients, $G_{ij}$, are constant, and hence remain fixed during the solution procedure, such that only the relative conductivities have to be computed again at each iteration. For instance, if nodes $A_i$ and $A_j$ are saturated water nodes it follows that $k_{ij} = 1$, otherwise $k_{ij}$ has to be updated. The following method is chosen to achieve this (Fig. 4.1(a)).

$$k_{ij} = \begin{cases} 1 & \text{if } p_i \geq 0 \text{ or } p_j \geq 0 \\ \varepsilon & \text{otherwise} \end{cases} \quad (4.2)$$

This is slightly different from the method used by Larabi and De Smedt (1997) for steady state water flow, suggesting that $k_{ij}$ to be smoothly updated for a water table region. In contrast, unsteady problems require immediate release of water from unsaturated to saturated nodes, such that small perturbations due to water table retardation effects are avoided. $\varepsilon$ is theoretically zero, but chosen here as a small number in order to allow for the finite element equations corresponding to unsaturated nodes to remain in the algebraic equation system, without ob-


structing the numerical solution procedure by making the matrix singular. This allows a small but negligible amount of water movement in the unsaturated zone, enabling recharge to pass from the soil surface to the water table through the vadose zone.

4.2.2y Idealized Watery Retention Curve

Most computer models use various constitutive or characteristic relations describing the soil storage properties. Herein, the updating of the nonlinear storage or time dependent term in the right hand side of Equation 3.44 is evaluated numerically in the FUP numerical technique. The nodal storage variation depends on the water table position, and is evaluated using a mass lumping scheme (Neuman, 1972), such that

\[ \int_{V_c} \frac{d\hat{\theta}}{dh} b_i \, dV \approx \int_{V_c} \frac{d\theta}{dh} b_i \, dV = d\hat{\theta} \frac{d\theta}{dh} V_i \] (4.3)

A clear physical interpretation of the mass lumped approximation for unsaturated flow, is that within each element the water content change is independent of the space domain. Whereas, in contrast, a mass distributed scheme assumes a trilinear distribution in the element. This is most probably the reason why a mass distributed scheme exhibits numerical oscillations. The ultimate advantage of using a lumped formulation is therefore that it is unconditionally oscillation free (Celia et al., 1990).

The method used to evaluate the derivative term in Equation 4.3 affects significantly the convergence behavior of the iterative schemes, due to steep gradients and discontinuities or points of inflection in the soil curves as shown in Fig. 4.2. A natural choice of an idealized moisture retention curve would be a step function given by

\[ \theta(p) = \begin{cases} \theta_s & \text{if } p_i \geq 0 \\ \theta_r & \text{if } p_i < 0 \end{cases} \] (4.4)
4.2 The FUP Numerical Technique

![Diagram showing the relationship between water content, θ, and pressure head, p.](image)

Figure 4.2: Water retention curves for specific soil types, and errors in the water capacity tangent approximation (modified from Istock, 1989).

However, this function suffers from the same disadvantages cited above, especially at the waterable position (p = θ), hence another method should be adopted. It is of great importance to understand physically the reason of failure of the above-mentioned procedure, while it seems to be attractive and simple. Actually, spurious oscillations are observed in cases where there are sharp pressure head variations near convergence such that waterable movement does not change enough to cross at least one node from top to bottom. An such case, due to a null storage variation, i.e., [S] = 0, a severe cancelation or an eventual jump to steady-state flow conditions occurs. Another interpretation of such behavior, is that nodal points are assumed to be physically either saturated or unsaturated, while the elements have three different water status, i.e., they may be saturated,
unsaturated, or partially saturated as depicted for a hexahedral element in Fig. 4.3. It turns out that these elements are the source of failure. The neglect of the partially saturated nodes in the formulation of Equation 4.4 is an illustrative example. As shown in the table at the left side of Fig. 4.4, for the corresponding hexahedral element, the relative position of the water table inside this element to a fictive plane joining specific points at the element z-vertices is depicted. For instance, if the free surface crosses a vertex at a position between this fictive plane and the top element plane, the lower node sharing this vertex will be considered to be saturated, while the other node becomes partially saturated. The distance separating a given node from that fictive plane, is interpreted as a sharp depth position from fully saturated or unsaturated state to partially saturated in the node neighborhood; this distance is evaluated as

\[ d_i = \frac{\Omega_i}{\Gamma_i} \]  

(4.5)

where \( \Omega_i \) is the control volume attributed to node \( i \), and \( \Gamma_i \) is the corresponding FE patch surface in the xy plane.

We define the soil moisture curve used in the described conceptual model as

\[ \theta(p) = \begin{cases} 
\theta_s - \theta_r & \text{if } p_i > + \frac{d_i}{2} \\
\theta_s - \theta_r + \frac{\theta_r - \theta_s}{d_i} p_i + \frac{\theta_r - \theta_s}{d_i} \frac{d_i}{2} & \text{if } |p_i| \leq + \frac{d_i}{2} \\
\theta_r & \text{if } p_i < - \frac{d_i}{2} \end{cases} \]  

(4.6)

and the specific water capacity function as

\[ \frac{d\theta S}{d\theta S(p)} = \begin{cases} 
\frac{\theta_s - \theta_r}{d_i} & \text{if } |p_i| \leq + \frac{d_i}{2} \\
0 & \text{otherwise} \end{cases} \]  

(4.7)

which means that the specific yield is released over the total length of each element if a nodal variation occurs in one node sharing it. Therefore, at least two nonlinear iterations are needed to achieve convergence.
**Figure A1.3:** An illustrative example of mapping saturated, unsaturated, and partially saturated nodes from a partially saturated element.

The functionals showed in Fig. A1.4 are used to achieve the numerical differ-A
entiation method adopted in this model. At the beginning of each time step the

differential function (b) is directly used to approximate the chord slope, in the
next iterations nodal water content values are relaxed following the function
(a) defined explicitly in Equation 4.6 and the differential expression is calculated
thereafter as:

\[
\frac{\Delta \theta}{\Delta h} = \frac{\theta(t + \Delta t) - \theta(t)}{h(t + \Delta t) - h(t)} \approx \frac{\theta^m(t + \Delta t) - \theta(t)}{h^m(t + \Delta t) - h(t)}
\]

(4.8)

where \( n \) is the iteration level. It can be noticed that Equation 4.8 does not exist whenever \( \theta(t + \Delta t) = \theta(t) \) which is especially the case at the beginning of each nonlinear time step. This explains the use of the analytic specific water capacity as defined in Equation 4.7.

There remains only an exception when starting from already conditions, i.e., if the initial water table position is put exactly at the lower layer of nodes. Under such initial conditions, the FUP has difficulty to start as the lower nodes in the finite element mesh will remain partially saturated, which is an unrealistic...
possible remedy is adaptation of the idealized water retention curve, by fixing the parameter \( \alpha_i \) for such nodes as small as possible \( (d_i = \Delta \rightarrow 0) \) and add this same preset value to initial heads \( h_i = \hat{h}_i + \varepsilon, \forall i \). Another strategy consists of adding another nodal layer beneath the actual mesh. A second choice may involve however an important number of nodes for large applications. However, it is often rare to meet such extreme conditions for large scale groundwater flow models, and this issue is only of importance for local scale simulations and theoretical scenarios.

![Diagram](image)

Figure 4.4: (a) Idealized water retention curve, and (b) analytic differentiation of slope tangent at nonlinear first iterate.
4.3 Numerical Solver

4.3y Numerical Solvery

Direct solution techniques are not attractive because they cannot handle matrices of large dimensions, and they do not make use of the special properties of the conductance matrix. Alternative techniques are more suitable, but a crucial issue is choosing a good iterative method from the many available, since any one of these methods may solve a particular system in a few iterations while diverging for other problems. Hence, preconditioned conjugate gradient methods are preferred, because they are highly successful, reliable, and more efficient for solving linear and positive definite symmetric systems, and also because of their reasonable cost per iteration.

4.3.1 Lineary Preconditioned y Conjugate Gradient y (PCG) y Solver

Among the existing iterative solvers, the conjugate gradient method (CG) is used mainly to solve positive-definite systems (Hestenes and Stiefel, 1952). This method is very popular for solving FD and FEA systems arising from groundwater flow equations. The method does not require the coefficient matrix, only the result of a matrix-vector product is needed. It also requires a relatively small number of vectors to be stored per iteration since its iterates can be expressed by a three-term vector recurrence. The convergence is theoretically guaranteed after an iteration, but in practice the algorithm converges after much less number of iterations. The CG algorithm is given in Table 4.1 in which the residual at the th iteration is \( A_k = A - B y - G h_k \).

Notice that the algorithm is more efficient when the matrix is symmetric and very sparse, because the heaviest operation is the matrix vector multiplication. If the matrix satisfies also the requirements of an M matrix the convergence will be even more faster.
(Axelsson and Barker, 1984). An important characteristic of the CG algorithm is its connection with the Lanczos method which allows to obtain estimates of the eigenvalues of the matrix with only little work per iteration, in this way Van der Vorst (1988) determined the condition spectral number, which is the ratio of the highest and lowest eigenvalues of the matrix, and reported that the CG convergence speed depends on this number. Therefore, the more resembles the identity matrix the faster the convergence, otherwise the matrix is all-conditioned and the algorithm requires a substantial number of iterations to converge. To overcome this problem, a transformation can be applied, which is commonly called scaling or preconditioning (Van der Vorst, 1989).

Table A.1: Unpreconditioned Conjugate Gradient Iterative Algorithm (Barrett et al., 1994).

<table>
<thead>
<tr>
<th>Choose an initial estimate $a_{04}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{04} = B y - G h_{04}$</td>
</tr>
<tr>
<td>For $k = 0, 1, 2, ...$ until convergence</td>
</tr>
<tr>
<td>$\alpha_k = A \frac{r_k^T \cdot r_k}{p_k^T \cdot G p_{kd}}$</td>
</tr>
<tr>
<td>$h_{k+1} = a_k - \alpha_k p_{kd}$</td>
</tr>
<tr>
<td>$r_{k+1} = r_k + \alpha_k G p_{kd}$</td>
</tr>
<tr>
<td>$\beta_k = A \frac{r_{kd}^T \cdot r_{k+1}}{r_k^T \cdot r_k}$</td>
</tr>
<tr>
<td>$p_{k+1} = p_{k+1} + \beta_k p_k$</td>
</tr>
<tr>
<td>End For</td>
</tr>
</tbody>
</table>

### 4.3.2y Preconditioning

Preconditioning accelerates greatly the convergence behavior of CG methods, which becomes necessary in dealing with practical applications of large size. The
idea behind it is to multiply the system by a matrix $\mathbf{C}^{-14}$ that resembles $\mathbf{G}^{-1}$. A
Basically, the algorithm does not change except multiplying by $\mathbf{C}^{-14}$ whenever $\mathbf{A}$ is required. Using initial guess values of the potentials, $\mathbf{A}_{04} = \mathbf{C}^{-1} \mathbf{B} \mathbf{y}$ (Gustafsson, 1984; Gambolati, 1988b) leads to the preconditioned algorithm as shown.

### Table 4.2: Preconditioned Conjugate Gradient Iterative Algorithm (Barrett et al., 1994)

<table>
<thead>
<tr>
<th>Step</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{04} = \mathbf{C}^{-1} r_{04}$</td>
<td></td>
</tr>
<tr>
<td>For $k = 0, 1, 2, ...$ until convergence</td>
<td></td>
</tr>
<tr>
<td>$\mathbf{A} \mathbf{r}_k^T \cdot \mathbf{C}^{-1} \mathbf{r}_k$</td>
<td></td>
</tr>
<tr>
<td>$p_k' \cdot \mathbf{C} \mathbf{p}_{bd}$</td>
<td></td>
</tr>
<tr>
<td>$h_{k+1} = p_k - \alpha_k p_{bd}$</td>
<td></td>
</tr>
<tr>
<td>$r_{k+1} = \mathbf{A} r_k + \alpha_k \mathbf{G} p_{bd}$</td>
<td></td>
</tr>
<tr>
<td>$\beta_k = \mathbf{r}<em>k' \cdot \mathbf{C}^{-1} \mathbf{r}</em>{k+1}$</td>
<td></td>
</tr>
<tr>
<td>$p_{k+1} = \mathbf{C}^{-1} r_{k+1} + \beta_k p_k$</td>
<td></td>
</tr>
</tbody>
</table>

The preconditioning matrix $\mathbf{A} \mathbf{C}$, which is close to $\mathbf{G}$, is considered as a good estimate if it fulfills the following conditions:

- The condition spectral number of $\mathbf{G} \mathbf{C}^{-14}$ is less than that of $\mathbf{G}$;
- The eigenvalue distribution of $\mathbf{G} \mathbf{C}^{-1}$ is more favorable to the CG algorithm than that of $\mathbf{G}$ itself;
- The coefficients of $\mathbf{C}^{-14}$ should be easily determined and do not require excessive storage.
However, these conditions restrict the choice of a good preconditioner. The methods proved to be of value when they are used in conjunction with CG like methods: Adiagonal scaling (DS), incomplete Cholesky decomposition (IC), incomplete factorization (IF), modified incomplete factorization (MIF), and so on. II preconditioners, except DS are derived from the class of incomplete tri-Aangular factorization of \( \mathbf{G} \), in this case we set \( \mathbf{C} \mathbf{y} = \mathbf{L} \mathbf{U} \), where \( \mathbf{L} \) and \( \mathbf{U} \) are lower and upper triangular matrices.

The choice of the best preconditioner is still a matter of debate, indeed in each particular situation a method can perform better than others. It was also shown that their efficiency is hardware dependent. DS is best suited for vector supercomputers, while IC is better in scalar/distributed memory computers. Thus there is no general rule of thumb. It is also important to point out that the performance of these preconditioners depends on how they are coded, i.e., vectorized, parallelized, etc.

**Incomplete Factorization Preconditioner**

In this study IF preconditioning is preferred, i.e.

\[
\mathbf{C} \mathbf{y} = (\mathbf{L} \mathbf{y} - \mathbf{D}) \mathbf{D}(\mathbf{L} \mathbf{y} - \mathbf{D})^T
\]  

where \( \mathbf{L} \mathbf{y} \) is the strictly lower triangular part of \( \mathbf{G} \), and \( \mathbf{D} \mathbf{y} \) is a positive diagonal matrix, such that \( \text{diag} \left[ \mathbf{C} \right] = \text{diag} \left[ \mathbf{G} \right] \) (Meijerink and Van der Vorst, 1977). Only the entries of \( \mathbf{D} \) need to be computed, thus the method does not require much additional storage and computational work. The entries of the diagonal matrix \( \mathbf{D} \) can be computed recursively as

\[
D_{ii} = G_{ii}^2 - \sum_{k=14}^{i-1} \frac{C_{ik}^2}{D_{kk}}
\]  

Larabi and De Smedt (1994) concluded after many comparisons based on several test problems, including hypothetical and field applications, that the preconditioned conjugate gradient (PCG) method based on DS is the most robust because...
it never fails, which is due to the fact that the preconditioning matrix always exists. However, this is not always true for other preconditioners. In such cases, the AMA property of $G_Y$ proves to be a key factor for the successfulness of these solvers (Meijerink and Van der Vorst, 1977), because under this condition all preconditioners are guaranteed to exist. Therefore, in this study, we will use an incomplete factorization preconditioner obtained on the $M$ type transformed conductance matrix as recommended by Larabi and De Smedt (1994).

### 4.3.3y M-Matrix Transformation Procedure

The flow domain is often divided into irregular finite elements, and leads naturally to a conductance matrix which is not an $M$ matrix. Indeed, the more irregularity in the shape of the elements, the higher the deviation from the $M$ matrix property. This will greatly hamper the numerical solution procedure, because of some constraints related to the existence of preconditioners as discussed in the previous paragraph. Larabi and De Smedt (1994) suggested that, for obtaining a preconditioner, $G_Y$ can be transformed to $A^T G_M A$ M-matrix by maintaining all negative off-diagonal terms of $G$, while all positive terms of $G_Y$ are added to the diagonal; hence:

\[
(G_M)_{ij} = A \min(G_{ij}, 0) \quad (4.11)A
\]

\[
(G_M)_{ii} = A \sum_{j=1}^{n} \max(G_{ij}, 0) \quad (4.12)A
\]

In the next paragraph, we will show that $A^T G_M A$ is a good $M$ matrix estimate of $G_Y$ and, that an incomplete factorization preconditioner used in conjunction with the conjugate gradient solver is always guaranteed.

### 4.3.4y Modified Nonlinear Picard Iteration

Among the most popular linearization schemes are Picard and Newton-Raphson methods (Huyakorn and Pinder, 1983; Astok, 1989), with the Picard method being...
more popular because it is easier to implement, cheaper than a single iteration basis and does not need larger storage requirements or changes in the system structure. In contrast, the Newton method involves additional costs related to storage and approximation of derivatives of the Jacobian, and leads to an unsymmetrical nonlinear system, which restricts the linear solvers choice. A comprehensive comparison between these two strategies with several test cases has been carried by Paniconi and Putti (1996), who concluded that the Newton method is faster in case the initial estimate is good enough, otherwise convergence performance may be poor, and they propose a mixed approach for a remedy. Herein, a modified Picard scheme is preferred because we believe that combined to the developed M matrix based linear PCG solver and the FUP, a robust and yet more stable solution method is obtained as will be demonstrated further in this chapter.

To solve the nonlinear FE system of equations at time step \( t_{k+14} \) starting from the initial potential distribution \( \Phi(x, t_k) \), the modified Picard algorithm involves the following steps described below for \( \Phi_t = 1, 2, ... \)

1. The new position of the water table is determined at iterate \( \Phi_t \) using Equation 3.15, where \( \Phi_{m+1}^{t} \) denotes the \( m^{th} \) iterate of \( \Phi(x, t_{k+1}) \);

2. The global conductance matrix is adjusted using Eqs 4.1 and 4.2;

3. The capacitance matrix terms are adjusted using Eqs 4.3 and 4.6 to 4.8;

4. The linearized system of FE equations is solved using standard conjugate gradient solvers, preconditioned with point incomplete factorization method enhanced with an automatic M-matrix transformation as described earlier, which yields a potential distribution \( \Phi_{m+14}^{t} \);

5. For the next iteration, an improved estimate of \( \Phi_t \) is derived from

\[
h_{k+14}^{m+1} = \Phi_{m+1}^{t} \left( h_{k+14}^{m+14} - h_{k+1}^{m} \right) + h_{k+14}^{m} \tag{4.13}
\]
where $A\omega_{m+14}$ is an automatic underrelaxation factor used here in order to amortize possible oscillations of the potential iterates. An optimal value of $\omega_{m+14}$ is determined upon convergence rate at the previous iteration.

6. The iterative procedure described above steps A to 5 is repeated until the following convergence criterion is satisfied:

$$e_{\text{max}}^{m+14} = \max \left| h_{m+14}^{k+1} - h_{m}^{k+1} \right| < tolw$$  \hspace{1cm} (4.14)

but, only over nodes lying in the saturated zone domain, while $tolw$ is a prescribed groundwater potential tolerance.

**Relaxation Technique**

Relaxation is suggested to enhance nonlinear iterative schemes, as by Cooley A (1983) and Huyakorn A. A (1986), this is because iterations could be slowed for oscillations to occur. Here we used an adaptation of the Huyakorn's procedure (1996) which quantifies $A\omega_{m+14}$ as a function of the convergence rate in the previous linear iteration, we suggest the following expression:

$$\omega_{m+14} = \max \left[ \min \left( \frac{e_{\text{max}}^{m+14}}{\Delta h_{\text{max}}} \right), 1 \right], \omega_{\text{min}}$$  \hspace{1cm} (4.15)

where $(\Delta h_{\text{max}} = \max \left| h_{m+14} - h_{\text{min}} \right|)$ is the absolute value of groundwater potential at the point of extrema over all nodal values, and $\omega_{\text{min}}(0 < \omega_{\text{min}} < 1)$ as a minimal preset value. Typically, this has a greater affect at the beginning of the iterations, and as the solution procedure continues it is obvious that $A\omega_{m+14} \to \omega_{\text{min}}$ where $e_{\text{max}}^{m+14} \to 0$, especially when convergence is close. The technique shows to be very effective in accelerating the convergence of the nonlinear iteration by an order of magnitude of few iterations at the beginning of the process.

**4.3.5 Time Stepping Scheme**

The model automatically adjusts time step sizes, in order to avoid possible in-stabilities during the solution. $A_{t_{\text{max}}}$ is the maximum number of linear iterations; $A_{t_{\text{max}}}$ is
allowed for each nonlinear time step, and if exceeded without convergence the solution is recomputed at the current time level ‘back stepping’ using a reduced time step size. At the computation continues until a maximum preset time \( \Delta t_{\text{max}} \). This time step is calculated following expression A

\[
\Delta t = \min\left( \frac{S_{\text{tol}} w}{B_i - \sum_j G_{ij} h_{ij}^k} \right) \tag{4.16}\]

However, if a fixed time step value \( \Delta t_{\text{user}} \) is specified the simulator will assume time step sizes as integer multipliers \( (2\Delta t_{\text{user}}, 3\Delta t_{\text{user}}, \ldots) \) of this preset value. AIf necessary, the time step is adjusted to coincide with a target time value \( \Delta t \) at which simulation output is required.

In conclusion, three nested iterative loops are necessary to build the overall numerical flow solver. They are from the outer to the most inner one, the time stepping loop, the nonlinear Picard iteration, and the linear matrix based incomplete factorization preconditioned conjugate gradient method.

### 4.4y Model Validation and Application

#### 4.4.1 Naturally Drainage in a Yinyay Soil Column

**Problem Definition**

Water drainage in a wet vertical soil column of length \( L \) and unit cross section \( S = 1 \times 1 \text{ m}^2 \) as depicted in Fig. A.4.5 as investigated, assuming an hydrostatic initial pressure head distribution in the entire domain. At \( t = 0 \), the pressure head at the flow outlet becomes equal to the flow, the soil column starts draining, and an unsaturated zone will develop from the top to the bottom. The an moving water table position \( h(t) \) is predicted using the developed FUP, and compared to other solutions including the moving mesh solution method and the exact analytical solution.
Analytical Expressions

Assuming an instantaneous drainage in an porous medium (Boussinesq's approximation), the discharge rate through the saturated column length is constant, and equals $Q_s = KS$. The drained volume of water at time $t$ is calculated equivalently using one of the following expressions given as

$$V(t) = A \int_{t_0}^{t} \frac{Q_s}{n} \, dt = (\zeta(t) - L)S$$  \hspace{1cm} (4.17)$$

which yields a simple linear decreasing water table height

$$\zeta(t) = A - \frac{K}{n} t$$  \hspace{1cm} (4.18)$$
Numerical Results and Discussion

We assume a 10m length fine sandy soil column, having a saturated hydraulic conductivity \( K = 1 m/d \) and an effective porosity \( \epsilon_e = 0.25 \). Although the problem is naturally 1-D, 2 nodes along plane x and y directions are necessary to run our 3-D model. A finite element mesh is composed of 100 vertically-ordered hexahedral elements, thus 4 nodes per element. All models are executed for 24 days simulation time, with a user-specified time step of 0.1 days, and a predefined \( \Delta t \) parameter equal to \( 10^{-3} \). At each time step, a solution, sharp
water table position occurring at zero pressure head value as linearly interpolated from obtained values at nodal positions.

Decreasing water table heights are plotted in Fig. 4.6 with a time increment of 0.1 days. The numerical solution exhibits very small deviations from the analytical solution, while the moving mesh procedure yields excellent results (Fig. A4.7). This is interpreted as a consequence of discarding the unsaturated zone, because water is released immediately at the water table nodes located at the top of the newly adjusted mesh, which somewhat comply with Boussinesq’s approximation used to
Figure A1.8: Water content profiles at specific target times. Dots indicate nodal water positions.

Figure A1.8: Water content profiles at specific target times. Dots indicate nodal water positions.

derive the analytic equation. Soil moisture content profiles at selected times plotted in Fig. 4.8 shows that the saturation process at a given elevation point is by no means linear in time. One should not confuse the linear water retention curve model which is locally dependent on the moving water table position, and the overall global representation at the end of the solution procedure.

The efficiency of the two calculation methods is investigated in terms of run-time, CPU(s); total number of inner PCG iterations, NI(PCG); and total num-
Table 4.3: A Comparison of the FUP and Moving Mesh Solver Efficiency for the Test Problem y

<table>
<thead>
<tr>
<th></th>
<th>CPU(s)</th>
<th>NI(PCG)</th>
<th>NI(PICARD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUP</td>
<td>1.1A</td>
<td>545A</td>
<td>111A</td>
</tr>
<tr>
<td>Moving mesh method</td>
<td>5.4A</td>
<td>725A</td>
<td>157A</td>
</tr>
</tbody>
</table>

The number of outer Picard iteration loops required to satisfy convergence requirements, NI(Picard)\(^1\). These values are given in Table 4.3 for the FUP and moving mesh A methods. The FUP technique is faster, and requires fewer number of total PCG iterations and Picard iteration loops, meaning that the successfully updated matrix equations are much easier to solve. However, saving an CPU time is largely due to the implicit reconstruction of the conductance and capacitance matrix terms as explained in Section 4.2.

4.4.2y Drainage of a Soil Column Through a Leaky Outlet Problem

This problem is taken from Ababou et al.\(^1\) (1998) who solved the same problem with a partially saturated finite volume based approach. This test case is basically similar to the previous example, except that a soil medium of lower hydraulic conductivity, \(K_b \ll K\) and small length \(A < L\) is placed beneath the original soil column as illustrated in Fig. A.9.

**Analytically Developmentsy**

Following again the Boussinesq's approximation, the flow rate remains the same through the upper and lower unit cross-sections of the outlet, such that direct application of Darcy's law at these two cross-sections yields

\[
Q_s = K \frac{\zeta - h(b)A}{\zeta - b} = K_b \frac{h(b)A}{b} 
\]

\(^1\) All test runs in this chapter are executed on an Intel CPU platform (Pentium-166 with Intel CPU! classI, and 64MB RAM).
Figure 4.9: A schematic view of a soil column draining through a leaky outlet. 

such that the groundwater potential at the upper outlet section, \( h(b) \), can be deduced as:

\[
 h(b) = \frac{(\beta + b)\zeta}{\beta + \zeta} \tag{4.20}
\]

where \( \beta = \log \left( \frac{K}{K_b} \right) - 1 \). Substitution of Equation 4.20 into one part of Equation 4.19 yields:

\[
 Q_s = A\kappa \frac{\zeta}{\beta + \zeta} \tag{4.21}
\]

which is a relationship relating the discharge rate at the outlet to the water table height \( \zeta \).

Application of Equation 4.17 to the expression derived in Equation 4.21 yields the analytical form of the moving free surface position \( \zeta \):

\[
 \zeta + \beta \ln \left( \frac{\zeta}{L} \right) = A - \frac{K}{n} t \tag{4.22}
\]
Therefore, the water table height $h$ is expected to decrease with a much lower rate than in the first example.

**Comparison with Numerical Results and Discussion**

Now, we take a soil column of 80m length, in which a coarse sandy soil is placed above a 10m height clayey outlet. Hydraulic conductivity and effective porosity values are respectively $K = 10m/d$, $n = 0.3$; $K_b = 0.1m/d$ and $n_b = 0.6$ for the macro-porous and the leaky outlet. 80 box shaped elements and a total number of 324 nodes are used to approximate the entire domain. The model runs...
Figure A4.11: Water table drawdown law \( \zeta(t) \), FUP and moving mesh numerical methods are compared to the analytical solution.

For 80 days with a prescribed \( A_0 \) parameter equal to \( 10^{-3} \); this time is quite sufficient for the soil column to drain completely, as an estimate can be made a priori from Equation 4.22 for \( A_0 = A \). The computational time step is fixed to 0.5 days and the numerical results are obtained within a period of 4 days. Predicted discharge rate values at the analytical curve perfectly as shown in Fig. A4.10, except close to the end, when the water table crosses the two mediums interface. The simulated moving water table fits exactly the analytical one as shown in Fig. A4.11; such that it can be concluded that the FUP model is able to predict accurately the time for the sandy soil to become dry.
The effect of the outlet low permeability on retarding the water table decrease, could be deduced also from soil moisture content profiles at target times shown in Fig. A4.12.A

Here again, the FUP based solver performance is checked against the moving mesh method, it is shown from Table 4.4 that still the superiority of the new technique is preserved.

<table>
<thead>
<tr>
<th></th>
<th>CPU(s)</th>
<th>NI(PCG)</th>
<th>NI(PICARD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUP</td>
<td>98.3A</td>
<td>13722A</td>
<td>791A</td>
</tr>
<tr>
<td>MOVING MESH METHOD</td>
<td>197.3A</td>
<td>6620A</td>
<td>725A</td>
</tr>
</tbody>
</table>

### 4.4.3 SeepageyinyayReservoiryfromyaySemi-InfiniteyUncon-yfinedyAquifery

**Problem Definition**

This is another academic test problem: a two-dimensional groundwater flow in a semi-infinite aquifer of rectangular shape. The natural drainage of a water table aquifer starts when the water level in a bounding reservoir is lowered from $h_{04}$ to $h_{14}$ as illustrated in Fig. A4.13.A The governing equation in this case is the well-known 1-D Boussinesq's equation

$$ \frac{\partial h}{\partial t} = \frac{K_s}{n_e} \frac{\partial}{\partial x} \left( h \frac{\partial h}{\partial x} \right) $$

(4.23)

Initial and boundary conditions are defined in such case as

$$ h(x, 0) = \hat{h}_0, \quad h(0, t > 0) = \hat{h}_{14} \quad \text{and} \quad h(\infty, t \geq 0) = \hat{h}_{04} $$

(4.24)
Analytical Solutions

Several attempts were taken to derive approximate and exact analytical solutions for this problem, which can be found in a number of standard and comprehensive textbooks. For instance, solutions given by Bear (1972; 1979) are established by using linearized forms with respect to $u$ and $u^2$ as the dependent variable in the original equation (see Appendix A.1). A more accurate solution was presented earlier by Polubarinova-Kochina (1962), which is given in the form of $u = A_1 u$, where $A$ is a truncated power series expansion to the forth term of the parameter $A$. 

Figure 4.12: Water content profiles at target times.
Figure 4.13: A schematic view of test problem 3.w

\[ l = \frac{h_{04}-h_{14}}{h_{14}} \]

\[ u = \sum_{m=04}^{+\infty} l^m u_m \cong 1 + l u_{14} + l^2 u_{24} + l^3 u_{34} + ... \]  \hspace{1cm} (4.25)A

where \( A_{14}, u_2, \) and \( A_{34} \) are functions of the dimensionless parameter \( A = \frac{x \sqrt{n_e}}{2 \chi k h_{14} t} \); analytical expressions of these functions and a set of tabulated numerical values are given in Appendix A.2. However, Equation A.5 is not valid for the extreme case of a seepage into an empty stream channel, i.e., when \( A_{14} = 0 \), this is not mathematically neither physically correct. In such situation the following expression was derived alternatively:

\[ h(\eta, t) = 2.365 h_0 (y - 2y + 3y^7 - \frac{4A_{104}}{11A} - ... ) \]  \hspace{1cm} (4.26)A

where \( \eta = 0.4873 \sqrt{\eta} \), the discharge flux to the stream is evaluated as:

\[ q_{x=0} = 0.332 \left( K_s n_e \right)^{1/24} h_{04}^{3/2} t^{-1/24} \]  \hspace{1cm} (4.27)

Another solution based on the Boltzmann transformation is recently proposed by Guo (1997), which has the feature of minimizing computational costs in comparison to the Polubarinova-Kochina’s solution. We find it comprehensive to recall some key assumptions under which the Boussinesq’s equation is derived, namely the horizontal flow or Dupuit-Forchheimer
approximation, which is known to be not valid near the stream channel where the vertical flow component cannot be neglected (Bear, 1972), this is more obvious as \( \frac{h_0}{h_1} >> 1 \) or \( l >> 1 \), and also in case of relatively deep unconfined aquifers. Another limitation of the analytical solution is the inability to predict the seepage face extent. These limitations are not applicable to the numerical model being under study, and differences with the expected numerical results may therefore occur. To be able to interpret these deviations in numerical results, a couple of simulations is performed by disabling and enabling respectively the seepage face detection procedure. The first run is especially chosen to comply with the analytical prescribed head near the stream, while the second simulation is judged to be more realistic as it will occur in either laboratory or field conditions.

![Finite element mesh for test problem 3.](image)

**Figure 4.14:** Finite element mesh for test problem 3.

**Comparison with Numerical Results and Discussion**

The parameters for running the numerical simulations are \( h_0 = 1m, h_1 = 0.5m, K = 1m/day, n_e = 0.1; \) and 1m fixed groundwater potential at 4m downstream the reservoir, assuming that the flow is stationary landward. The finite element mesh used in this example has a uniform spacing of 0.1m as shown in Fig. 4.14.

The total simulation period \( T_{\text{max}} \) is 1 day, and the dynamic time step size con-
trol algorithm is enabled. Numerical solutions of the groundwater potential distributions, \( h(x, t) \); the free surface positions for runs 1 and 2; and Polubarinova-Kochina’s analytical free surface are simultaneously plotted in Fig. 4.15.

![Graphs showing groundwater potential distributions](image)

Figure 4.15: Comparison of moving numerical and analytical water tables; contour heads with an interval of 0.05m are also plotted. Dots represent the analytical solution.

**First Run**

For the first run no seepage face condition is set at the outlet face. The results show that drainage is faster than expected, especially at starting time levels, but becomes less pronounced as the time increases. Groundwater potential contours
are plotted within a regular interval of 0.05m, it shows important gradients (and hence velocities) at the outlet boundary of the reservoir, diminishing in time. We can think that not allowing the numerical seepage face existence has led to these difficulties, and explains the motivation for the second run.

**Secondaryrun**

In this run the seepage face is taken into consideration. It is observed that the newly simulated water tables fit more accurately the analytical solution where the previously simulated ones have the maximum deviations, while deviations near the reservoir are obvious due to the limitation of the analytical solution. These conclusions shed more light on the simplifications of some 'exact' analytical solution, and the superiority of numerical modeling techniques.

The efficiency of the FUP is again investigated for this test problem, these results are shown in Table 4.5. Also, for previous examples the FUP was found to be very efficient. Due to sharp water table variations, time steps were automatically adjusted, and the simulations needed additionally 2 time levels (different than output time levels) at 0.02days and 0.0448days which is not excessive but necessary to prevent numerical oscillations around the true solution, which proves that the used time stepping scheme is effective and well implemented.

| Table 4.5: FUP solver performance for test problem y3, y4 |
|-----------------|----------------|------------------|
| CPU (s) | NI (PCG) | NI (Picard) |
| FUP (Run 2) | 12.2A | 360A | 53A |

4.4.4y Validation with a Three-Dimensional Laboratory Model

**Background and laboratory model description**

In general, practical simulations involving moving surface seepage in field conditions are complex due to local medium heterogeneities, non regular geometries,
and time dependencies. A closed form analytical solutions for these problems are hard to formulate and are limited as discussed in the previous section. Hence, to establish a confidence in the numerical model results, validation by using experimental or/and field measurements is an asset for the developed model. In this study, the results of an earlier experimental tests are used to validate the FUPA model, these laboratory experiments were carried in department of Civil Engineering and Engineering Mechanics (University of Arizona, Tucson) by Baseghi and Desai (1987), which is a useful contribution since it is a unique study that reported laboratory observations on three-dimensional free surface flow.

Laboratory tests were performed on a three-dimensional glass bead model. The model configuration allows for simulation of homogeneous and nonhomogeneous materials such as core in dams, which are simulated by glass beads of different diameters (1 mm and 3 mm). Anisotropic flows are used to approximate the Asole domain, in which specific actions are of interest, especially in the front section, F; back sections, B1 and B2; and side section S. During the experiments, transient movement of free surface was recorded photographically along these sections, upon transient (rise, steady-state, drawdown) fluctuations on the upstream section of the earth dam model. These observations are compared to numerical predictions from the presented model.

Model Parameters

Hydraulic conductivity values for granular glass beads are reported by the authors (Baseghi and Desai, 1990), and have been determined using a constant head laboratory test, specific storage coefficients were also calculated. But, neither specific yields nor porosities are given for the used materials. This is a limiting factor for the simulations to be performed, but a relatively acceptable estimation of the porosity is found from the Kozny-Carmen equation (Freeze and Cherry,
Figure 4.16: Configuration of the laboratory model showing the finite element mesh, the photographed sections, and the core dam location.

1979) relating it to the hydraulic conductivity by

\[ K = \left( \frac{\rho g}{\mu} \right) \frac{n^2}{(1-n)^3} \left( \frac{D^2}{180} \right) \] (4.28)

where \( D \) is the mean-size granular soil diameter [L]. Equation 4.28 yields the following equation

\[ 1 - 3n + (3 - \lambda)n^2 - n^3 = 0 \] (4.29)

where \( \lambda = \frac{(\rho g)D^2}{180K\mu} \), \( n \) is therefore the root of the polynomial Equation 4.29 which satisfies necessarily the condition \( n \in ]0, 1[ \).
Comparison of predicted (continuous lines) and observed free surface during transient and steady state for the homogeneous dam. \( F \) (front section), \( B_1, B_2 \) (back sections); \( S \) (side section).

Glass beads of 1 mm diameter are used in this experiment. Starting from dry conditions (e.g., \( h = 0 \) cm everywhere), the upstream water level is raised to a 17.4 cm in about 20 min, and maintained at that level for about 140 min. The upstream water level is decreased thereafter with a fixed rate of 0.96 cm/min. Measured values were recorded at 4 min, 8 min, and 12 min for the rising phase, and at 156 min, 158 min, 162 min, and 164 min for the drawdown stage. Accordingly, these water levels are specified as input to the model. Comparisons...
between the FUP numerical predictions and the observed water table positions are shown in Fig. 4.17 for the rise and steady state conditions and in Fig. 4.18 for the drawdown stage. It is clear that the developed numerical procedure produces satisfactory predictions of either the free surface or the seepage face height. The observed deviations at given time levels, are due in large extent to inaccurate estimations of the saturated and residual water contents in our models. A parametric study to estimate these values is possible by trial and error calibration procedure with the observed steady state measurements on a long term transient simulation basis, however this would need more effort which is beyond the scope of the present study.
Figure 4.19: A comparison of predicted (continuous lines) and observed free surface during transient and steady state for the heterogeneous dam. 

- (F) front section; 
- (B1, B2) back sections; 
- (S) side section.

Note on Computer Time

It took about 3 CPU hours on average per time step on a CYBER A05 supercomputer of 10 years ago (as reported by the authors) for their 3-D model, and 2 CPU s for the present FUP model on a simple scalar desktop PC platform. This shows the rapid evolution of computer hardware and computational methods at the last era of the current century.
Casey2: Heterogeneous Dam

This is a more difficult and challenging problem, because the dam is sectioned as shown in Fig. 4.16. In the core area the 1 mm size glass beads are used while 3 mm size beads are placed elsewhere. The upstream head variation for the rise period is similar to the first experiment, the maximum water level is maintained for 40 min, and decreased at a much faster drawdown rate of 9.33 cm/min. Observed outputs are taken at 4 min, 7 min, and 12 min for the water table rise, and at 60.5 min, 60.7 min, 61 min, and 61.5 min for the drawdown. The difficulty of this problem arises from the important drawdown for a steep time period, which...
need special attention. Here again dynamic time step sizes control is turned on, and very small time steps were needed at the beginning to relax the numerical solution. Comparison between predicted and observed values was very good for the two stages, i.e. rise, steady state and drawdown of the free surface as shown in Figures 4.19 and 4.20. The seepage face height is also predicted accurately.

4.5y Summary

A computer model for prediction of three-dimensional groundwater flow involving a moving phreatic boundary is developed based on the Galerkin finite element approximation in space and a fully implicit finite difference time approximation with a mass lumped capacitance term.

The embedded numerical approach does not completely neglect the flow in the vadoze zone, but it is assumed to be a small fraction of magnitude as in the saturated domain. The moving water table boundary is iteratively adjusted based on nodal water status (i.e. saturated, unsaturated, partially saturated) which are deduced from the relative water table position within each element. The FUPA technique is shown to be cost-effective and efficient due to inexpensive update of the conductance matrix, and accurate estimation of the capacitance terms, which are less expensive in comparison with standard approximation methods.

The overall numerical solver is robust and implements attractive state of the art features and powerful reputed algorithms, such as the modified incomplete factorization preconditioner based on a M matrix transformation, a linear conjugate gradient solver for the most inner loops, and a dynamic time stepping scheme with automatic determination of the under-relaxation factor for updating groundwater heads to avoid possible numerical oscillations and/or instabilities. The numerical model also enables various types of complex nonlinear boundary conditions (seepage face, drainage, time-varying heads, leakage, abstraction, etc.)

The usefulness of the FUP approach and the developed model is clearly put
in evidence, based on a series of test problems. These examples are of different complexity, dimensions, and groundwater flow behavior. Numerical results are compared to analytical solutions whenever they are available, and show very good agreement. Comparisons are also made with the moving mesh method, which is slower in comparison to the FUP for transient nonlinear problems. The last test validates the model by comparison with laboratory measurements in a 3-D earth dam model. The model structure allows for homogeneous and heterogeneous formations with more dams. Satisfactory agreement is observed in case of rise, steady state and drawdown of the free seepage flow for all these experiments.
Chapter 5G

A 3-DySharpInterfaceApproach for Modeling Seawater Intrusion in Coastal Aquifers

Contents

5.1 Introduction .................................................. 105
5.2 Conceptual Model ............................................ 107
5.3 The Generalized FUP Approach ......................... 109
5.4 Validation and Application Examples ................. 113
5.5 Model Application to Seawater Intrusion in Martil Aquifer ........................................... 130
5.6 Summary ........................................................ 142

5.1 Introduction

Coastal aquifers often involve complicated and varying conditions in time and space, owing to the occurrence of a moving fresh-saltwater interface, rather than a natural stationary interface. Initially existing A Practical Examples include A pumping stations Aof Avariable Ascheduled Awithdrawal; Aartificial Arecharge Afor Asite Aremediation; Aconstruction of cutoff walls and barriers; and other possible scenarios Afor Aseawater Aintrusion Acontrol. A Moreover, unconfined flow in coastal aquifers A
involves additionally a moving water table as discussed in the previous chapter, so adding further complexity.

Three-dimensional numerical modeling of saltwater intrusion is investigated in case of a sharp fresh-saltwater interface approach, neglecting mixing of the two fluids, meaning that the two fluid fields become independent. A numerical approach is developed to estimate at the same time the air-freshwater free surface and the fresh-saltwater positions in 3-D space. The technique is again based on a Galerkin finite element scheme and a generalized form of the FUP technique developed previously for a transient, variable, and unsaturated flow. The generalized FUP accounts for the dual free boundaries separating the freshwater flow from the unsaturated zone and the saltwater respectively. In contrast to water table aquifers only, care is taken to include density effects in the formulation.

A thorough analysis of this numerical formulation, and assumptions in this approach are presented first. In particular, the choice of a sharp interface approach is discussed and justified. Afterwards, several validation and application examples are shown to establish confidence in the obtained numerical results. These test problems include a number of analytical solutions which have been chosen carefully, such that the limitations and applicability of the numerical solution technique will be highlighted and explained. An important test case is a three-dimensional laboratory model (Sugio and Rahim, 1992) which demonstrates the usefulness of the newly developed numerical procedure, enabling to accurately predict positions of fresh-saltwater interface and free surface in complicate and irregular configurations.

The developed three-dimensional groundwater and saltwater intrusion model 'GEO-SWIM', is applied to the coastal aquifer of Martil in Morocco, as a validation of the model package, and also as an example showing the integration of GIS support to prepare a basic framework for the model application. Initial conditions and some model unknown parameters of the aquifer are found using an aerial and error calibration procedure. This study enables to understand the
aquifer response to changes in recharge and total rate of pumped water, and their affects on seawater intrusion. Different scenarios are investigated for the period of 1966 to 2006, to predict future situations and the salinization risk from seawater intrusion. The obtained results show that the interface will move fast and travel over considerable distances in forthcoming years, and will produce an irreversible degradation of the groundwater quality, especially along the coast and in the center of the Martil plain. An alarming optimal management scheme in the near future is necessary for its safeguard.

5.2y Conceptually Model y

In this model, distinction is made of three main areas, namely the unsaturated, the saturated freshwater, and the saltwater zones respectively as shown in Fig. 5.1. The flow hydrodynamics behave differently in each area. In this conceptual problem, only freshwater flow is taken into account, but without excluding completely the other zones from the simulation or the system of FE equations. The unsaturated zone is treated as explained in chapter 4, while the saltwater zone is transformed to an equivalent freshwater zone having the same pressure head distribution as that of the saltwater.

As introduced in chapter 3 this procedure avoids the simultaneous solution of a coupled system of governing differential equations for fresh and salt water zones, or ideally the flow and an mass transport equations. The principle consists on dividing the FE domain to three groups of nodes as depicted in Fig. 5.1A

1. Unsaturated nodes where the pressure is negative, such that

\[ h(x, t) < z \]  \hspace{1cm} (5.1A)

2. Saturated fresh water nodes where the pressure should be larger than the pressure in the saltwater zone, assumed to be in hydrostatic equilibrium

\[ h(x, t) > z \]  \hspace{1cm} (5.1B)
Figure 5.1: Potential pressure conditions along the free boundaries; in the unsaturated, fresh and salt water zones.

with the sea. These nodes satisfy therefore the following equations

\[ h(x, t) \geq z \quad (5.2a) \]
\[ h(x, t) \geq -\delta z \quad (5.2b) \]

3. Saltwater nodes which necessarily satisfy the condition

\[ h(x, t) < -\delta z = h_s \quad (5.3) \]

where \( h_s \) [L] is the equivalent saltwater potential at the interface position.

This procedure is implemented on an iterative basis, meaning that the positions of the free and moving boundaries are iteratively changing inside the initial fixed mesh domain, but are not implemented as boundary conditions of the dual problem, and thus avoiding a computational difficulty. Hence, the three different zones are changed accordingly.
This mode permits change of the interfaces in response to boundary conditions applied in the freshwater moving domain. Care should be taken for specification of the boundary condition nodes, which should belong exclusively to the initial and final freshwater domains. Otherwise, problems may occur during model execution. This limitation is only applicable to the saltwater zone, because specifying other conditions than the sea outflow face in the salt domain, is not physically acceptable under the sharp interface approximation, and in the specifically designed numerical approach to be explained further in the upcoming section.

A clear example is that of an interface alone below a pumping well near the coast, where the apex of the upcoming beneath the well crosses the lower well filter section. The only way around this at present is to ensure that the pumping well does not extracts out large amounts of saltwater from the aquifer. Research efforts need to be pursued for developing better and advanced numerical techniques to better include these complicated conditions, which would be feasible in the context of our model. But, in the framework of this study, we report much of the developments towards a numerically stable and a mass-conservative solution by extending the scope and feasibility of the FUP Approach as discussed earlier.

5.3.4 TheyGeneralizedFUPyApproachy

The solution of the FE system for this class of problems is solved in a similar fashion as presented for the variable unsaturated flow equations. However, the numerical techniques needs further modifications and tweaking to comply with physical conditions involved in the saltwater zone. Hence, we take advantage of the FUP procedure which was developed for the case of the free air-freshwater interface, to extend it in cases where dual free boundaries exists in the domain.
5.3.1 Relative Hydraulic Conductivity

The relative hydraulic conductivity is updated depending on the relative position of the nodes versus both the water table and the saltwater interface iterative positions. The updating process of the relative hydraulic conductivity coefficients, $k_{ij}$, is therefore generalized as:

$$k_{ij} = \begin{cases} 
1 & \text{if } p_i \neq 0 \text{ or } p_j \geq 0 \text{ and } A_i \text{ and } A_j \text{ are apart from the water table} \\
\varepsilon & \text{otherwise}
\end{cases}$$

(5.4)

$$k_{ij} = \frac{\eta _{i}k(\eta _{i}) - \eta _{j}k(\eta _{j})}{\eta _{i} - \eta _{j}}$$

if $A_i$ and $A_j$ are apart from the saltwater interface (5.5)

where $\eta_i$ is the distance of node $A_i$ from the saltwater interface, and given by:

$$\eta _{i} = \delta z _{i} + \delta x _{i}$$

(5.6)

$k(\eta)$ is a relative hydraulic conductivity function, which is defined as:

$$k(\eta) = \begin{cases} 
1 & \text{if } \eta = 0 \\
1 + \varepsilon & \text{if } \eta > 0 \\
\varepsilon & \text{otherwise}
\end{cases}$$

(5.7)

5.3.2 Water Retention Curve y Density Dependence

The nonlinear storage in time from the right hand side of Equation 3.44 is evaluated numerically in the FUP numerical technique. The numerical procedure is similar to that developed in the previous chapter, except that the storage variations due to changes of the saltwater zone displacement should be included. These changes are evaluated to be equal to the saturated water content, $\theta_s$. However, to ensure numerical stability the variation should be smooth across the nodes around the salt-freshwater interface. This has been performed by modifying the functions in Figure A.4 representing the idealized water retention curve, and the...
water capacity term at the first nonlinear iterate, to the functions represented in  
Fig. A5.2, which may be expressed as follows:

\[
\theta(p) = \begin{cases} 
\theta_r & \text{if } p_i < -\frac{d_i}{2A}, \\
\frac{\theta_s - \theta_r}{d_i} p_i + \frac{\theta_s + \theta_r}{2} & \text{if } \left| p_i \right| \leq +\frac{d_i}{2A}, \\
\frac{\theta_s}{\delta d_i} p_i + \frac{\theta_s}{2A} \left(1 + \frac{A z_i}{d_i / 2A} \right) & \text{if } \left| p_i - \delta z_i \right| \leq +\frac{\delta d_i}{2A}, \\
0A & \text{if } p_i > -\delta z_i + \frac{\delta d_i}{2A}.
\end{cases}
\]  

(5.8)

and

\[
\frac{d\theta}{dh}(p) = \begin{cases} 
\frac{\theta_s - \theta_r}{d_i} & \text{if } \left| p_i \right| \leq +\frac{d_i}{2A}, \\
\frac{\theta_s}{\delta d_i} & \text{if } \left| p_i - \delta z_i \right| \leq +\frac{\delta d_i}{2A}, \\
0A & \text{otherwise}.
\end{cases}
\]  

(5.9)

The newly distinguished pressure distribution around the fresh-saltwater interface is smaller in size in comparison with the equivalent pressure distribution existing around the water table, because small variations in the water table position involves greater displacement of the fresh-saltwater interface. And hence, much larger variations in the storage term. The coefficient of proportionality is taken equal to $\delta$, according to the Ghyben-Herzberg (GH) relationship.

### 5.3.3y Numerically Solution Procedure

Here again, the same numerical procedures developed in the previous chapter are still applicable. In this case, except that changes of the saltwater Alomain should be accounted for the density variations. Hence, the generalized AFUPA approach is built around the same nonlinear Picard solver in time, and the inner preconditioned PCG solver as discussed earlier.
Figure 5.2: A generalized (a) water retention curve, and (b) analytic differentiation of the slope tangents at the nonlinear first iterate.

A great feature in this conceptual generalization phase is the straightforward, modular, implementation using existing software components developed previously.

5.3.4 Other Features of the Proposed Approach

Though the model is based on an iteratively adjusted position of the fresh-saltwater interface, based on the Ghysben-Herzberg relation known to be very limiting. Approximations and inaccuracies related to this 1-D approach are re-solved in the scope of the presented numerical formulation. In particular, the
geometry of the sea outflow face window boundary condition, to be determined in relation with the freshwater flow behavior in the aquifer system, is automatically adjusted (see Chapter A for details). The interface position near the sea shore will not be over-estimated as would be the case within the GH approximation.

Situations involving two moving interfaces in unconfined flow are solved efficiently, these solutions are cheap and run within a minimum hardware requirements. The model has the ability of simultaneous determination of the interface and freshwater potential heads distributions at different time levels. The obtained potentials are continuous across the interfaces, but only potentials comprised between the water table and the freshwater-saltwater interface are significant. Hence, a verification model based on measured groundwater heads is still feasible even without taking into account salinity measurements for validating the interface position simultaneously, since piezometric measurements are affected by the axis-tenence of the interface. Because, this as a practical limiting factor in many projects in coastal aquifers, the developed model will prove to be useful for representing the global response of the saltwater interface in relationship with the changing freshwater heads conditions.

It will be shown through the field application presented in Section 5.5, that this model as a useful tool for calibration of groundwater heads. Moreover, the model could be used as a practical tool for providing routinely management support at a professional level, e.g. in a groundwater management office or a governmental department, because of low requirements in data input as well.

### 5.4 Validation and Application Examples

In this section, examples are discussed and used as a basis for comparison with the model predictions. The developed model as validated for several situations, involving confined and unconfined aquifer systems, under a variety of physical...
boundary conditions such as recharge and groundwater abstraction. In the majority of these examples, the problem accommodates an available analytical solution. Most of these solutions are limited to steady state conditions. It is difficult to find a solution for transient interface flow, and the existing ones lack general applicability. Since this is not always the case for a real case study, a validation is made with laboratory experiments achieved by Sugio and Rahim (1992) in an irregular box allowing for a three-dimensional shape of the interface, and variable conditions.

Analytical solutions involving a saltwater interface in confined aquifers are relatively more encountered, especially if the saltwater is at rest. This is because under such conditions, exact potential functions can be derived separately for each region. Strack (1976) has contributed a continuous potential function across the interface either for confined flow, or unconfined flow as well. But, many other solutions were also given by Glover (1959), Van der Veer (1977a,b), Van Dam (1982), Haitjema (1991), and Bakker (1998).

### 5.4.1 Seawater Intrusion in a Confined Aquifer

This example concerns steady state seawater intrusion under natural conditions (i.e., no recharge and/or pumping conditions exists) in a cross-section of a rectangular confined aquifer having a uniform inland horizontal recharge flux, $q$, as depicted in Fig. A5.3. This problem as seen from Aaken from Larabi and De Smedt (1997) who used the same problem for validation of the old computer code version. It is considered here again, to demonstrate the improvement directly obtained from the use of the newly implemented solver. Table 5.1 summarizes the problem's physical and computational parameters which were used to obtain the numerical solution.
Table 5.1: A Glover’s sytem problemphysicallyandycomputationaly model parameter values.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upstream Aixed Alux, q</td>
<td>3.9 cm²/s A</td>
</tr>
<tr>
<td>Rectangle Anconfined Aquifer Adimensions, L, D</td>
<td>400 cm, 27 cm A</td>
</tr>
<tr>
<td>Grid Atensions A</td>
<td>106x2x10 A</td>
</tr>
<tr>
<td>Grid Apacings, Δx, Δy, Δz</td>
<td>4 cm, 4 cm, 3 cm A</td>
</tr>
<tr>
<td>Density Aifference Aatio, δ</td>
<td>0.029 A</td>
</tr>
<tr>
<td>Saturated Adraulic A conductivity, Kₙ</td>
<td>69 cm/s A</td>
</tr>
<tr>
<td>Start Ander relaxation Aector, ω⁰⁴</td>
<td>0.25 A</td>
</tr>
<tr>
<td>Water Aolerance Aarameter, tolw</td>
<td>0.02 cm A</td>
</tr>
</tbody>
</table>

Comparison between the two Aolutions is shown in Fig. 5.4 where a very Aood A agreement is obtained. ALike in most Aquirer systems, the vertical dimension aspect ratio is increased to enable a better view of the Aresults. However, for coastal Aquifers especially, distortion of the figure does not illustrate the orthogonality between the groundwater potential isolines and the saltwater interface. Therefore, for the actual problem a zoom is performed on the intrusion zone, while keeping the same scale, an horizontal and Avertical Adimensions (Fig. A.5). A Groundwater Aheads are also plotted showing clearly the hydraulic Aradient increase seaward.

Another Abject of Aterest is the Aessment of the Aolver A robustness and Aiciency. ASince, steady Aate Aolutions Are calculated, a Aingle
Figure 5.4: Comparison of Glover's analytical solution and numerical results.

Figure 5.5: Zoom window showing the fresh-saltwater interface position and the groundwater heads (X and Z axis have the same scale).

sequence of nonlinear Picard iterations is involved. Fig. 5.6 plots the convergence history of such process, only 7 nonlinear iterations were needed to achieve convergence, with each step acquiring an average of 7 PCGA iterations. The mass balance error acting as an indicator for the quality of the computed results is equally excellent, 0.135x10^{-2}.

5.4.2y Seawater Intrusion in Any Unconfined Aquifer

Problem Description

Van der Veer (1977) has proposed an analytical solution for the steady interface flow in a coastal aquifer systems involving a phreatic surface. This approach is a two-dimensional, and assumes the existence of a distance $L_0$ between the point where the interface and the phreatic surface reach the sea level as shown in FigA 5.7.
Analytical Expressions

The analytical solution is derived based on a non-linear algebraic expression in $A$ the complex potential $\Omega$, defined by $\Omega = \Phi + i\Psi$, where $\Phi$ and $\Psi$ are respectively the velocity potential and the stream functions. Boundary conditions must hold at the interface and the phreatic surface, which gives respectively the position of the interface $\eta$ and the phreatic surface $h$ as:

$$
\eta(x) = A \left[ \frac{-(N x^2 + 2q^*)^2}{(\delta + \frac{N}{K})(\delta + 1)A} \right]^{1/24}
$$

(5.10)

$$
h(x) = A \left[ -(N x^2 + 2q^*)^2 - \left( \frac{q^*}{K} \right)^{24} \frac{(1 - [\delta + \frac{N}{K}])A}{(\delta + 1)(1 - \frac{N}{K})A} \right]^{1/24}
$$

(5.11)

where $A\nu$ is the effective precipitation, $K$ is the saturated hydraulic conductivity of the homogeneous aquifer, and $A$ is a density gradient ratio. The $A$ axis origin is taken at the point where the interface reaches the sea level, and $A\eta^*$ is a flux
quantity $q^* \equiv \dot{Q} + N l_e$ \hspace{1cm} (5.12)

where $\dot{Q}$ is the outflow of fresh water towards the sea. Note that $\dot{Q}$ is calculated from the boundary condition $\dot{Q}(-l_e) = 0$, which leads to the following expressions:

$$l_e = \begin{cases} \frac{q^*}{2K} (A^2 - 1) & \text{if } Av = 0 \ A \\ \frac{q^*}{N} \left[ 1 - \left( 1 - (\frac{N}{K})^2 \left( 1 - \frac{N}{K} \right) \right) \right]^{1/2} & \text{if } Av \neq 0 \ A \end{cases}$$

(5.13)

There remains only one undetermined variable, $q^*$. Two different situations are discussed:

**Symmetrical Flow in an Island** In case of symmetrical flow, we denote $l$ as the distance separating the two points where the interface intersects the sea level at each side, the outflow in steady state is given as $\dot{Q} = NV(l_t - l_e)$ and it results that $q^* = NV l_t$.
Non-Symmetrical Flow  Another case which can be encountered is a situation where a fixed head value is attributed at the upstream boundary where \( \eta \) is unknown, see Fig. 5.7 for details. In this situation the quantity \( \eta^* \) is calculated analytically from the fixed potential \( \eta_i \) as

\[
q^* = \eta_i K_i - \left[ a_1 (a_{14} - \frac{N}{K}) (K_i t)^{24} - a_2 (K h_i)^2 \right]^{1/2}
\]  

(5.14)

where

\[
a_{04} = (1 - \frac{N}{K})[1 - (\delta + \frac{N}{K})], \quad a_{14} = \eta_0 (\delta + \frac{N}{K}), \quad \text{and} a_{24} = \eta_0 (1 + \delta)
\]

Comparison with Numerical Results and Discussion

Two test problems were run to check the model accuracy by comparing the analytical solution and the numerical results. The first example corresponds to the case \( N=0 \) i.e. no precipitation. The fixed finite element mesh used in this example is composed of 211x2x51 nodes respectively along x, y and z coordinate system, which leads to a mesh size of 1522 nodes and 4050 elements. In the FEA mesh the portion above the sea reference is refined to allow for the calculation of the phreatic surface with greater precision. The mesh dimension is 10m and a 5m respectively along x and y-directions, while it varies from 1m above the m.s.l. to 0m below. All nodes at starting from a position \( x=1968 \text{m} \) are considered as isolated nodes in the model, because \( \eta_e \) is found to be equal to \( 32 \text{m} \) from Equation 5.13 with \( \eta_i=2000 \text{m} \). Additional physical parameters of this problem are \( K=10 \text{m/d}, \delta=0.002 \) and \( h_i(x=0)=10 \text{m} \).

As boundary conditions, an outflow sea boundary condition is attributed to all nodes starting from \( x=1968 \text{m} \) in the plane \( z=0 \text{m} \), and a fixed head boundary condition \( h=10 \text{m} \) is attributed to all nodes in the plane \( x=0 \text{m} \). For the second test problem \( N=1440 \text{mm/year} \) which impose another fixed flux boundary condition of \( 0.004 \text{m/d} \) at the upper plane \( z=10 \text{m} \) and \( x \leq 1968 \text{m} \). Fig. 5.8 shows simultaneously the obtained numerical results for test runs 1 and 2. The steady fresh-saltwater interface and the phreatic surface positions are compared with
the analytical solution, and these comparisons show a good agreement between the two solutions and the model ability to handle situations involving recharge conditions. Fig. 5.9 and 5.10 show the freshwater heads and interface response to the recharge boundary condition, the interface moves seaward due to more steep hydraulic gradients near the shoreline. The location of the point where the interface reaches the sea level has been determined 'numerically' very close to $x=2000\,\text{m}$ for the first problem as expected.

Here again, it is instructive to show how the model solver takes control of the equations system. The nonlinear iterative process for runs 1 and 2 is plotted in Fig. A5.11. An average of 214, and 223 PCG iterations were needed for convergence, the two tests used an average of 214, and 223 PCG iterations respectively per nonlinear iterative step. Recorded mass balance errors were satisfying, the global solution accuracy, they are equal to $0.145\times10^{-1}\%$ and $0.733\times10^{-2}\%$. 

Figure 5.8: Comparison between numerical and Van der Veer's analytical solution.
Figure 5.9: Freshwater potential heads distribution for the first run.

Figure 5.10: Freshwater potential heads distribution for the second run.

respectively.

5.4.3 Plan Seawater Intrusion Control with Artificial Recharge
Problem Description

Seawater intrusion control with pumping or recharge wells is an important issue in field methods and practices for site remediation in coastal aquifers. In particular, artificial injection has been practiced in the field in several projects, with the aim to push the fresh-saltwater interface toward to sea. Two strategies exists: (i) positioning a pumping well field near the coast to withdraw saltwater; or (ii) a battery of recharge wells inland injecting a fixed amount of freshwater, in the later case the injected freshwater is usually of lower quality than that stored in the aquifer. In such situation delineation of the backward movement of the injected water is also important.

In this test example, we consider several wells parallel to the coast line at a fixed distance, \( d \), as represented in Fig 5.12, which are injecting an equal amount
of water, $Q_w$, and where $A$ is the distance separating two successive wells. The inland uniform flow rate per unit arc length normal to the coast is denoted by $A$.

**Analytical Solution**

A steady state analytical solution for this problem is derived by Hunt (1985). This solution accommodates for the case of (i) one inland recharge well; and (ii) an infinite number of recharge wells. Herein, the second solution was attained, because the first solution implies boundaries at infinity, which can not be handled in the numerical model. Hence, the second analytical solution was derived based on a modified form of the Strack’s (1976) potential. For an unconfined flow, an analytical approximation was used to locate the interface.
curve for an unconfined aquifer, which has the following equation:

\[
\frac{K D^{24}}{2} \delta (1 + \delta) = \mathbf{A} Q + \frac{Q}{2\pi} \ln(r) A
\]  

(5.15)

where \(D\) is the vertical distance between the mean sea level and the bottom of the unconfined aquifer, and \(A\) is given by:

\[
\rho^{24} = \frac{\sinh^{24} \pi \left( \frac{x + d}{l} \right) \cos^{24} \pi \left( \frac{y}{l} \right) + \sin^{24} \pi \left( \frac{y}{l} \right) \cosh^{24} \pi \left( \frac{x + d}{l} \right)}{\sinh^{24} \pi \left( \frac{x - d}{l} \right) \cos^{24} \pi \left( \frac{y}{l} \right) + \sin^{24} \pi \left( \frac{y}{l} \right) \cosh^{24} \pi \left( \frac{x - d}{l} \right)}
\]  

(5.16)

Equation (5.16) has extrema located at \(y = 0\) and \(y = \frac{l}{2A}\). For a fixed \(y\) value, Equation (5.15) can be solved by a numerical procedure such as Muller's method (See Appendix B) for finding the root of a real function with a single variable.
Comparison with Numerical Results and Discussion

The physical and computational data set used for this test problem are given in Table 5.2. Because of the problem symmetry we consider only the half plane \([0, \frac{1}{2A}]\) in the \(y\)-direction separating two wells, thus only one well as virtually. A study of the current modeling study. A Notice also that the elements around the element sizes are uniform, \(\Delta x = \Delta y = 10m\), and \(\Delta z = 4m\), except for the first element layer above the water table, \(\Delta z = 10m\), because in this example a detailed water table profile is not the main interest. It is assumed also that the recharge occurs for nodes lying below the sea level, and that the upstream uniform recharge should always adapt to the water table increase/decrease.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upstream Aided flux, (q)</td>
<td>3m²/dA</td>
</tr>
<tr>
<td>Rectangle confined aquifer size, (L, D)</td>
<td>410m, 40mA</td>
</tr>
<tr>
<td>Grid dimensions (A)</td>
<td>4111112A</td>
</tr>
<tr>
<td>Density difference ratio, (\delta)</td>
<td>0.025A</td>
</tr>
<tr>
<td>Saturated hydraulic conductivity, (K_s)</td>
<td>10m/dA</td>
</tr>
<tr>
<td>Well injection rate, (Q_w)</td>
<td>1000m³/dA</td>
</tr>
<tr>
<td>Distance from wells from the coast, (d)</td>
<td>100mA</td>
</tr>
<tr>
<td>Distance between two wells, (l)</td>
<td>200mA</td>
</tr>
<tr>
<td>Start Appler under relaxation factor, (\omega^{04})</td>
<td>0.25A</td>
</tr>
<tr>
<td>Water tolerance parameter, (\tau_{bw})</td>
<td>0.01mA</td>
</tr>
</tbody>
</table>

The maximum seawater intrusion at the bottom of the aquifer is plotted against the analytical solution, and shows a good agreement. Although differences exist, the numerical solution is acceptable. Regarding, first (i), the coarse mesh resolution used; and secondly (ii) the upstream Neuman boundary condition which is applied at a given limit, 400m inland, while it exists 'theoretically' at the infinity, so as a consequence the computed interface has a tendency to intrude more inland as concluded.
Figure 5.13: Comparison between analytical and numerical results for the hunt’s test problem at the aquifer bottom. Results are plotted together with the countour lines of potential heads.

Figure 5.14: Cross-section view of the computed interface and freshwater potential heads along the injection well.

The most seaward interface profile at the plane y=0 is plotted in Fig. 5.14, where the artificial injection through the well location is clearly shown by the potential contour lines. A better 3-D enlarged view of the simulated interface is showed in Fig. 5.15 where intersecting potential iso-surfaces are also plotted, it is noticed that the interface line at the aquifer bottom coincides with the potential isoline h=1m. Convergence within the iterative procedure was achieved within 4 outer iterations, requiring an average of 48 PCG iterations, and performing a mass balance error of $0.176 \times 10^{-2}\%$. 

Figure 5.15: A zoom view on the interface, with potential isolines shown along with the outflow to the sea face.

5.4.4y Saltwatery Intrusion in Multilayer Aquifer System

This example is taken from Huyakorn et al. (1996) who solved the same theoretical problem with their SIMLAS code. The test problem involves the simulation of staggered fresh-saltwater interfaces in a multilayer aquifer system. Figure 5.16A shows the conceptual problem, in which the chosen parameter values are given, as the hydraulic conductivities of the upper aquifer, the intermediate thin leaky layer, and the lower confined aquifer, respectively. A Comparison was conducted against the steady state analytical solution derived by Mualem and Bear (1974). It is however important to point out that this solution was based on a linearized form of the governing equation, and the use of the Dupuit-Forchheimer assumption. Hence, the analytical solution is not exact, and this was given as an argument by the authors to explain the SIMLAS results deviations from the analytical results as shown in Figure 5.17A.

We did run the same problem with the data already given in Figure 5.16, and
using a uniform orthogonal mesh composed of 49x2x21 nodes. Convergence was achieved within 3 nonlinear iterations, and an average of 53 PCG iterations per each outer iterate. Comparison between analytical and numerical results is shown in Fig 5.18. The predictions are satisfactory regarding the approximations used in the numerical solution, also comparisons could not be performed for the lower aquifer unit, outside of the leaky layer extension, because the boundary conditions for the analytical solution are specified at the aquitard boundaries. Interestingly, GEO-SWIM (Sbai and De Smedt, 1999; Sbai et al. 1998) and SIMLAS results are very close, a comparative study of the two simulators based on other examples, will certainly be fruitful for a wide range of practical applications.
Figure 5.18: A comparison of analytical and numerical solution with the GEO-SWIM code, for the fourth test problem.

5.4.5 Moving Saltwater Interface in a 3-D Laboratory Sandy Box Model

To demonstrate that the numerical model is able to accurately simulate 3D groundwater flow problems with a moving freshwater-saltwater interface, a computation is made with results from a 3-D sand model (Sugio and Rahim, 1992). The model consists of a 3D box, 165.8 by 17.5 cm, and having a width of 63.2 cm in the first 82.3 cm section, and 40 cm in the remaining part. During the experiment, acrylic plates were placed at upstream and downstream sections, while the aqueous saltwater solution was colored with dye to distinguish it from the freshwater part.

The upstream and downstream water levels are 44.15 and 40.67 cm respectively; other data are K=1.293 cm/s and α = 0.03. The behavior of the saltwater interface was measured at the front, back, and bottom sides of the box for a steady-state (initial conditions), and each 30 min thereafter, when the upstream head is reduced to 42.65 cm with a linear decrease rate of 0.01 cm/min.

The model is applied by discretizing the box into a total number of 154275 elements, and 80800 active elements, which are included in the simulation, as shown in the finite element mesh in Fig. 5.20. The positions of the saltwater
5.4 Validation and Application Examples

Figure 5.19: A descriptive view of the experimental sandbox model used by Sugio and Rahim (1992).

The interface is plotted for each section of the model and compared with the numerical results in Fig. 5.21 a, b, and c). These steady and transient interfaces show that the predicted 3D results compare well with the measurements. The interface has a tendency to advance rapidly in the back section than in the front section, which is because the well has pronounced three-dimensional behavior of the flow created by the shorter width section, in which the flow is faster, and the intrusion is likely to be more important. This is a better understood from the plotted maximum interface profiles (or interfaces to positions) shown in the bottom section. It is also worth noting that small variations of the upstream groundwater potentials had led to a noticeable tracking of the salt-freshwater interface; a situation often encountered in practice.
Figure 5.20: Finite element mesh used in Sugio’s laboratory sand box model validation.

5.5 Model Application to Seawater Intrusion in Martil Aquifer

5.5.1 General Situation and Background

Aquifer systems in northwestern part of Morocco are known to be of small extension. The Martil aquifer (Fig. 5.22) is not an exception with its 80km² surface area (Fig. 5.23). However, it is one of the important local groundwater resources, especially for water supply of Tetouan city, and its industry and irrigated areas located in the center of the plain. In recent years, the aquifer has become vulnerable to potential pollution due to leachate of domestic and industrial wastewater in the Martil river, and also due to seawater intrusion from the Mediterranean sea.

Relevant hydrogeological information on this aquifer is scarce as for many other sites in Morocco, such that before a representative model of this aquifer
can be set up, a significant fraction of the work involves aquifer characterization and reinterpretation of previous measurements. A Combined Use of Geographic Information Systems tools (GIS), and developed software interfaces (chapter 7), give consistent support to correlate unavailable data, and a robust approach for its interpretation. A total of 59 boreholes were selected to make digital elevation maps of contacts between different geological layers.

A three-dimensional finite element model for the Martil aquifer is developed using the GEO-SWIM package and associated graphical interfaces. First, a steady-state groundwater flow was simulated. A Calibration of this model will A
establish natural conditions and hydraulic conductivity ranges for the different aquifers. Afterwards a transient simulation is performed to predict future lateral extension of the saltwater encroachment due to pumping of groundwater. Future trends of the salinization risk from saltwater intrusion are investigated. This is the first time that a simulation model for groundwater flow and seawater intrusion in the Martil aquifer system is performed.

### 5.5.2 Data Analysis

Previous studies carried out in the plain (Ennouhi and Melouki, 1984; El Morabit et Pulido-Bosch, 1993; Larabi et al., 1998) identified two aquifer units, sepa-
Figure 5.23: Study area and locations of cross-sections of interest.

The upper aquifer is formed from Quaternary alluvial deposits of the Martil river; the lower aquifer unit is composed from sandstone-limestone-Pliocene formations, while the aquitard is mainly an Arenaceous clay. Variations in thickness are shown to be significant from North to South, and also along the West-East direction. However, this information was not taken into account in studies made in the past. To better characterize the thickness of the three hydrogeological units a reinterpretation of data obtained from a total of 59 wells and boreholes is performed with support of GIS tools.

The aim of this study is to better define the structure of the aquifer system (upper, lower aquifers and aquitard) and their extension. Digital elevation models of these surfaces are shown in Fig. 5.24, and are obtained by interpolating a scatter point data sets obtained from contacts between different formations or at the
Figure 5.24: Interpolated contour maps of (A) aquifer topography, and bottom of (B) phreatic aquifer, (C) aquitard and (D) confined aquifer respectively.

surface (topography) of each borehole into a numerical grid (usually the same as used for the numerical FE model). A summary of statistical analysis results of thickness of each of the three units is presented in Table 5.3.

5.5.3 Construction of the Conceptual Model

Model Set Up

The conceptual model used in this study is a multi-layer aquifer system composed of two aquifer strata, separated by an aquitard layer of variable thickness. A structured surface mesh of 93 columns and 121 rows is used to approximate the
Table 5.3: Information on thickness of model layers.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Minimum Thickness (m)</th>
<th>Maximum Thickness (m)</th>
<th>Mean Value (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Aquifer</td>
<td>8.9</td>
<td>24</td>
<td>16</td>
</tr>
<tr>
<td>Aquitard</td>
<td>0.0</td>
<td>28.8</td>
<td>11.5</td>
</tr>
<tr>
<td>Lower Aquifer</td>
<td>0.0</td>
<td>36.4</td>
<td>17.3</td>
</tr>
</tbody>
</table>

aquifer domain, and some sub-zones are set to be inactive to fit the remaining part at the domain boundaries, especially at the weastern boundary near Tetouan city. In total, the aquifer system is divided into 9 nodal layers, with 88320 hexahedral elements and 101277 nodes. The two-dimensional mesh is projected over the digital elevation maps, including the topography map, as shown in Fig. 5.24. Nodal elevations in intermediate layers are found by linear interpolation from values of existing surfaces above and below. This three-dimensional mesh
conforms to the aquifer configuration and makes it easy to fill in soil types of A elements. Aside, each of the main three layers. The three model units are assumed to be homogeneous and isotropic in this study, to allow for a faster and efficient A calibration.

Fig. A5.25 shows the finite element mesh used for the numerical simulation of the aquifer system which is adjusted to fit the structure and the extension of the hydrogeological layers (cross-sections AB-B’ and F-F’). The locations of these cross-sections are indicated in Fig. 5.23. Cross-section B-B’ is directed S-NA while cross-section F-F’ is directed E-W.

**Boundary Conditions**

Domain boundaries are set impervious, except at the eastern part, which is in direct contact with the sea. This boundary receives the 'special' sea outflow face boundary condition as described in a chapter. The extent of the fresh water outflow to the sea is therefore automatically determined as part of the results. Nodes along the rivers paths are taken as fixed ahead with the assumption that water levels are equal to the elevation. Another condition taken into account is the effective rainfall, which is assumed to be uniformly distributed over the whole surface. Saturated hydraulic conductivity values are deduced from a calculated transmissivities of pumping tests analysis, conducted by the Regional Department of Tetouan. These values are characterized by a high variability, and range between 2.3 m/d and 4.0 m/d for the unconfined aquifer; and 3.5 m/d and 4.8 m/d for the confined aquifer. Only average layer thickness values were taken into account as a basis for this estimation, which explain why the obtained A values should be calibrated afterwards.
Figure 5.26: Comparison between (a) computed and (b) observed (in 1966, before heave pumping) steady state groundwater potentials (presented as meters above sea level).
5.5.4y Model Application and Results

Natural Groundwater Flow and Seawater Intrusion

The first set of steady state simulations is performed under natural conditions (i.e., no pumping at wells). The objective of this calibration process is to reproduce a natural groundwater flow pattern of the aquifer system and at the same time provide a range of confidence limits for the model parameters, such as hydraulic conductivity and effective recharge. The obtained results for each test run were compared to piezometric levels measured in 1966. It can be assumed that the aquifer was not yet heavily pumped at that time. A trial and error calibration procedure is used to estimate the hydraulic conductivity of the different layers and the natural recharge. The tests show that the conceptual model is more sensitive to changes in the effective recharge value, moreover, varying hydraulic conductivity values inside a given range for the same recharge does not produce a drastic increase or decrease of the water table.

Computed groundwater potential heads versus observed values are compared in Fig. A5.26, the fit is satisfactory except at the center of the plain and near the coastline where differences are obvious. There might be two reasons for this. First, the observed values were obtained during an aperiod, in which the exploitation of the aquifer already started, in a form of pumping for domestic use and irrigation in the center of the plain. Secondly, the differences may be explained also by the fact that soil heterogeneity and anisotropy were not included in the calibration procedure. However, in general, the model is able to reproduce the same flow pattern, indicating that the main groundwater flow is directed W-E, with some convergence tendency to the rivers.

In the GEO-SWIM code (Sbai and De Smedt, 1998) the fresh-salt water interface is computed iteratively in parallel with the groundwater potential heads, and the simulated interface is obtained after convergence. An enlarged 3D view of the simulated interface is depicted in Fig. 5.27, showing that preferential paths...
for seawater intrusion are located along the rivers, especially along the Martil river where the maximum intrusion equals 1100m in the lower aquifer. W-E cross-sections displayed in Fig. 5.28 show the shape of the steady interface and water heads distribution in different parts of the aquifer. By comparing cross-sections C-C’ and E-E’ which are respectively parallel to the Alila and Martil rivers, it follows that saltwater intrusion is more sensitive along the Martil river as the hydraulic gradient is the smallest due to the flat topography.

**Calibration Results**

The parameters obtained after calibration are summarized in Table 5.4
Figure 5.28: W-E cross-sections showing the initial groundwater heads and fresh-saltwater interface position. (The vertical scale magnification factor equals 100 for cross-sections C-C’ and E-E’, and 66.7 for F-F’)

Future Prediction Scenario for Seawater Intrusion

A second set of simulations is performed, to predict the present and future groundwater flow and seawater intrusion in the aquifer. A long-term transient simulation is made of 40 years starting in 1966. The previous calculated heads are used as initial head values.

The pumping of groundwater is assumed to decrease the natural recharge by half during this period, because no other precise information is available about the number of wells and their pumping rates. Fig. 5.29 shows the computed
Figure 5.29: Computed moving toe positions of the sharp freshwater saltwater interface each 8 years from 1974 to 2006. The shaded surface represents the bottom of the lower aquifer.

fresh/saltwater interface in the lower aquifer each 8 years from 1974 until 2006. Corresponding positions of the moving interface are also plotted along cross-section F-F’ as shown in Fig. 5.30.

Upcoming effects due to pumping are negligible, since pumping rates are not concentrated in specific locations as it would be in practice. Also here, the maximum seawater intrusion occurs at the Martil river as clearly shown in Fig. 5.31, where the lateral distance to the coast from the maximum interface toe position versus time is plotted. Three different time periods can be distinguished: before 1986, the interface moves inward linear in time; a second period between 1986 and 1992, when the interface is intruding quadratically in time; and finally after 1992 where the movement is exponential in time.
Table 5.4: Model calibrated parameters under steady state conditions.

<table>
<thead>
<tr>
<th>$K_s$ (M/D)</th>
<th>$\theta_s$</th>
<th>Density Ratio</th>
<th>Effective Recharge (mm/y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>II</td>
<td>III</td>
<td>I</td>
</tr>
<tr>
<td>3.5</td>
<td>0.05</td>
<td>4.5</td>
<td>0.4</td>
</tr>
</tbody>
</table>

1: Upper aquifer unit; II: Aquitard unit; III: Lower aquifer unit

cross-section F-F'

Figure 5.30: Cross-sectional view of the seawater intrusion; simulated moving interface positions from 1974 to 2006 are plotted.

5.6 Summary

A finite element model for simulating seawater intrusion in fully three-dimensional groundwater aquifer systems is developed. This accounts for free and moving boundaries, either between the saturated and unsaturated domains (air-freshwater interface) or between the freshwater and saltwater domains (fresh-saltwater interface). The key assumptions, is an iteratively based Ghyben-Herzberg approximation of the interface position, which permits the flow field to be completely dependent on the governing equations for the freshwater phase. Computer memory and storage savings are not only due to such approach. The presented generalized FUP approach is the second major step for making such economy. The FUP had to adapt for the density difference between the salt and fresh water zones. This was explicitly reflected in the shape of the idealized water retention curve, and in the technique for regenerating the relative hydraulic conductivity at each inner time step. The previously constructed flow solver was smoothly
implemented in the new model, without further costs, which is a proof for its strength and robustness.

A comprehensive set of 5 test problems is provided for validating the numerical model. These examples account for various analytical solutions, numerical results from other models, and measurements performed on a laboratory sand box model. Moreover, several types of aquifers were investigated, including confined, unconfined and multilayer aquifer systems. Several kind of boundary conditions were tested according to the type of solutions. All these tests yield satisfactory results regarding the scope of the application, and model specific situation for each test.

Finally, a model was set up for the aquifer system of Martil situated in Morocco using the GEO-SWIM numerical code, in order to study seawater intrusion effects in terms of the shape and lateral extension. First a steady-state model was
calibrated to obtain adequate model parameters, such as hydraulic conductivities and effective recharge, and the natural situation. Afterwards, a long-term transient simulation was performed to reproduce actual and future situations, with the aim to analyze the risk to salinization from the Mediterranean area. It is concluded that without further control the saltwater interface would travel a land area over considerable distances in the future. A control of saltwater intrusion can be obtained from a number of strategies. A best one is to restrict pumping but this might not be possible from economical and social point of view. However, it is recommended that this should be a future study, for which the present model results could be a starting basis. This present Martil mathematical model is being improved as further data are obtained from the field distribution of pumping, well rates at present and in the future, infiltration recharge in the irrigated area and better water electrical conductivity values measured with the depth in the observed wells along the coast that reveal seawater intrusion.
Chapter 6G

Finite Element Modeling of Three-Dimensional Transport using M-Matrix Preconditioners and Nonsymmetric Solvers

Contents

6.1 Introduction ........................................... 145
6.2 Theory .................................................... 147
6.3 Numerical Model ........................................ 150
6.4 Results and Discussion ............................... 157
6.5 Summary ............................................... 171

6.1 Introduction

Computational modeling of solute transport is undoubtedly one of the most exciting research areas for many hydrologists and modelers during the last decades. A pioneering work in this field has been presented by Remson et al. (1971), and Pin-Adler and Gray (1977) who introduce successively applications of finite difference and finite element methods to contaminant subsurface hydrology problems. These...
methods have gained wide popularity inside the groundwater community, and come available through famous packages and models, such as MOC (Konikow and Bredehoeft, 1978), FEMWASTE (Yeh and Ward, 1981), HST3D (Kipp, 1987) and MT3D (Zheng, 1990). However, solving an advective dispersive problem is a traditionally subject to many sources of possible failure. Typical errors are An-A countered for a high Peclet and Courant numbers, yielding numerical oscillations and/or numerical dispersion, overshooting and undershooting in case of sharp concentration fronts, which leads to constraints on the grid size and computational time step. Other sources of errors or a solution breakdown, depend on the ability of the numerical solver and the chosen preconditioner to converge.

With the noticed rapid advances in computer technology, computational speed and efficiency become affordable, and can be fully used to build complex three-dimensional models for solving practical contamination problems of large size, such numerical approaches yield large, sparse and nonsymmetric linear algebraic systems. The state of the art in nonsymmetric linear solvers is not yet satisfactory when compared to the highly successful preconditioned conjugate gradient solver for symmetric problems, such as those arising from the discretization of the groundwater flow equation. However, the latest achievements accomplished in this discipline are encouraging; modern methods are more reliable, computationally efficient, fast and have a smooth convergence behavior. These techniques known as conjugate gradient like or Krylov-based methods are becoming increasingly popular in dealing with nonsymmetric linear systems. In this class of methods, the conjugate gradient stabilized solver (BICGSTAB) presented by Van der Vorst (1992) is reputed to be the most efficient with respect to the trade-off between storage and convergence speed. Pini and Putti (1994) reported that the method is superior in solving finite element discretizations of the two-dimensional advection dispersion equation. Gambolati and Aslam (1995) applied the method to solve the partial differential equation of a dual porosity transport model.

Preconditioning is considered as a key factor in improving the convergence
behavior of these solvers, in this study, three preconditioning strategies are investigated in terms of efficiency and computational speed, diagonal scaling (DS) is the simplest and made always possible by construction. Incomplete factorization (IF0) which is reputed to be efficient (Larabi and De Smedt, 1994), but can fail for non M type of matrices. M matrix transformation is then proposed on the general transport matrix to allow for the IF0 preconditioner existence (De Smedt A and Sbai, 1998). This transformation proves to be very easy to implement and robust. Its computational efficiency is dependent on the solver used in conjunction (Sbai and De Smedt, 1997), but in all chosen test problems this scheme is the most performing.

This paper starts with a brief review of the finite element scheme used in the presented model to discretize the general three-dimensional advection dispersion equation. Next, the theory of a conjugate gradient-like methods for the solution of nonsymmetric linear systems is highlighted, and algorithms such as minimal residual (MR) and BICGSTAB methods. With some preconditioning strategies are presented. Comparison between these solvers is performed by test problems, using preconditioning by three schemes: DS, IF0, and modified incomplete factorization (MIF0).

6.2y Theory

6.2.1 Governing Equations

The equation governing the transient movement of a reactive chemical in three-dimensional groundwater flow systems, taking into account the advection and dispersion processes, as well as adsorption, first order degradation of chemicals, and source and sink terms is given by Zheng and Bennett (1995)

$$\frac{\partial}{\partial t}(\theta C + \rho S) = \nabla(\theta D \nabla C) - \nabla(q C) - \lambda(\theta C + \rho S) + R_e$$  \hspace{1cm} (6.1)
where $\theta$ is the soil moisture content, $C$ is the dissolved concentration [ML$^{-3}$], $ho$ is the bulk soil density [ML$^{-3}$], $S$ is the adsorbed concentration of the solute [M/M], $t$ is time [T$^{-1}$], $\textbf{D}$ is the hydrodynamic dispersion tensor [L$^2$T$^{-1}$], $q_y$ is the groundwater flux vector [LT$^{-1}$], $\lambda$ is the first order decay coefficient, $R_c$ is an external source/sink rate term [MT$^{-1}$L$^{-3}$], and $\nabla$ is the del operator [L$^{-1}$]. The left hand side of equation (6.1) represents the rate of mass accumulation over a differential volume. In the right hand side, the first term represents the net rate of mass flux due to dispersion and diffusion, the second is the net mass flux due to advection, the third is the degradation rate of the chemical species, and the last term is a source/sink term corresponding to artificial injection and/or withdrawal.

A linear isotherm adsorption model is used to couple the concentrations in the aqueous and adsorbed phases, and may be formulated by

$$S = \mathcal{K}_d C$$

(6.2)

where $\mathcal{K}_d$ is the adsorption distribution coefficient [L$^3$M$^{-1}$].

Combining equations (6.1) and (6.2) yields a linear partial differential equation as expressed below

$$R \frac{\partial C}{\partial t} = \nabla \cdot (\textbf{D} \nabla C) - \nabla (v C) - \lambda RC + \frac{R_c}{\theta}$$

(6.3)

where $R = 1 + \frac{B}{\theta} \mathcal{K}_d$ is the retardation factor, and $\textbf{D}_y = \frac{\partial \textbf{v}}{\partial \theta}$ is the groundwater age velocity [LT$^{-1}$]. The dispersion tensor $\textbf{D}_y$ includes diffusion coefficient $D_{0y}$ [L$^2$T$^{-1}$], and dispersivities coefficients $\alpha_L$, $\alpha_{Th}$ and $\alpha_{Tv}$, respectively the longitudinal, transverse horizontal and vertical dispersivities [L], which multiplied by the different velocity components yield the mechanical dispersion as discussed by Bear (1972).

The groundwater velocity vector is retrieved from a previously computed potential head distribution as

$$\textbf{v}_y = -K_0 \frac{\textbf{v}}{\theta} \nabla h$$

(6.4)
where $\mathbf{K}$ is the hydraulic conductivity tensor [$LT^{-1}$], and $A$ is the hydraulic head [$L$], which is obtained from the solution of the governing equation for a three-dimensional steady state groundwater flow

$$\nabla(\mathbf{K}\nabla h) + Q_h = 0 \quad (6.5)$$

where $Q_h$ is a source/sink flow rate term [$LT^{-1}$].

Solving equation (6.3) for a well posed initial value problem, requires a priori definition of initial and boundary conditions in the domain being under study.

### 6.2.2 Boundary Conditions

Initially, a spatial concentration distribution is known such that

$$C(x, 0) = C_0(x) \quad (6.6)$$

where $x = (x, y, z)$ is the vector position, and $C_0$ is the initial distribution of concentrations.

Generally, there are three types of conditions used in solute transport models:

(a) Prescribed concentrations, or first type Dirichlet boundary condition

$$C(x, t)|_{S_D} = C_D(x, t) \quad t > 0 \quad (6.7)$$

where $C_D$ is a source distribution function specified along the Dirichlet boundary $S_D$.

(b) Fixed dispersive flux, or second type Neuman boundary condition

$$-\theta \mathbf{D} \nabla C|_{S_N} = \mathbf{A}_c(x, t) \quad t > 0 \quad (6.8)$$

where $A_c$ is the solute flux specified along the boundary $S_N$ that is considered impervious for water ($q = 0$).

(c) Fixed total flux, or mixed type, or third type Cauchy boundary condition

$$-\theta \mathbf{D} \nabla C + \mathbf{q} C|_{S_C} = \mathbf{A}_c(x, t) \quad t > 0 \quad (6.9)$$

where $A_c$ is the solute flux specified along the Cauchy boundary $S_C$. 
6.3 Numerical Modeling

6.3.1 They Conforming Finite Element Method

Equation (6.3) subject to initial and boundary conditions (eqns. 6.6 to 6.9) can be solved numerically with the standard Galerkin weighted residual finite element method (FEM), referred also as the conforming FEM in the literature, and with a fully implicit finite difference time stepping scheme. The finite elements used in this study are isoparametric hexahedral elements. The principle of the method is to subdivide the domain into a given number of conveniently small elements sharing a given number of nodes at their corners. A trial solution is given by a function interpolating the dependent variable from the corresponding values at the nodal points, using nodal basis functions, such that the approximate solution over the domain becomes

\[
\hat{C} = \sum_{j=1}^{n} C_j b_j
\]

where \( n \) is the total number of nodes, and \( b_j \) is the linear basis function associated to node \( j \). To fulfill the weighted residual FEM approach, residuals resulting by substitution of Equation (6.10) into Equation (6.3) are minimized by making them orthogonal to the basis functions, yielding a linear system of algebraic equations at time \( t + \Delta t \) (\( t \geq 0 \)), which can be written in matrix form as

\[
([Q] + \frac{1}{\Delta t} [M])\{C\}^{t+\Delta t} = \{A\} + \frac{1}{\Delta t} [M]\{C\}^t
\]

where \([M]\) is the mass transport matrix, \( \{C\}^{t+\Delta t} \) is the vector of unknown concentrations, \([Q]\) is the transport matrix, and \( \{A\} \) represents the external effects on the domain namely the boundary conditions and eventual sources and/or sinks. The entries of the given matrices and vectors are

\[
M_{ij} = \int R_{i} b_j \, dV
\]
\[ Q_{ij} = A \int \nabla b_i (D \nabla b_j - \mathbf{v} b_j) dV + \lambda \int R b_i b_j dV \quad (6.13) \]
\[ B_i = A \int b_i \frac{R_c}{\theta} dV - \int b_i \frac{q_c}{\theta} dS \quad (6.14) \]

The matrix \( \mathbf{Gy} = \mathbf{Q} + \frac{\mathbf{My}}{\Delta t} \) arising from this FE formulation is highly sparse. It is therefore suitable to use a compact vector form by storing separately the upper and lower triangular parts of the Crout LU decomposition of \( \mathbf{G} \). It is obsolete \( A \) from (6.13) that the matrix is symmetry, a property which restricts the choice of suitable solvers. Indeed, preconditioned conjugate gradient methods which are highly efficient for solving linear symmetric and positive definite systems arising from the groundwater flow equations (Larabi and De Smedt, 1994) cannot be applied.

### 6.3.2y Iterative Solvers

During the last two decades methods which are specifically developed for solving linear nonsymmetric systems have attracted the interest of many researchers. Efforts were spent on porting the successfullness of the conjugate gradient methods (CG) for solving linear symmetric systems to the class of nonsymmetric and indefinite systems; these methods are so-called Krylov projection type methods. The literature devoted to this topic is rather huge, and the proposed algorithms can be classified as either orthogonal residual (OR) or minimal residual (MR) based methods (Sleijpen and Van der Vorst, 1993; Barrett et al., 1994; Gambolati et al., 1995).

Among the existing up to date MR solvers, the generalized minimal residual method (GMRES) proposed by Saad and Schultz (1986) is successful for solving this kind of problems as noted by Pini and Zilli (1990). The major drawback of the method is the increasing number of vectors and related storage required for each iteration step. GMRES(m) algorithms are therefore constructed in such a way that the linear iterations are started after \( n \) steps. The choice of \( n \) is also A...
a delicate problem since determining an optimal value still relies on the modeler's experience, which needs typically several runs for tuning up this parameter for each specific problem. Thus, the method is not user-friendly, and especially cumbersome for large three-dimensional systems. In this paper we use the most simple and lowest-cost AMR-based method. The preconditioned AMR method as presented below where $P_y$ is the preconditioning matrix, as explained later.

Table 6.1: The preconditioned Minimum Residual iterative method (Barrett et al., y 1994).

<table>
<thead>
<tr>
<th>Calculate $r_{04} = \tilde{A} - GC_{04}$ from the initial guess $C_{04}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $i = 1, 2, ..., $ until convergence</td>
</tr>
<tr>
<td>Solve $A$ from $P_z = A$</td>
</tr>
<tr>
<td>$\alpha_i = N_i^T G e_i$</td>
</tr>
<tr>
<td>$C_i = C_{i-14} + \alpha_i z_{id}$</td>
</tr>
<tr>
<td>$r_i = \tilde{A}_{i-14} + \alpha_i e_i$</td>
</tr>
<tr>
<td>End For</td>
</tr>
</tbody>
</table>

The biconjugate gradient method (BCG) introduced by Lanczos (1952) was considered as a first natural extension of the three-term short recurrences biorthogonal Algorithms, requiring less storage and computational work per iteration per step. However, an erratic convergence behavior or an eventual breakdown can be observed for ill-conditioned systems. The explicit use of $G^T$ in the algorithm makes it also inappropriate to use for many practical applications. To overcome these limitations the conjugate gradient squared method (CGS) has been developed by Sonneveld (1989). The method is quite competitive, because it preserves the low cost per iteration and avoids the multiplication with the matrix $G^T$. It is reported that the method converges two times faster than BCG whenever this one converges, but as noted by Van der Vorst (1992) as convergence behavior can be worse than BCG. One local instabilities resulting in failure. The biconjugate A
gradient stabilized method was proposed as a smoother variant of CGS, but the convergence can still oscillate remarkably for difficult problems having complex eigenvalues with big imaginary parts as noticed by Sleijpen and Fokkema (1993) and later by Pini and Putti (1994). The preconditioned BICGSTAB seems to be the most attractive OR method with respect to computer storage and convergence speed. Below, are depicted the iterative steps An involved An A preconditioned BICGSTAB method, where $\mathbf{P}$ is again a preconditioning matrix.

<table>
<thead>
<tr>
<th>Table 6.2: The preconditioned BiCGSTAB iterative method (Barrett et al., 1994).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate $\mathbf{A}<em>{04} = \mathbf{A} - \mathbf{G} \mathbf{C}</em>{04}$ from the initial guess $\mathbf{x}_{04}$</td>
</tr>
<tr>
<td>Choose an arbitrary vector $\mathbf{A}<em>{04}$ such that $\mathbf{A}</em>{04}^{T} \mathbf{A}<em>{04} \neq 0$ (e.g. $\mathbf{A}</em>{04} = \mathbf{A}_{0}$)</td>
</tr>
<tr>
<td>Initialize the iterative parameters: $\mathbf{A}$</td>
</tr>
<tr>
<td>$\rho_{04} = \mathbf{A}_{04} = 1$</td>
</tr>
<tr>
<td>$\rho_{04} = \mathbf{A}_{04} = 0$ A</td>
</tr>
<tr>
<td><strong>For</strong> $i = 1, 2, \ldots$ until convergence</td>
</tr>
<tr>
<td>$\rho_{i} = \mathbf{A}<em>{i-1}^{T} \mathbf{A}</em>{i-1}$</td>
</tr>
<tr>
<td>$\beta = (\rho_{i}/\rho_{i-1})(\alpha/\omega_{i-1})$</td>
</tr>
<tr>
<td>$p_{i} = \mathbf{A}<em>{i-1}^{T} p</em>{i-1}$</td>
</tr>
<tr>
<td>$v_{i} = \mathbf{A}<em>{i-1}^{T} p</em>{i-1}$</td>
</tr>
<tr>
<td>$t = \mathbf{A} \omega_{i}$</td>
</tr>
<tr>
<td>$\mathbf{A}<em>{i} = \mathbf{A}</em>{i-1}^{T} \alpha v_{i} + \omega_{i} t$</td>
</tr>
<tr>
<td>End For</td>
</tr>
</tbody>
</table>

Gutknecht (1993) has refined this algorithm for the case of complex eigen-A values, which leads to better convergence behavior. He showed that the latter is a combination of ABCG and GMRES(2). At this point, A View was generalized by Sleijpen and Fokkema (1993) who describe how to derive a BICGSTAB variant of order $l$: BICGSTAB $B(l)$, and give a practical implementations for low orders. A
6.3.3j Preconditioning Methods

Preconditioning is a key factor to the success of any iterative solver, especially when dealing with practical applications of large size. The state of the art is still not satisfactory, because much effort was given for developing more robust solvers at a first stage. The idea behind it is to transform the original system of equations by multiplying with $P^{-1}$, such that the new system is easier to solve, which implies that the preconditioning matrix $Py$ should resemble $G$ as close as possible. A good preconditioner produces enough again an convergence rate to overcome the extra cost and related to its own construction. For a more detailed discussion in this subject we refer to Barret et al. (1994).

Three preconditioning schemes were implemented in this study to seek the performance of the selected solvers:

1. A diagonal scaling (DS): $Py = diag(G)$, point Jacobi preconditioning is the easiest to implement since there is no extra storage beyond that of $G$. One can notice the unconditional existence of $G$ transformation. By scaling the original matrix the spectrum becomes smaller, and thus the convergence rate is improved. Unfortunately, in many cases, it is observed that $G$ preconditioner does not drastically improve the solver performance, and the convergence might be slow. A more powerful preconditioning scheme is therefore needed for such problems.

2. A point incomplete factorization (Armed: $AIF0$): the preconditioning matrix is often formulated as the form $Py = (L + D)D^{-1}(D + U)$, where $L$ and $A$ are respectively the strictly triangular lower and upper parts of $G$, and $D$ is a positive diagonal matrix, such that $diag(P) = diag(G)$. The essence
of the method is to prohibit fill-in positions where \( A_{ij} = 0 \) during the fac-A

orization. A\( \text{Meaning} \) \( A_{ij} \), \( L_y \) and \( U_y \) have the same sparsity pattern as \( A \).

An attractive feature of the method is a reasonable construction cost. Since \( A \) only \( \hat{D} \) \( \text{diagonals} \) \( A \) \( \text{needed} \) \( A \) \( \text{storage} \) \( A \) \( \text{required} \). A\( \text{The} \) entries of \( \hat{D} \) \( \text{can} \) be computed recursively:

\[
D_{ii} = \hat{K}_{ii} - \sum_{k=14}^{i-1} \frac{G_{ik} G_{ki}}{D_{kk}}
\]

(6.15)

3. A point modified incomplete factorization of order zero (MIF0): \( A \) \( \text{the same form} \) as \( \text{MIF0} \) except that \( \hat{D} \) \( \text{is} \) \( \text{constructed} \) \( \text{in} \) \( \text{such} \) \( \text{way} \) as \( \text{MIF0} \). A\( \text{the row sum constant, the entries of} \) \( \hat{D} \) \( \text{are} \) computed in this case as:

\[
D_{ii} = \hat{K}_{ii} - \sum_{k=14}^{i-1} \sum_{j=k+14}^{n} \frac{G_{ik} G_{jk}}{D_{kk}}
\]

(6.16)

For a more practical implementation the entries of \( \hat{A} \) are calculated as:

(Beauwens, 1990)

\[
U_i = \sum_{j=14}^{n} G_{ij} - \sum_{k=14}^{i-1} \frac{G_{ik} U_k}{D_{kk}}
\]

(6.17a)

\[
D_{ii} = \hat{N}_{ii} - \sum_{k=i+14}^{n} G_{ki}
\]

(6.17b)

The preconditioners \( IF0 \) and \( MIF0 \) only work when the obtained \( A \) \( \text{diagonals} \) \( A \) \( \text{positive} \), and it turns out that there are cases for which \( IF0 \) and \( MIF0 \) are not \( A \) \( \text{guaranteed} \) \( A \) \( \text{exist} \) \( A \) \( \text{Meijerink} \) and \( \text{Van der Vorst} \) (1977) proved that precondi-

tioning by \( IF0 \) is possible only if \( \hat{A} \) \( \text{satisfies} \) the requirements of an \( M \) matrix \( A \) \( G_{ii} > 0 \), and \( \hat{A}_{ij} < 0 \) for \( A \neq \hat{A} \), and \( \hat{A} \) \( \text{is} \) \( \text{non} \) \( \text{singular} \). A\( \text{However}, \) for \( MIF0 \) the \( A \) \( \text{existence} \) \( A \) \( \text{not} \) \( A \) \( \text{guaranteed} \), Beauwens and Quenon (1976) noticed that \( A \) \( \text{factorization} \) \( A \) \( \text{exists} \) \( \text{for} \) \( \text{diagonally} \) \( \text{dominant} \) \( A \) \( \text{matrices} \). A\( \text{Hence}, \) for \( A \) \( \text{any} \) \( A \) \( \text{practical} \) \( A \) \( \text{applications} \) these kind of methods may fail. A\( \text{To overcome} \) this drawback we \( A \) \( \text{suggest} \) the use of the following \( M \) matrix approximation \( \hat{A} \) \( A \) \( \text{of} \) \( A \) \( A \) \( \text{such} \) \( A \)

\[
(G_M)_{ij} = \text{Anin}(G_{ij}, 0)
\]

(6.18)
\[(G_M)_{ii} = \sum_{j=1}^{n} \max(G_{ij}, 0) \]  

(6.19)

Practically, the global matrix is lumped by moving positive off-diagonals to the diagonal position while keeping negative off-diagonals, such that row sums are kept constant. The proposed method was successfully used by Larabi and De Smedt (1994) to solve linear systems stemming from three dimensional hexahedral finite element discretizations of the steady state groundwater flow equation. The previously presented algorithms and preconditioners are efficiently implemented in a user friendly numerical tool kit, which is integrated to GEO-PROF numerical code (De Smedt, 1996).

Figure 6.1: Schematic representation of the test problems; (A) Injection in a uniform flow field; (B) Radial injection with equilibrium counter dispersion; (C) Radial injection; (D) Field problem 1; and (E) Field problem 2.
6.4 Results and Discussion

6.4.1 Results and Discussion

Five test problems are selected to investigate the solvers and preconditioners performance. These examples include theoretical and field applications illustrating a diversity of physical situations as shown in Fig. 6.1. Table 6.3 shows several of the numerical features of the studied examples, as the total number of nodes and elements used, the problem discretization, the shape of the elements, the degree of sparsity of the GM matrix, the maximum Peclet and Courant numbers, and if the M matrix requirement is satisfied or not. Additionally, physical parameters of the test problems used to run the models are given in Table 6.4. A. The numerical results are obtained by executing the numerical models on a Sun UltraSparc 2A desktop workstation. For each test problem, the tolerance related to the convergence stopping criterion (sum of squared residuals) is denoted by tol parameter, this is either previously specified or determined automatically by an internal routine implemented in the models. The obtained results in terms of total number of iterations required to achieve the convergence criterion, and the used CPU calculation time are also checked.

In the remainder we denote by I the solution procedure without preconditioning, and by AF0-M and MIF0-MA he complete factorization preconditioners with M matrix transformation, as proposed in equations 6.18 and 6.19.

<table>
<thead>
<tr>
<th>Test No.</th>
<th>No. of Nodes</th>
<th>No. of Elements</th>
<th>Element Shape</th>
<th>Degree of Sparsity (%)</th>
<th>M Matrix</th>
<th>PE</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>50000!</td>
<td>54756!</td>
<td>box!</td>
<td>99.97!</td>
<td>Yes!</td>
<td>4</td>
<td>-!</td>
</tr>
<tr>
<td>1b</td>
<td>50000!</td>
<td>54756!</td>
<td>irregular!</td>
<td>99.97!</td>
<td>No!</td>
<td>4</td>
<td>-!</td>
</tr>
<tr>
<td>2a</td>
<td>672!</td>
<td>300!</td>
<td>wedge!</td>
<td>99.98!</td>
<td>No!</td>
<td>1!</td>
<td>-!</td>
</tr>
<tr>
<td>2b</td>
<td>6432!</td>
<td>3000!</td>
<td>wedge!</td>
<td>99.98!</td>
<td>No!</td>
<td>2!</td>
<td>-!</td>
</tr>
<tr>
<td>3</td>
<td>1608!</td>
<td>600!</td>
<td>wedge!</td>
<td>99.98!</td>
<td>No!</td>
<td>1!</td>
<td>5!</td>
</tr>
<tr>
<td>4</td>
<td>21000!</td>
<td>16284</td>
<td>irregular!</td>
<td>99.94!</td>
<td>No!</td>
<td>37!</td>
<td>-!</td>
</tr>
<tr>
<td>5</td>
<td>77964</td>
<td>69696!</td>
<td>irregular!</td>
<td>99.93!</td>
<td>No!</td>
<td>48!</td>
<td>-!</td>
</tr>
</tbody>
</table>
### Table 6.4: Physical parameters of the test problems

<table>
<thead>
<tr>
<th>Physical parameters</th>
<th>Test problems</th>
<th>Layer 1</th>
<th>Layer 2</th>
<th>Layer 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K!(\text{m/d})! )</td>
<td>10!</td>
<td>5!</td>
<td>10!</td>
<td>12!</td>
</tr>
<tr>
<td>( \alpha_L) (m!)</td>
<td>5!</td>
<td>5!</td>
<td>0.1</td>
<td>100!</td>
</tr>
<tr>
<td>( \alpha_{TB}) (m!)</td>
<td>0.5!</td>
<td>5!</td>
<td>0.1</td>
<td>33</td>
</tr>
<tr>
<td>( \alpha_{TV}) (m!)</td>
<td>0.5!</td>
<td>5!</td>
<td>0.1!</td>
<td>0.33!</td>
</tr>
<tr>
<td>( D_0!(\text{m}^2/\text{d})! )</td>
<td>0</td>
<td>0!</td>
<td>0!</td>
<td>10(^{-7})</td>
</tr>
<tr>
<td>( n )</td>
<td>0.25!</td>
<td>0.35!</td>
<td>0.387!</td>
<td>0.25!</td>
</tr>
</tbody>
</table>

6.4.1 Test Problems 1: y Continuous Point Injection Uniform Flow Field y

The first test example is based on an available three-dimensional analytical solution from Hunt (1978) for steady state solute transport in a uniform flow field from a continuous point injection source, and given by

\[
C = A \frac{M}{8\pi n R D_T} \exp \left( \frac{v x}{2 D_L} \right) \left[ \exp \left( -\frac{R v}{2 D_L} \right) \text{erfc} \left( \frac{R - v t}{2 \sqrt{D_L t}} \right) 
+ \exp \left( \frac{R v}{2 D_L} \right) \text{erfc} \left( \frac{R + v t}{2 \sqrt{D_L t}} \right) \right] \tag{6.20A}
\]

where \( R = \sqrt{x^2 + (y^2 + z^2)} \) [L], \( M \) is the mass injection rate [MT\(^{-1}\)], \( v \) is the uniform groundwater flow velocity [LT\(^{-1}\)] along the longitudinal distance \( x \), \( D_L \) and \( D_T \) are the longitudinal and transverse dispersion coefficients respectively [L\(^2\)T\(^{-1}\)], and \( \text{erfc} \) is the complementary error function.

In the present test example the pollution plume is simulated for a constant groundwater velocity of 0.5 m/d and a mass input rate of 1000 kg/d; other parameters are given in Table 6.4. The model region is reduced to one quadrant with a solute source injection point at the origin. Two test runs are conducted with different meshes to study the effect of the matrix transformation. The first run (1a) is done by discretizing the domain into box sized elements of 4x2x2 m\(^3\). While in the second test (1b) the previous mesh is modified by adding or subtracting a constant value of 0.5 m randomly to the coordinates of all interior nodes. At test A
problem (1a) yields a M matrix, while problem (1b) is non M matrix.7

![Figure 6.2: Convergence history analysis of test problems (a) 1a and (b) 1b.](image)

Figure 6.2: Convergence history analysis of test problems (a) 1a and (b) 1b.

The obtained results are shown in Table 6.5. In the table are indicated: the number of iterations needed to obtain convergence, NI; the total runtime, CPU(s); the solute mass balance error, MBE expressed in percentage, and the maximum error between calculated and exact concentrations over the hole mesh, excluding the injection point. The highest errors are found at the model boundaries, as the analytical solution as established assuming infinite spatial dimensions. The total mass balance error is somewhat higher for the unpreconditioned methods, nevertheless all are satisfactory.

All algorithms except MIF0 converge for test problem 1a, and an M type of
transformation is unnecessary. Obviously, the MR method preconditioned with IF0 is the fastest, although more iterations are needed compared to BICGSTAB. For the other preconditioners, BICGSTAB is the fastest. The convergence behavior of the different solvers as illustrated in Fig. A6.2(a). Obviously, IF0 enables a steady decrease in the relative residual norm. The same convergence behavior was observed for the second run 1b, as can be concluded from Fig. A6.2(b). For this case, the IF0 preconditioner unexpectedly exists, and turns out to be the most efficient solver, while M transformation requires a little less iterations but somewhat more execution time.

6.4.2 Testy Problem 2: Steady State Transport in a Radially Flow Field with Country Dispersion

Recharge well is injecting clean water in an initially polluted aquifer. A fixed concentration boundary condition is imposed at a distance $aR$ from the well, such that steady-state advection and dispersion act in inverse directions. By neglecting sorption and decay mechanisms, the steady-state transport equation simplifies to:

$$\frac{\partial^2 C}{\partial r^2} + \frac{1}{r} \frac{\partial C}{\partial r} = \frac{\partial C}{\partial S}$$

(6.21)

where $\alpha_r = \frac{r S}{\alpha_L}$, and $\alpha$ is the radial coordinate [L]. Equation 6.21 is easily solved for the boundary condition:

$$C = C_{04} \quad \text{at} \quad R = \infty$$

(6.22)

such that

$$\frac{C}{C_{04}} = \exp \left( \frac{r - R}{\alpha_r} \right)$$

(6.23)

Assuming a well radius $a = 0.1m$, a discharge rate of $15m^3/d$, a radius $R$ of $0.001m$, with a unit upstream fixed concentration, i.e. $AC_{04} = 1$, this problem is solved...
<table>
<thead>
<tr>
<th>Test</th>
<th>Solver</th>
<th>Preconditioner</th>
<th>NI (s)</th>
<th>CPU (s)</th>
<th>MBE (s)</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MR!</td>
<td>5!</td>
<td>1024</td>
<td>696.2</td>
<td>1.17</td>
<td>1.467</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DS!</td>
<td>799</td>
<td>369.5</td>
<td>0.85</td>
<td>1.467</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IFO!</td>
<td>67</td>
<td>64.6</td>
<td>0.15</td>
<td>1.467</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIFO! 1</td>
<td>*1</td>
<td>*1</td>
<td>*1</td>
<td>*1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IFO-M!</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>(a)!</td>
<td>BICGSTAB!</td>
<td>II!</td>
<td>144</td>
<td>111</td>
<td>0.27</td>
<td>1.467</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DS!</td>
<td>138.7</td>
<td>133.9</td>
<td>0.05</td>
<td>1.467</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IFO!</td>
<td>31</td>
<td>74.3</td>
<td>0.14</td>
<td>1.467</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIFO! 1</td>
<td>*1</td>
<td>*1</td>
<td>*1</td>
<td>*1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IFO-M!</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>II!</td>
<td>960</td>
<td>665.6</td>
<td>1.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DS!</td>
<td>735</td>
<td>359.5</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IFO!</td>
<td>77</td>
<td>63.45</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIFO! 1</td>
<td>*1</td>
<td>*1</td>
<td>*1</td>
<td>*1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IFO-M!</td>
<td>73</td>
<td>64</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIFO-M!</td>
<td>460</td>
<td>448.6</td>
<td>0.54</td>
<td></td>
</tr>
<tr>
<td>(b)!</td>
<td>BICGSTAB!</td>
<td>II!</td>
<td>159</td>
<td>142.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DS!</td>
<td>129</td>
<td>155.6</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IFO!</td>
<td>36</td>
<td>83.4</td>
<td>0.09</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIFO! 1</td>
<td>*1</td>
<td>*1</td>
<td>*1</td>
<td>*1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IFO-M!</td>
<td>38</td>
<td>90.9</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIFO-M!</td>
<td>72</td>
<td>309.5</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

- ! Solver not executed!
- ! Residual stagnates before maximum number of iterations (10000) was reached!

using a wedge and adopting two mesh configurations. Test runs (2a) and (2b) A use respectively 21x16x2 and 201x16x2 nodes along a three-dimensional radial coordinate system. For test run (2a) the spacing along the radial axis is 5m and for (2b) it is 3m except for 50m around the well center where the nodal positions follow an exponential sequence in incremental order starting form 0.045m at the well.

For the first run with the coarse mesh and using a tolerance of 10^{-12} all pre-conditioned algorithms perform very well and need very few iterations to converge.
Figure 6.3: Comparison of analytical and numerical solutions of normalized concentration versus radial distance for test examples 2a and 2b.

as shown in Fig. 6.4(a). Surprisingly, the obtained numerical results are not very accurate as demonstrated by the comparison versus analytical results shown in Fig. 6.3. Especially close to the well, the computed results deviate noticeably; this lack of accuracy results from the fact that the advective component of the solute transport is very important in this region, such that element velocities need to be approximated more accurately.

For the second run the different solvers and preconditioners are tested with the fine mesh and severe convergence criteria ($tol = 10^{-30}$). The performance of the different solvers are presented in Table 6.6, showing that all MR based methods fail to converge; in this case the iterative process either stagnates, as for DS, IF0 and IF0-M, or eventually breaks down as for MIF0 and MIF0-M. The same convergence behavior is observed for the BICGSTAB based methods, except for IF0 and IF0-M, which are the only successful methods (Fig. 6.4(b)). The error norm over the hole mesh becomes very small, and the obtained results of the
Figure 6.4: A convergence history analysis of test problems (a) 2a and (b) 2b.w numerical solutions are much more accurate than in the first run, as shown in Fig. A. However, slight deviations from the exact solutions remain, indicating that this seemingly simple problem is rather difficult to solve numerically.

6.4.3 TestyProblemy3:y TransientyTransportyinyayRadially Velocity Fieldy

The radial dispersion problem is typical for describing the movement of a tracer, injected from a recharge well of finite radius. Several authors have analyzed this problem by deriving approximate and exact analytical solutions, such as Tang and
Babu (1979), Hsieh (1986) and Yates (1988), who gives also additional solutions for variable boundary conditions at the well. The same problem was also solved by Hoopes and Harleman (1967) using a finite difference based model.

finite element wedge of 600 elements in 3 layers composed of 200 elements along the radial direction, with uniform nodal spacing of 0.1m is adopted. Model parameters and analytical solution values used for comparison purpose are taken from Segol (1994). A user-specified time step of 0.01 day is used for computing the solution at time 0.1 and 1 day. Corresponding numerical and analytical solutions -in terms of \( A^0 / C_{0a} \) versus radial distance- are plotted in Fig. 6.5 showing a fairly good agreement. A fixed tolerance of 10^{-5} was used to obtain these results.

All test runs except BICGSTAB preconditioned with IF0-M fail to converge. The solution obtained with BICGSTAB is computed in about 547.3 CPU and a total number of 439 iterations. All other solvers exhibit a typical oscillatory convergence behavior at a time which fails to converge before the maximum number of allowed iterations. At 10000, a converged solution is obtained. A test problem as a very difficult to solve numerically.

<table>
<thead>
<tr>
<th>Solver</th>
<th>Preconditioner</th>
<th>NI</th>
<th>CPU (s)</th>
<th>Error (10^{-2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR!</td>
<td>I!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td></td>
<td>DS!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td></td>
<td>IF0!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td></td>
<td>MIF0!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td></td>
<td>IF0-M!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td></td>
<td>MIF0-M!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td>BICGSTAB</td>
<td>I!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td></td>
<td>DS!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td></td>
<td>IF0!</td>
<td>452!</td>
<td>208.35!</td>
<td>1.36!</td>
</tr>
<tr>
<td></td>
<td>MIF0!</td>
<td>426!</td>
<td>173.4!</td>
<td>1.36!</td>
</tr>
<tr>
<td></td>
<td>IF0-M!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td></td>
<td>MIF0-M!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
</tbody>
</table>

*!Residual!stagnates!before!maximum!number!of!iterations!(10000)!wastreached!
#!Iterations!stopped!when!division!by!zero!was!encountered!
6.4 Results and Discussion

![Graph showing analytical and numerical solutions of normalized concentration versus radial distance for a test example. Outputs are plotted at time levels 0.1 day and 1 day.]

6.4.4 TestyProblem4: yFirstyFieldyExampley

The objective of this field application study is to predict the possible contamination of a well from a nearby waste disposal site. The water abstraction rate at the groundwater pumping station equals 12,000 m$^3$/d and a pollution injection rate of 100 kg/d is infiltrating from the waste site as depicted in Fig. 6.1(d). The velocity field obtained by solving the groundwater equation by an adaptive moving mesh procedure such that only the saturated part of the domain is considered. The water table position fits exactly with the iteratively adjusted upper mesh layer as shown in Fig. 6.6(a), where also iso-potential surfaces and velocity vectors at the water table are shown, clearly indicating the radial and convergent flow near the well. The element velocity extrema is found to be equal to 6.17 m/d near the well location, which results in a high Peclet number of approximately 37 for this problem. Test runs are performed for the steady state transport sit-
Figure 6.6: Simulated steady state (a) potentials, water flow velocities, and (b) concentration iso-surfaces for test problem 4.

ulation, and solver performances corresponding to $tol = 10^{-12}$ are presented in Table 6.7. Most solvers fail, except when DS preconditioned or M matrix preconditioned. The successful preconditioned MR based methods are slow, while the IF0-M preconditioner needs less number of iterations compared to DS. The successful preconditioned BICGSTAB methods show a very important speed-up and less number of iterations. We can also notice the good performance of DS for this problem, but IF0-M is exceptionally good, while MIF0-M has a poor performance in comparison. As an illustration, numerical results are visualized in Fig. 6.6(b), with iso-concentration surfaces, showing the extent of the pollution
plume at steady state.

![Graph showing convergence history analysis of test problem 4.](image)

Figure 6.7: Convergence history analysis of test problem 4.

### 6.4.5y TestyProblem5:ySecondyFieldyExampley

The pollutant transport from a landfill area of irregular geometry is studied in a phreatic aquifer system with three different lithologic layers. The landfill is in contact with the water table, such that as a boundary condition the concentration at the water table are considered fixed. The groundwater movement passes through and underneath the landfill, flowing to a nearby canal. Also, recharge is added to the aquifer, diluting the pollution plume. The transport is calculated until steady state is reached. A three-dimensional finite element mesh is designed consisting of 73x89x12 nodes. Figure A.6.8 shows simultaneously the groundwater heads isolines, and the direction and magnitude of the velocity vectors driving
the steady pollution at the aquifer bottom. A The specified tolerance for a convergence was chosen as $\alpha_0 = 10^{-124}$ and numerical results are presented in Table 6.8. Exceptionally, the IF0 preconditioning exists although the problem is non-symmetric matrix type. The most important finding is that BICGSTAB based methods are faster, and that DS shows to be competitive when used in combination with a robust solver.

By comparing all previous results, an immediate conclusion is that the BICGSTAB method is more efficient than MR, because it requires less computational time and is always the fastest. A For problems 2a and 3a, where AMR fails with all the preconditioners, BICGSTAB is still successful with some of the studied preconditioners. Unpreconditioned algorithms do not converge except for test problem 1a, because the elements have a simple and regular geometry such that $Gy$ is an $M$ matrix. With respect to the preconditioners, one can notice that DS converges for all tests except for problem 2b. And, but is much slower than IF0. However, DS could be more efficient in massively parallel computers, since it is the only...
preconditioner that can be parallelized easily. Also, it is very robust and always exists. Clearly, IF0 and especially IF0-M perform very well, and in combination with BICGSTAB yield the most efficient and fastest solver. MIF0 and MIF0-M preconditioning strategies are less efficient due to the related expensive cost per iteration, this is in agreement with the conclusions obtained by Jacobs (1984) and Larabi and De Smedt (1994) for preconditioned conjugate gradient methods.

The results arising from runs (1a) and (1b) show that the loss of the desirable M matrix property does not affects the rate of convergence of the solvers, and the superiority of each method is preserved. It is also instructive to notice that the MIF0 preconditioner does not exist for test problems 1b, 2b, and 5, but that an IF0 decomposition exists even for a non M matrix. This is not surprising since the M condition is a sufficient but not essential condition for the existence
### Table 6.8: Solver performance for the fifth test problem

<table>
<thead>
<tr>
<th>Solver</th>
<th>Preconditioner</th>
<th>NI</th>
<th>CPU(s)</th>
<th>Error($10^{-2}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR!</td>
<td>!</td>
<td>1!</td>
<td>1!</td>
<td>1!</td>
</tr>
<tr>
<td></td>
<td>DS!</td>
<td>1544</td>
<td>772.2!</td>
<td>0.437!</td>
</tr>
<tr>
<td></td>
<td>IF0!</td>
<td>166!</td>
<td>257.5!</td>
<td>0.567!</td>
</tr>
<tr>
<td></td>
<td>MIF0!</td>
<td>x!</td>
<td>x!</td>
<td>x!</td>
</tr>
<tr>
<td></td>
<td>IF0-M!</td>
<td>184!</td>
<td>288.9!</td>
<td>0.597!</td>
</tr>
<tr>
<td></td>
<td>MIF0-M!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
<tr>
<td>BICGSTAB</td>
<td>!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DS!</td>
<td>106!</td>
<td>150.2!</td>
<td>1.73!</td>
</tr>
<tr>
<td></td>
<td>IF0!</td>
<td>35!</td>
<td>126.1!</td>
<td>0.635!</td>
</tr>
<tr>
<td></td>
<td>MIF0!</td>
<td>x!</td>
<td>x!</td>
<td>x!</td>
</tr>
<tr>
<td></td>
<td>IF0-M!</td>
<td>37!</td>
<td>152.9!</td>
<td>0.9!</td>
</tr>
<tr>
<td></td>
<td>MIF0-M!</td>
<td>#!</td>
<td>#!</td>
<td>#!</td>
</tr>
</tbody>
</table>

*!Residual stagnates before maximum number of iterations! (10000)! was reached! 
#! Iterations stopped when division by zero was encountered! 
x! preconditioner does not exist!

The AF0Adecomposition (Meijerink and Van der Vorst, 1977). A However, in problems having irregular and wedge shaped elements, preconditioning and MA matrix transformation are essential. At The BICGSTAB solver’s preconditioned example IF0-MA shows the most efficient, and proves to be of value in repeatedly long-term transient simulations, as in a case of problem A. Indeed, in another case of the MRA solver, IF0-MA and DS are quite comparable, but for BICGSTAB preconditioning A by IF0-M results in an important speedup. One can also observe that the repeated oscillatory changes of the residual norm for the BICGSTAB applications becomes less pronounced when preconditioned by AF0, and that MR preconditioned by AF0A has in contrast a smoother convergence behavior, and shows to be competitive. However, for all cases, BICGSTAB preconditioned with AF0-MA performs always the best.
6.5 Summary

Three-dimensional numerical modeling of pollutant transport in aquifer systems by the finite element method leads to large linear systems that are sparse and nonsymmetric. The global transport matrix arising from the discretization using hexahedral elements does not in general satisfy the requirements of an $M$ ma-trix, which is a very desirable property with respect to the numerical solution procedure. The efficiency of preconditioned conjugate gradient like solvers such as AMR and BICGSTAB was investigated. Five representative test examples were selected as a basis for this comparison, and several strategies were adopted to conduct different numerical simulations, based on the implemented preconditioners, stopping convergence criteria, and solutions benchmarks when an exact solution was available.

Preconditioners as diagonal scaling, incomplete factorization, and modified incomplete factorization were tested. An $M$ matrix transformation was proposed which guarantees the existence of an incomplete factorization, the most efficient preconditioner with respect to the number of iterations, and CPU cost were used as a basis for these comparisons, also a series of benchmarks for all succeeded tests were performed for problems, which could be solved analytically, to ensure the accuracy of the obtained results.

It is found that BICGSTAB preconditioned by incomplete factorization per-A forms well for all test problems. However, for difficult problems such as incomplete factorization is unlikely to exist, and the proposed $M$ matrix transformation proves to be effective, leading to the guaranteed existence of a robust and efficient solver.
Chapter 7

SoftwareyDevelopment and yGUIy
For Models Support y

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Introduction</td>
<td>173</td>
</tr>
<tr>
<td>7.2</td>
<td>General Overview</td>
<td>174</td>
</tr>
<tr>
<td>7.3</td>
<td>GEO-SWIM Architecture</td>
<td>175</td>
</tr>
<tr>
<td>7.4</td>
<td>Visualization Tools</td>
<td>179</td>
</tr>
</tbody>
</table>

7.1 Introduction

In this chapter we shall discuss issues which are more relevant to software engineering, rather than the models concepts, mode of functioning, and implemented approaches themselves. These aspects are reported herein, because they are valuable tools to support some of the models already discussed in the previous chapters, and take effective role in a number of ways for a model setup, run and analysis.

The following descriptions are restricted to the saltwater intrusion model GEO-SWIM. However, pre and post-processor modules are shared within GEO-C PROF numerical code for which the models described in chapter 4 and 6 are the newly implemented pieces, contributed in the framework of this work.
7.2.2 General Overview

A new system for finite element modeling of three-dimensional groundwater flow and saltwater intrusion in Aquifer Systems is developed, AGEO-SWIM (Geohydromological Saltwater Intrusion Model). It has several features including flexibility, computational efficiency, portability, and handling a large variety of physical conditions, making it suitable for a wide range of practical and real-life applications. A steady-state as well as transient problems can be investigated in a three-dimensional, heterogeneous, porous media. The software builds upon a different module, including a pre-processor and a post-processor module. The implemented numerical algorithms are efficiently coded to optimize computer storage, memory management, and computing time. A Visual Support to input data, and a GUI output results, is given through an integrated program that interfaces with different ways with the simulators, and another developed GUI (Graphic User Interface) tools.

Modeling of groundwater flow taking into account variations in density as a traditionally prerequisite when Aquifer Systems are involved. A Finite Difference and Finite Element Computer Codes have been developed for such applications, such as SWIM (Sa da Costa and Wilson, 1979); SUTRA (Voss, 1984); ASHRAE, Essaid, 1990a); AMAGNAS (Huyakorn et al., 1994a,b); A and SIMLAS (Huyakorn et al., 1996). Sharp interface models are yet the most common, economical and practical for these types of simulations. Almost all existing models in this category are either 2D or quasi-3D and have limited built-in GUI and visualization capabilities, except for SUTRA, recently supported by a number of 2D visual routines in Argus ONE (Voss et al., 1997). Driven by the need for a fully three-dimensional sharp interface model, GEO-SWIM has been developed (Sbai and De Smedt, 1989) including a pre-processor and a post-processor package that communicate with the simulators through an interface calls. The developed system is highly efficient, simple in use, portable, interactive in a way...
to provide modular and user-friendly tools to handle separate modeling tasks, and open to other software packages leading in their field.

### 7.3y GEO-SWIM Design

Before presenting the general structure of the developed software packages, how they interface and inter-depend, the objectives are discussed first.

#### 7.3.1 Design Goals

The following were design goals for GEO-SWIM with their motivation and implications.

**Highly Numerical Performance**

GEO-SWIM takes advantage from latest developments and approaches for solving the sharp fresh-salt water interface, by avoiding solving the equivalent two fluid flow problem. This was achieved by transforming the saltwater (and eventually the unsaturated) part of the domain, to an equivalent freshwater domain having the same pressure distribution as in the saltwater domain. Also, numerical difficulties prone to the non-linear discrete Richards equation are handled. Another reason for such numerical robustness is due to the efficiency and level of optimization. An example, an important improvement was observed when converting many parts of the code from FORTRAN 77 to FORTRAN 90 language, by using dynamic memory allocation, intrinsic array functions, and modules for organizing data structures, objects and shared variables, and avoiding the use of obsolescent FORTRAN 77 features.

**Portability**

GEO-SWIM was designed to run under any platform and operating system. The code was tested on the following systems:
- Dos/Windows 95/98/NT for Intel based PCs,

- UNIX-based Solaris for SUNSPARC workstations,

- UNIX-based IRIX for SGI. A

Numerical models are ported successfully, at no cost. There is only an exception for Windows based interfaces, because such tools use Windows 32-bit based GUI and graphic library calls, which are not easily ported to other environments. A

**Simple Programming Interface**

It was decided to use a primary language to develop all packages. For instance, all numerical routines used to build different models were developed from scratch, making the numerical code self-dependent, or on the other hand, independent of any hardware, compiler, or system specific library calls. We preferred also to keep each program independent of others, so that have a programming layout as simple as possible, which allows a user with an average programming knowledge, the ability to navigate through and change code parts according to his specific need. However, this gives however more flexibility, but less security. But, according to the previous goal this was allowed in the release. Another strategy is to lock access to the user. This is to prevent possible damage by providing a set of PI (Application Programming Interface) calls to built-in libraries. This issue may be considered in the next release. A

**Highly Interactive Research Tool**

Recently developed GUI and graphical packages, aim at providing a highly interactive and valuable research tool. These programs are developed in a modular way, which implies an easier level of maintenance. A
7.3 GEO-SWIM Architecture

**Simulators**
- Saltwater Intrusion Models
  - Geo_Sswi | Geo_Tswi
- Flow Lines Simulator
  - Geo_Flow
- Nodal Velocities and Fluxes Calculator
  - Geo_Flux

**Post-Processors**
- GIS Converter for Soil types
  - Geo_GIS
- Tecplot Converter Program for 3D Visualization
  - Geo_Tecp
- CAD Converter Package (DXF)
  - Geo_Acad

Figure 7.1: GEO-SWIM modules

### 7.3.2y Structurey

GEO-SWIM is composed from different packages as depicted in Fig. 7.1. AttheyA are classified as pre-processors, post-processors or models, which constitutes the core system component. A

**Pre-Processor Packagesy**

A typical finite element simulation in GEO-SWIM starts Aby constructing a three-A dimensional mesh of the aquifer system, and attributing soil types and related physical parameters Ato Athe Asystem. AThis Ais Aaccomplished Ausing Athe AGEO-A
GRID mesh generator for making structured grids from irregular hexahedral finite elements, and GEO-SOIL a soil types interpolator package, which attributes soil types to all elements from an initial number of given vertical geological cross-sections.

Boundary conditions on external boundaries and/or internal nodes (e.g. water abstraction nodes) are then specified prior to run the saltwater intrusion model. The package GEO-FCON handles this task and additionally a user-friendly windows GUI version of this package was designed to facilitate user input.

Twelve types of nodal boundary conditions representing a wide range of possibilities encountered in practice are allowed. GEO-FCON also has the ability to specify soil types as conditions confined to elements based on structured sub-zones of the whole mesh, this option is not powerful as the use of GEO-SOIL, but it is faster in case of simple situations (e.g. horizontally layered aquifers).

**Saltwater Intrusion Models**

Two models are included in the software package, plus two other auxiliary programs for tracking flow lines, and velocity calculation.

GEO-SSWI: is the steady state saltwater intrusion model, which is based on a symmetric conjugate gradient flow solver, preconditioned with a modified M-matrix incomplete factorization. The conductance matrix coefficients are implicitly corrected, depending on the nodal water pressure status.

GEO-TSWI: simulates the moving salt-fresh water interface in groundwater aquifer systems. This program uses a finite difference approximation in time and a finite element discretization in space. The number of equations in each time step is a modified Picard iteration scheme adopted to solve the non-linear set of equations. An idealized soil characteristic curve is used, and proves to be very efficient in enhancing the behavior of the numerical model, by eliminating irregularities encountered while solving standard non-linear equations.

GEO-FLOW: This program calculates flow lines individually, until leaving the A
flow domain, or arriving in elements with zero soil type (inactive elements), or A
when the flow Aime becomes larger than a pre-set Amaximum simulation Aime. The A
mobile water fraction parameter has an important effect here, since it represents A
the fraction of the groundwater that is considered to be effectively flowing A

GEO-FLUX: This program produces a continuous nodal flow velocity field,
the corresponding nodal flux vectors are also calculated. A lumped formulation
is used to solve the finite element equations obtained from post processing the A
simulated groundwater potential heads. A

Post-Processorsy

This group includes converter programs, which converts GEO-SWIM data sets to
files format compatible with third party software. GEO-GIS program converting
user-specified Alaska to Areadily Apost-processed AGISA compatible files. A Other A
programs in this category are GEO-TECP a converter for visualization with A
external software: Tecplot$T^M14$, and GEO-ACAD a CAD converter. A

7.4y VisualizationyTools

Another alternative for post processing GEO-SWIM data sets, is the use of the A
integrated modelling environment (Fig. 7.2), which has been developed recently. A
The motivation for developing such program is to provide a number of routines A
for visualizing the model input and results, which will enable easy verification A
and Acalibration of the model. A The same time, a significant environment for A
building easily groundwater models for practical and field studies is obtained, A
without Atoo much Aeffort, suchAthat Athe modeler will Aconcentrate Amainly A
using the model and interpretation of the obtained results A

The developed user interface can run under Windows operating systems, such A
as Windows 95/98/NT. It is a configurable windows menu-driven interface, with A

1 Tecplot!is!a!trademark!off!Antec!Engineering!Inc!
many significant characteristics and implemented packages. Some of these pack-A
ages give GUI support for previously described programs such as GEO-FCONA
and GEO-TECP. Other built-in packages include:A

**Cross-SectionalyCutter Packagey**

2D cross sectional cutter package for previewing cross-sectional or plan viewsA
ofAheAloadded model, thisApackageAhas theAability toAview modelAfeaturesA.e.g.A
mesh, soil types as separate layers of information, such that it works as a trueA
GeographicAInformationASystem. ATheAlistAofAInformationAoverlaysAincludes:AsoilA
types, simulatedAgroundwaterApotentials, pressures, andAfresh/saltwaterAinterface.A
MapsExport Packagey

This package enables exporting displayed maps to popular raster and vector formats (e.g., AutoCAD DXF, Windows Metafile, PostScript) which can be used in third-party CAD and engineering graphic programs.

![Diagram of a software interface](image)

Figure 7.3: Flow conditions package. The list shows boundary conditions used for the processed case study.

FlowyConditions Packagey

This package is a comprehensive GUI version of the previously described around-A water flow conditioner program (Fig. 7.3). The interface works in a similar way as for the batch program version, and enables to speed up this process for practical
Models Interface Package

This package is a collection of interactive dialog boxes and tools for collecting model input parameters e.g. density, soil types parameters, solver options (Fig. A7.4), time dependent parameters, etc. The collected data is translated to GEO-A SWIM specific format, as required by the models. This makes the model independent of that of the interface and enables the user to have support for other groundwater flow models if required, eliminating the need to write an-A specific code for other models. When directories of programs executables are defined in the Interface environment options as displayed in Fig. 7.5, several model versions can run from the same box.

GEO-SWIM to Tecplot Converter Package

This is a GUI program (Fig. A7.6) specifically developed to enable easy automatic production of high quality presentation graphics with the help of the Tecplot visualization software (Amtec Engineering Inc., 1996). Although, prior knowledge of how to manipulate different data sets for best visualization results...
with this powerful software is necessary, this program acts as a simple generator to build custom data sets which could be used directly from within Tecplot, thus transforming data formats from GEO-SWIM. The transformation is done smoothly in different ways, and a variety of data formats are supported; ASCII and Binary Tecplot file formats are supported, the latter version could be of choice when data sets are large (i.e., excessive number of nodes and/or transient data sets). This enables making files of high compression ratios, which results in a significant storage media saving and a smaller access time to the files. ASCII data sets are useful only for small size problems, such as theoretical ones, for which immediate checking of the results is faster through immediate viewing of the ASCII Tecplot file.
Figure 7.6: A tab view of the GEO-SWIM to Tecplot converter GUI package.
Chapterery8G

Conclusionsyandy
Recommendationsy

Contentsy

8.1 Conclusions ............................................. 185
8.2 Recommendations ..................................... 189

8.1 Conclusionsy

The results And Asignificant findings are Asummarized'in the Asub-sections Agiven A
below, following the order at which they were discussed earlier. A

8.1.1 Variably SaturatedyGroundwateryFlowyProblemsy

A computer program for modeling time dependent, and Athree-dimensional vari-A
ably saturated groundwater flow has been developed based on several numerical A
techniques, enabling great savings in computer time, and giving rise for running A
large problems of billion of unknowns, efficiently on cheap desktop workstation A
or PC’s. AThe model uses a Galerkin finite element approximation in space and A
a fully implicit finite difference time approximation with a mass lumped storage A
term. It features the following items A
- Automatic location of the moving water table boundary, based on the FUPA technique, was excessively cheap in comparison to the standard finite element method. The water table location is based on the changing nodal water status which are mapped from the relative water table position within each hexahedral element.

- Other non-linearities related to the difficulty in estimating the soil characteristic curves are tackled using nodal specific idealized water retention curves, by taking into consideration the saturated and the residual water contents for each soil type exclusively. This approach is found to be useful, and very attractive especially in combination with the solver.

- The implemented numerical solver shows to be very efficient, stable, and mass conservative. The embedded methods which largely contribute to such robustness are: the modified incomplete factorization preconditioner, with an M matrix transformation, the linear conjugate gradient solver, and the dynamic time marching scheme with automatic under-relaxation.

- Special attention was paid to complex and nonlinear boundary conditions, such as, seepage face, drainage, time varying heads, leakage and abstraction, these conditions are effectively implemented and tested on several application samples, thus providing a meaningful way to build applications with many levels of complexities.

The model is verified and validated using 4 test examples. Numerical results are compared to analytical, measured results, and other numerical techniques such as the moving mesh method. For all these test problems, a good agreement is obtained. The most important test validates the model by comparison with respect to laboratory measurements on a 3-D earth dam model (Baseghi and Desai, 1987). Numerical experiments were carried out in case of a rise, steady state, and drawdown of the free seepage flow, which allow for modeling heterogeneities.
8.1 Conclusions

represented by core samples materials. The model was able to predict accurately the moving water table location and seepage face extent.

8.1.2 Saltwater Intrusion Problems

Simulation of the saltwater encroachment in 3-D aquifer systems was studied based on a newly developed finite element based computer code: AGEO-SWIM. A model can handle difficult cases where two free and moving boundaries exist in the domain, e.g., in coastal unconfined aquifers. An iteratively based Ghyben-Herzberg approximation was implemented, and thus avoiding the solution of the coupled two-phase governing equations simultaneously. Instead, the saltwater zone was replaced with an equivalent freshwater zone having the same pressure distribution as an equivalent saltwater. Other achievements are as follows:

- The FUP numerical technique was generalized for the case of dual moving boundaries, especially explicit modifications for the shape of the idealized water retention curve, showing its dependence on the density difference ratio.
- The model uses the same numerical solver, and time marching scheme developed for the variably saturated groundwater flow model.

Five test problems are provided to support the numerical model. Various solution types (analytical, numerical, and experimental) are adopted for comparison purposes. The behavior of the saltwater encroachment has been studied in confined, unconfined and multilayer aquifers with different complexities and conditions. All these tests yield satisfactory results regarding the scope of the application, and model specific situation for each test.

The GEO-SWIM numerical code is applied to study saltwater encroachment in the coastal aquifer system of Martil situated in the northern part of Morocco. The
detailed three-dimensional geometry of the aquifer was reconstructed from the available data sets as a further step to pursue the modeling study. A calibrated steady state model shows that sensitivities are more relevant to the natural recharge, in comparison to other parameters such as hydraulic conductivity and salinity density. Afterwards, a long-term transient simulation was performed to reproduce actual and future situations, with the aim to analyze the impact of salinity from the Mediterranean Sea. It is concluded that without further control the saltwater interface would travel inland over considerable distances in the future.

8.1.3y Solute Transport Problems

In addition to the investigated three-dimensional groundwater flow class problems, numerical modeling of pollutant transport in aquifer systems by the conforming finite element method leads to large linear systems that are sparse and nonsymmetric. However, compared to the flow conductance matrix, it is observed that the global transport matrix arising from the discretization using hexahedral elements is not an M matrix, which is a very desirable property guaranteeing the numerical stability of the solution procedure. Therefore, several issues were investigated to overcome this difficulty, such as:

- The efficiency of preconditioned conjugate gradient like solvers such as GMRES and BICGSTAB

- Preconditioners as diagonal scaling, incomplete factorization, and modified incomplete factorization were tested. An M matrix transformation is posed which guarantees the existence of incomplete factorization, the most efficient preconditioner of all. The number of iterations, and CPU cost were used as a basis for these comparisons, also a series of benchmarks for all succeeded tests were performed for problems, which could be solved analytically, to ensure the accuracy of the obtained results.
8.2 Recommendations

Five representative test examples were selected as a basis for this comparison, and several strategies were adopted to conduct all different numerical simulations, based on the implemented preconditioners, stopping convergence criteria, and solution benchmarks when an exact solution was available.

It is found that BICGSTAB preconditioned by incomplete factorization performs well for all test problems. However, for difficult problems, such factorization is unlikely to exist, and the proposed M matrix transformation proves to be effective, leading to the guaranteed existence of a robust and efficient solver.

8.2y Recommendations

General guidelines for suggested future research are given herein. Throughout several possibilities and directives, we recommend either an extension of this work or another alternative and complementary developments, which may take profit from this thesis findings. We think at present that future modifications and/or extensions of the existing computer packages and models is manageable to success any of these suggestions.

- The developed FUP numerical technique for tree and moving interfaces, including the water table and the salt-freshwater interface was tested on problems. All constructed unstructured meshes. However, we should keep in mind that, this is not a restrictive condition with respect to the numerical approach since the FUP involves updating nodal storage and conductance with respect to the hexahedral element behavior. We expect the numerical scheme to work for problems with unstructured D-D problems using exclusively hexahedrals. This could prove to be useful for a particular class of problems where local refinement is advised.

- Under tidal conditions in the sea, the pressure distribution in the saltwater zone may become important, and hence it is advised that the coupled...
system of freshwater and saltwater equations are solved simultaneously in a time.

- Detailed probes are needed for the evaluation of the vertical flow components. An assumption about the moving interface flow, this will be very useful for a proper adjustment of the boundary conditions at wells which may withdraw saltwater.

- For the preconditioning methods, either for the flow or the transport problems, discussed in this work, the ability of each method to adapt to a particular hardware is dropped. In particular, the cost and ease of the implementation in parallel architectures are not discussed. Parallelization of computer codes is considered as one of the most exiting research topics in development. Today's software engineering market. A promising new generation of programming languages, optimized compilers, and operating systems will provide a solid platform for an integrated development of automatic parallel codes generation. However, the best results are obtained through a direct implementation at the computer code level.

- We are aware that for many situations where there exists a strong dependence of the water density upon the solute concentration, the solution of a coupled flow and transport problems is necessary. To solve effectively these class of problems, it is often assumed that spatial differences in the mass of pure water per unit volume due to variations of pressure are negligible in comparison. Particularly, in case of unsaturated transport a robust fully coupled solver may be based on the AFUP technique. In this context, and more specifically, in a case of a nearly advective flow, a second order accurate CVFE approach could be more feasible in solving solute transport equations. The latest developments are still not satisfactory for all domains, especially for an exahedral elements. Most of the existing formulations can only
for 2-D triangular elements), these new strategies should take into account the cost of the developed numerical engine, as in 3-D these are expected to be very high;

- Unstructured meshes are efficiently designed, if they effectively minimize the overall solver time for the same number of nodes. This is achieved in particular for a number of problems, using at least a combination of two elements (triangles and quadrilaterals in 2-D, and hexahedrals and tetra-Ahedrals in 3-D), the possible extension of the M matrix transformation to other kind of elements can provide challenging applications to be performed efficiently;

- The visual interface program presented in chapter 7 is an attempt to give a sufficient level of support for the models being developed. As the software development was academically and research oriented, the program can be used as a learning tool for postgraduate level training in finite element groundwater flow principles. To boost the practical use of the program, we strongly recommend its continuous development and support, for a wider range of models.
ENDd
REFERENCES


Bear, J. (1972), *Dynamics of fluids in porous media*, Elsevier, New York.A


ModifiedIncompleteFactorization Strategies, Preconditioned Conjugate Gradient Methods, pp. 1–16.A


REFERENCES


REFERENCES


Sbai, M.A. & F. De Smedt (1997b), 3d sharp interface finite element model for simulation of saltwater intrusion, Progress report 95-96 of avi-73 project, Laboratory of Hydrology, Free University Brussels.A


REFERENCEs


REFERENCES


Appendix AG

Analytical solutions for transient seepage

A.1 Linearization techniques

The Dupuit theory simplifies the governing groundwater flow equation, but does not remove all the nonlinearity. A common method of linearization seeks to escape from this difficulty by transforming the original governing equation to an equivalent form, which can be directly solved using standard applied mathematics. For the sake of simplicity, the technique will be briefly described. A case of a 1-D Boussinesq’s equation, with no sources/sinks

\[
\frac{\partial h}{\partial t} = \frac{k}{n_e} \frac{\partial}{\partial x} \left( \frac{\partial h}{\partial x} \right)
\]  

(A.1)

1. The simplest idea consists on approximating the dependent variable by an averaged value \( \overline{h} \) in the right hand side of equation A.1, such that

\[
\frac{\partial h}{\partial t} \approx \frac{k \overline{h} \frac{\partial^2 h}{\partial x^2}}{n_e \frac{\partial^2 \overline{h}}{\partial x^2}} = \frac{\overline{T}}{n_e} \frac{\partial^2 h}{\partial x^2}
\]  

(A.2)

where \( \overline{T} \) is the apparent transmissivity. A Equation A.2 is equivalent to the famous heat conduction PDE for which a known space of basic solutions is available, e.g. Airy’s function. However, the approximation used in the equation A.2 remains valid for small variations in the flow field, otherwise errors are significant.
2. A second method assumes \( \frac{k}{n_{ed}} \) as a constant, such that \( \text{Equation A.1} \) becomes linear with \( A^2.7 \).

It has been demonstrated that the second method is more accurate than the first one, because the linearization in \( A^2.4 \) preserves somewhat the nonlinear behavior of the original equation (Polubarinova-Kochina, 1962; Guo, 1997).

### A.2y Polubarinova-Kochina’s syseriesy functions

The analytical solution provided in Section A.3 is given in a form of an expansion series, whose first three terms are evaluated by Polubarinova-Kochina as

\[
    u_1(\eta) = \frac{2}{\sqrt{\pi}} \int_0^\eta e^{-t^2} dt = \text{erf}(\eta) \quad \text{(A.3)}
\]

\[
    u_2(\eta) = \frac{1}{\pi} (1 - e^{-2\eta^2}) - \frac{1}{\pi} e^{-\eta^2} u_{14} - \frac{1}{2\pi} e^{-\eta^2} u_{14} + \frac{1}{2\pi} e^{-2\eta^2} u_{14} \quad \text{(A.4)}
\]

\[
    u_3(\eta) = \frac{1}{\pi} e^{-\eta^2} + \frac{3}{4\pi} e^{-\eta^2} u_{24} - \frac{1}{\pi} e^{-\eta^2} u_{24} + \frac{3}{\pi} e^{-\eta^2} u_{14} - \frac{1}{\pi} e^{-2\eta^2} u_{14} - \frac{1}{\pi} e^{-2\eta^2} u_{14} \quad \text{(A.5)}
\]

Tabulated values of this functions for values of \( \eta \) from 0 to 1 are given in Table A.1 below. One may notice however that by truncating the power series expansion in equation (4.25) to the second term, the solution simplifies to

\[
    h(\eta, t) = A_0 (1 + \text{erf}(\eta)) + A_{14+} (h_{04} - h_{14}) \text{erf}(\eta) \quad \text{(A.6)}
\]

which is the solution of the linearized Boussinesq equation using the first method (Bear, 1972).
### Table A.1: Coefficients of series used in the Polubarinova-Kochina's analytical solution.

<table>
<thead>
<tr>
<th>$\eta$</th>
<th>$u_{14}$</th>
<th>$u_{24}$</th>
<th>$u_{34}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0A</td>
<td>0A</td>
<td>0A</td>
</tr>
<tr>
<td>0.1A</td>
<td>0.1125A</td>
<td>+0.0141A</td>
<td>-0.0039A</td>
</tr>
<tr>
<td>0.2A</td>
<td>0.2227A</td>
<td>+0.0160A</td>
<td>-0.0081A</td>
</tr>
<tr>
<td>0.3A</td>
<td>0.3286A</td>
<td>+0.0073A</td>
<td>-0.0090A</td>
</tr>
<tr>
<td>0.4A</td>
<td>0.4284A</td>
<td>-0.0092A</td>
<td>-0.0049A</td>
</tr>
<tr>
<td>0.5A</td>
<td>0.5205A</td>
<td>-0.0300A</td>
<td>+0.0039A</td>
</tr>
<tr>
<td>0.6A</td>
<td>0.6039A</td>
<td>-0.0519A</td>
<td>+0.0159A</td>
</tr>
<tr>
<td>0.7A</td>
<td>0.6778A</td>
<td>-0.0718A</td>
<td>+0.0280A</td>
</tr>
<tr>
<td>0.8A</td>
<td>0.7421A</td>
<td>-0.0874A</td>
<td>+0.0373A</td>
</tr>
<tr>
<td>0.9A</td>
<td>0.7969A</td>
<td>-0.0975A</td>
<td>+0.0422A</td>
</tr>
<tr>
<td>1.0A</td>
<td>0.8427A</td>
<td>-0.1017A</td>
<td>+0.0418A</td>
</tr>
<tr>
<td>1.1A</td>
<td>0.8802A</td>
<td>-0.1004A</td>
<td>+0.0368A</td>
</tr>
<tr>
<td>1.2A</td>
<td>0.9103A</td>
<td>-0.0946A</td>
<td>+0.0281A</td>
</tr>
<tr>
<td>1.3A</td>
<td>0.9340A</td>
<td>-0.0855A</td>
<td>+0.0194A</td>
</tr>
<tr>
<td>1.4A</td>
<td>0.9523A</td>
<td>-0.0744A</td>
<td>+0.0078A</td>
</tr>
<tr>
<td>1.5A</td>
<td>0.9661A</td>
<td>-0.0626A</td>
<td>-0.0011A</td>
</tr>
<tr>
<td>1.6A</td>
<td>0.9764A</td>
<td>-0.0510A</td>
<td>-0.0079A</td>
</tr>
<tr>
<td>1.7A</td>
<td>0.9838A</td>
<td>-0.0394A</td>
<td>-0.0125A</td>
</tr>
<tr>
<td>1.8A</td>
<td>0.9891A</td>
<td>-0.0310A</td>
<td>-0.0147A</td>
</tr>
<tr>
<td>1.9A</td>
<td>0.9928A</td>
<td>-0.0232A</td>
<td>-0.0151A</td>
</tr>
<tr>
<td>2.0A</td>
<td>0.9953A</td>
<td>-0.0169A</td>
<td>-0.0141A</td>
</tr>
<tr>
<td>2.5A</td>
<td>0.9996A</td>
<td>-0.0024A</td>
<td>-0.0047A</td>
</tr>
<tr>
<td>3.0A</td>
<td>0.9999A</td>
<td>-0.0002A</td>
<td>-0.0006A</td>
</tr>
<tr>
<td>3.5A</td>
<td>1A</td>
<td>-0.0000A</td>
<td>-0.0001A</td>
</tr>
<tr>
<td>4.0A</td>
<td>1A</td>
<td>-0.0000A</td>
<td>-0.0001A</td>
</tr>
</tbody>
</table>
Appendix By
Meuller’s method

B.1 Synopsisy

This method finds a zero of a real function $f(x) = 0$. An initial approximation to the zero must be given.

This uses an interpolating polynomial $P(x)$ of degree two, by using three approximate values for a root and approximates $f(x)$ near the root to be obtained. One of the roots for $P(x)$ is taken as the next approximate root of $f(x)$. This way iteration is continued. This algorithm has the following features:

- Derivatives of $f(x)$ are not required
- The function is evaluated only once at each iteration

B.2y Descriptiony

Let $\alpha$ be a root of $f(x)$ and let three values $\alpha_{i-24}, x_{i-14}$ and $\alpha_i$ be approximations to the root (See later explanation for initial values $\alpha_{14}, x_{24}$ and $\alpha_3$). According to Newton’s interpolation formula of degree two, $f(x)$ is approximated by using the three values described above as follows:

$$P(x) = \alpha + f[x_i, x_{i-1}](x - x_i) + f[x_i, x_{i-1}, x_{i-2}](x - x_i)(x - x_{i-1})$$  \hspace{1cm} (B.1)

A
where $Y_i = Y(x_i)$, and $Y[x_i, x_{i-1}]$ and $Y[x_i, x_{i-1}, x_{i-2}]$ are the first and the second order divided differences of $Y(x)$, respectively, and are defined as follows:

$$f[x_i, x_{i-1}] = \frac{f_i - f_{i-14}}{x_i - x_{i-14}}$$

$$f[x_i, x_{i-1}, x_{i-2}] = \frac{f[x_i, x_{i-1}] - f[x_{i-1}, x_{i-2}]}{x_i - x_{i-24}} \quad (B.2)$$

$P(x) = 0$ is then solved and the two roots are written as:

$$x = A x_i - \frac{2 f_i}{\omega \pm (\omega^{24} - 4 f_i f[x_i, x_{i-1}, x_{i-2}]^{1/2}$$

$$\omega = A f[x_i, x_{i-1}] + (x_i - x_{i-1}) f[x_{i-1}, x_{i-2}] \quad (B.3)$$

Of these two roots for $AP(x) = 0$, the root corresponding to the larger absolute value of the denominator in the second term of equation (B.3) is chosen as the next iteration value $x_{i+1}$. This means that $x_{i+1}$ is a root closer to $x_i$. In equation (B.1), if the term of $x^{24}$ is null, i.e., if $Y[x_i, x_{i-1}, x_{i-2}] = 0$, the following equation is used in place of using equation (B.3).

$$x = A x_i - \frac{f_i}{f[x_i, x_{i-1}]}$$

$$= A x_i - \frac{x_i - x_{i-14} f_i}{f_i - f_{i-14}} \quad (B.4)$$

This is the secant method.

In equation (B.1) also, if both terms $x^4$ and $x^{24}$ are null, $P(x)$ reduces to a constant and the algorithm fails. (See later explanation.)

**B.3y Algorithm**

- Initial values $x_{14}$, $x_{24}$ and $x_{34}$

  The three initial values are set as follows: Let $x$ be an initial value set by the user in the input parameter $x_k$.

  When $x_k \neq 0$

  $x_{14} = 0.9x$
\[ x_{24} = 1.1x \]
\[ x_{34} = 0 \]

When \( x \) = 0,
\[ x_{14} = A - 1.0A \]
\[ x_{24} = 1.0A \]
\[ x_{34} = 0.0A \]

- When \( f(x_{i-2}) = \mathcal{A}(x_{i-1}) = \mathcal{A}(x) \)

This corresponds to the case when both terms \( x \) and \( x^{24} \) in the equation in sub. 4.3 are null, so Muller’s method cannot be continued.

The subroutine changes \( x_{i-2} \), \( x_{i} \), and \( x \) and tries to get out of this situation by setting
\[ x'_{i-24} = (1 + p^n)x_{i-24} \]
\[ x'_{i-14} = (1 + p^n)x_{i-14} \]
\[ x' = (1 + p^n)x \]

where \( p = A - u^{-1/10} \), \( u \) is the unit round off and \( n \) is the count of changes.

Muller’s method is continued by using \( x'_{i-24} \), \( x'_{i-1} \), and \( A' \). When more than five changes are performed the subroutine terminate unsuccessfully.

B.4y Convergence criteria

The following two criteria are used.

**Criteria I.** When the approximate root \( x \) satisfies \( \mathcal{A}(x_i) \leq \gamma \), the root \( x \) is taken as the approximate root. When the root is a multiple \( x_i \) root or very close to another root, \( \gamma \) must be set sufficiently large. If \( 0 \leq \gamma < u \), the subroutine resets \( \gamma = A \).