A Semi-Analytical Particle Tracking Algorithm for Arbitrary Unstructured Grids

by

Muhammad Ramadhan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Civil Engineering

Waterloo, Ontario, Canada, 2015

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public

ABSTRACT

Semi-analytical particle tracking methods are broadly used in determining advective transport within groundwater flow models. The Pollock method is currently implemented in the USGS model MODPATH as a semi-analytical particle tracking algorithm for structured grids in MODFLOW. However, this method is not valid for grid cells with irregular geometry, and new methods are needed to effectively represent local flow fields within arbitrarily shaped grid cells for tracking particles. There are currently two particle tracking methods that can create pathlines in arbitrary unstructured grids: 1) the SSP&A method and the method introduced by Painter *et al.* (2012). The SSP&A method uses local universal kriging interpolation of a MODFLOW hydraulic head solution in order to determine the particle velocity using head changes. The method by Painter *et al.* uses unconstrained least squares method on interior cells and constrained least squares method on boundary cells to calculate the particle velocity. However, these methods generate local interpolation-based velocity fields that do not respect the mass balance of the cell. Here, we develop an accurate semi-analytical particle tracking method for the new MODFLOW-USG, which uses a control volume finite difference (CVFD) approach for arbitrary unstructured grids.

This particle tracking method (the Waterloo method) is based on a locally analytical Taylor Series reconstruction of the local velocity, which can be used to generate pathlines of particle movement using an Euler or high-order Runge-Kutta scheme. The approach is valid for any cell geometry and may be used for both steady-state and transient simulations, while still respecting the mass balance of the cell. By following the particle path from cell to cell, this method can trace the movement of a particle through any arbitrary unstructured multidimensional flow field generated from a CVFD model.

The pathlines generated by the Waterloo method in rectilinear cells are compared to the pathlines by the Pollock method. The results show less than 1% differences in terms of the spatial and temporal distributions of the particle exit locations. The robustness of the Waterloo method is supported by its ability to handle complex flow field within a model that contains extreme heterogeneity and zero-hydraulic conductivity zones. The approach also bypasses the “weak sink” problem by explicitly representing the local flow fields near wells or controlled by distributed leakage/recharge. Furthermore, this semi-analytical method is capable of handling multiple, off-center wells located anywhere within one cell.

Acknowledgements

I would like to thank my advisor Dr. James R. Craig for the guidance and support given to me for the past two years. From S.S. Papadopulos & Associates, Inc., I would like to thank Chris Muffels who gave me an enormous help with understanding techniques regarding C++ and FORTRAN, insight regarding various methods for my research, and the development of mod-PATH3DU. I wish to thank my fellow graduate students Ayman Khedr and Mashrur Chowdhury for their support.

Finally, I would like to thank my family for their support and sacrifice that they have made to make all of this possible

Dedication

I would like to dedicate this thesis to my parents and brothers who supported me throughout my entire program, as well as Aya Goto for always believing in me and for encouraging me since the beginning of my Master's program.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol	Description
A	Cell area [L^2]
A_x, A_y, A_z	A constant that correspond to the component of the velocity gradient within the cell, used in Pollock method [1/T]
a_n	The Taylor series coefficients of the cell
b^n	Saturated thickness of the cell at the start of each MODFLOW time step[L]
C	Dissolved concentration [M/L^3]
D_{ij}	Hydrodynamic dispersion coefficient tensor [L^2/T]
h	Hydraulic Head [L]
K	Hydraulic conductivity of the aquifer/cell [L/T]
L_i	Length of each side of the cell [L]
M	Total number of control points of the cell
N	Order of approximation of the cell
n_{faces}	Number of faces of the cell (not including top and bottom face)
P	Cell perimeter [L]
Q_i	The time-averaged flow rates in through the sides of the cell obtained from MODFLOW cell-by-cell flow files; negative outward, positive inward (This parameter must be arranged in accordance to the cell vertices) [L^3/T]
Q_{spec}^i	The specified normal integrated discharge [L^2/T]
Q_{vert}	Cumulative vertical inflows that include fluxes from the top and bottom of the cell, as well as the storage term, also referred to as μ. (Negative upward, positive downward) [L^3/T]
Q_{well}	The well flow rates (Negative for pumping well, positive for injecting well) [L^3/T]
Q_x, Q_y	Horizontal integrated discharge [L^2/T]
Q_η^{cell}	The normalized cell fluxes, or integrated cell discharge [L^2/T]
Q_η^{vert}	The normalized vertical fluxes, or integrated vertical discharge [L^2/T]
Q_η^{well}	The normalized well fluxes, or integrated well discharge [L^2/T]
Q_η^{Taylor}	The normalized Taylor series discharge [L^2/T]

q_{top}	Fluxes through the top of the cell [L/T]
q_{bot}	Fluxes through the bottom of the cell [L/T]
R	Maximum radius between the cell center and each cell vertex [L]
R_n	Chemical reaction term [M/L^{3}T]
S	S [-] is either equal to $S_s b$ (the storativity) or S_y (the specific yield), depending upon whether the cell is confined or unconfined.
S_s	The specific storage of the cell (for confined aquifer) [1/L]
S_y	The specific yield of the cell (for unconfined aquifer) [-]
S_{ctrl}	The side control parameter that specifies how far along the perimeter of the cell each point is
t	Time; change in time could refer to MODFLOW time step (t_{MOD}) or tracking time step [T]
v_p	The current particle velocity, used in Pollock method [L/T]
$\lvert v_n \rvert$	The absolute value of the current complex horizontal velocity of the particle such that $\lvert v_n \rvert = \sqrt{{v_x}^2 + {v_y}^2}$ [L/T]
x_p, y_p, z_p	The particle current location in x-, y-, and z-coordinate [L]
$[x_i \quad y_i]$	Vertex coordinates of the cell in matrix form, arranged clockwise
$[x_c \quad y_c]$	The Cartesian coordinate of the cell's center
Z	The local complex coordinate: $\frac{z-z_c}{R}$
Z_{ctrl}	The local complex side control points specify the localized z_{ctrl}
Z_w	The complex coordinate of well in a cell
z	Mostly refer to complex coordinate ($x + iy$), unless specified otherwise (e.g. it is referred to as vertical coordinate a few time in this thesis)
z_b	Cell bottom elevation [L]
z_c	The complex coordinate of the cell center $[x_c \quad y_c]$
z_{ctrl}	The complex side control points specify the complex coordinate of S_{ctrl}, or also referred to as global complex control points
z_i	Cell vertices $[x_i \quad y_i]$ in complex coordinate z

$\bar{z}$	The complex conjugate of complex coordinate z
α_i	The angle for each side of the cell, calculated in radian
$\in$	The normalized absolute average error within the cell
θ	Porosity of the cell/aquifer [-]
Φ	Discharge potential function
Ψ	Stream function
Ψ_i^{Taylor}	The Taylor series stream function, necessary for calculating Taylor series coefficients
$\Psi_{i,j}^{unit}$	The unit stream function, determined at each control point along the boundary. Necessary for calculating the Taylor series coefficient
Ω	Complex potential function
$\mathrm{Re}(\ldots)$	The real part of a complex number
$\mathrm{Im}(\ldots)$	The imaginary part of a complex number

Chapter 1

Introduction

In the past few decades, human advancements in agriculture, industry, and society have had major negative impacts on hydrological systems. These impacts include aquifer depletion, ecosystem degradation, and reduction of water quality (Zhou, 2011). To generate sustainable water resources development strategies, decision-makers need to adequately gather information regarding groundwater, surface water, and their interactions. Groundwater modeling has played an important role in the management and development of groundwater resources, and in predicting groundwater behavior. Groundwater models can be used as interpretative tools for understanding groundwater system dynamics and flow patterns as well as predictive tools for estimating the impacts of human activities on subsurface systems (Zhou, 2011). For years, groundwater scientists and engineers have been developing improved groundwater models in order to better simulate and predict groundwater conditions.

Groundwater models are based upon the mathematical solution to a mass balance statement that is composed of continuity of a mass balance on water and Darcy's Law. These mathematical representations allow hydrologists to, for example, calculate water table height or identify advective particle pathlines. These pathlines are defined as the path a small parcel of groundwater would trace as it moves through the subsurface over a given period of time.

The accuracy of a groundwater model depends highly on the assumptions made during model construction and the variables chosen to reflect reality. Due to these limitations, computational groundwater models can only provide an approximation and are not able to perfectly simulate subsurface systems (Bredehoeft, 2005). The uncertainties associated with groundwater modeling include the deficient observational accuracy of the subsurface parameters, such as average horizontal and vertical

flow of water in the aquifer, hydraulic conductivities (which can only be inferred from flow and pressure data), saturated thickness of the aquifer, and porosity of the subsurface media.

There have been numerous groundwater modeling software products developed over the years to improve the accuracy of the groundwater model to better represent subsurface systems. MODFLOW (Harbaugh, 2005) is a three-dimensional finite-difference groundwater model created by the U.S. Geological Survey (USGS) and is one of the most commonly used software products for simulating aquifer systems. The standard version of MODFLOW-2005 uses rectangular-based structured grids for discretization of the flow domain, where grid cell sides align with the Cartesian coordinate system and grid spacing is uniform in the cardinal directions. In two-dimensions, each cell must be a square or a rectangle, with each side of the cell touching one other cell at most. In order to increase resolution of a particular zone in the conceptual model, the resolution of the grids must be refined in the horizontal and vertical direction of the zone of interest, which results in higher computational costs than necessary. In 2012, the USGS developed the new MODFLOW-USG (Panday *et al*. 2013), a completely modified implementation of MODFLOW, which can handle both structured and unstructured grids including nested grids, quadtree grids, Voronoi tessellations, and triangular-based grids. Unstructured grids are more desirable due to their ability to capture features like rivers and wells in a much more detailed manner and directly support variable resolutions. Although unstructured grids are more complex than structured grids, they are able to be refined only near the desired features within the conceptual model without significantly increasing the computational time of the simulation. MODFLOW-USG uses a Control Volume Finite Difference (CVFD) formulation that still allows mass conservation at a local scale, which provides accurate estimates of groundwater heads and flows around the desired features within the conceptual model.

One of the objectives for which groundwater models are developed is to determine the flow path of a contaminant. Modeling advective transport of particles is important to predict whether or not a contaminant will travel from a source to nearby rivers, wells, or other places, and how long it will take for the contaminant to travel to those places. The standard advective particle tracking algorithm for

MODFLOW-2005, MODPATH, uses the Pollock method (Pollock, 2012), which is only valid for rectangular-based structured grids. The method calculates particle pathlines using linear interpolation of the velocities sourced from the cell-by-cell flows generated by MODFLOW. MODPATH can analytically calculate when and where a particle may end up within a cell or exit the cell using only the geometry of the cell and groundwater velocities in through the sides of the cell. Recently, the method has been extended to support nested Cartesian grids (Pollock, 2015) by splitting the cell that has one side touching multiple cells into similar sub-cells. However, the limitation to rectilinear cells remains. The new extension to MODPATH still cannot handle other unstructured grid configurations. However, for a number of reasons, non-rectilinear grid configurations are desirable. For instance, Voronoi cells can handle complex geometries and boundaries with proficient refinement and high resolution around rivers, wells, and other boundary conditions.

A new semi-analytical particle tracking algorithm that can handle any type of structured and unstructured grid with arbitrary cell geometry in the x- and y-directions is here introduced. It presumes 1:1 cell connections in the vertical direction, i.e., each model layer has the same two-dimensional unstructured grid configuration, but can handle rectilinear (standard or quad-based), Voronoi, hexagonal, or triangular grids. This method is incorporated into the mod-PATH3DU software developed by SSP&A inc. (Muffels et al, 2014) for particle tracking in groundwater models that use arbitrary unstructured grids. This algorithm assumes independent cells, i.e., the reconstruction of the velocity field in one cell is independent of what is occurring in other cells and requires only the cell boundary fluxes and the geometry of that particular cell. This algorithm will help hydrogeologists in producing accurate particle pathlines for a conceptual model at low computational cost.

1.1 Problem Statement

Groundwater models are deterministic mathematical models that require the solution of partial differential equations. These models are based on conservation of mass, momentum, energy, and consituitive models

such as Darcy's Law (Konikow, 1992). Tools such as MODFLOW-USG and other unstructured groundwater flow simulators based on finite volume method or controlled volume finite element method (e.g., HydroGeoSphere (Therrien *et al.*, 2010)) allow hydrologists to create both structured and unstructured grids that are more refined to capture features like rivers and wells in more detail. However, there are few algorithms which can be used to effectively reconstruct sub-cell velocity fields for particle tracking on the unstructured grids, none of which fully respect mass balance in the cell. A new method that can effectively represent local flow fields in arbitrary unstructured grids is desirable. This new method should have the capability to accurately estimate velocities at the sub-grid scale while respecting mass balance.

1.2 Thesis Objective

The objective of this thesis is to develop and test of a new sub-cell velocity reconstruction method for saturated groundwater flow models (called the Waterloo Method), which can handle any type of structured and unstructured grid and calculate the velocity at any point within a cell semi-analytically. The method can handle heterogeneity, vertical anisotropy, and transient variation in saturated thickness, and uses either Euler or High-Order Runge-Kutta schemes for pathline integration. This method is tested extensively and is applied to a number of challenging test cases. Furthermore, this thesis will attempt to show how the algorithm may handle arbitrary geometry, transient cases, the presence of wells, and is consistent with the Pollock method for rectilinear grids. Attempts will be made in this thesis to show the ability of Waterloo Method to handle challenging situations that other particle tracking algorithms are not able to sufficiently resolve, and effort will be spent to explain in depth the reasoning and methodology behind the development of this algorithm. Additionally, the differences between the particle pathlines generated using this method and using other particle tracking method are presented in this thesis in order to show the advantages of this method in comparison to the other methods.

1.3 Scope of Research

This thesis includes only the development of semi-analytical particle tracking algorithm for structured and unstructured grids. This algorithm is inserted into the mod-PATH3DU software (Muffels *et al.*, 2014) as a particle tracking algorithm for the new MODFLOW-USG, where the main input files required for this software must come from MODFLOW-USG. Every result from this algorithm is produced with the assumption that the input files provided for mod-PATH3DU, such as cell geometries and cell-by-cell flows are accurate; i.e., the discrete flow solution has converged and preserve mass balance. Extensions to grids which are non-uniform in vertical direction, problems with lateral material anisotropy, and the handling of connected linear networks are considered beyond the scope of this thesis.

1.4 Thesis Outline

This thesis is outlined as follows:

Chapter 1 provides the reasoning behind the development of this algorithm. It briefly discusses how groundwater modeling is crucial for use by hydrologists to represent the subsurface system. Limitations regarding the current particle tracking algorithm for MODFLOW-USG are discussed. The objective and the scope of the research surrounding the development of this semi-analytical particle tracking algorithm is included.

Chapter 2 presents background information about groundwater modeling, including the history of groundwater modeling software and particle tracking methods. This chapter discusses different approaches available for modeling the subsurface system and in generating particle pathlines for use in water resources management. A brief overview regarding the advantages and limitations of existing methods is also included.

Chapter 3 provides detailed information on the new methodology used to reconstruct local flow fields and generate accurate particle pathlines in any type of structured and unstructured grid. The method includes the means of solving the local sub-cell governing equation in order to calculate the velocity distributions within a cell. Chapter 3 also discusses related assumptions used by the Waterloo Method and the methods for quantifying algorithm errors.

Chapter 4 presents the results of a number of tests used to assess the algorithm performance and discusses them in length. This chapter demonstrates the capability of the Waterloo Method to handle arbitrary unstructured grids. Comparisons between the outcomes of this method and other methods are provided; methods are contrasted in terms of both accuracy and computational costs. This chapter illustrates the sensitivity of the results if the preliminary assumptions are altered. The errors in this algorithm are assessed for spatial and temporal discretization effects and for changing model parameters, such as cell control points and order of approximation. The ability of this method to handle multiple wells within a cell that are not located at the center of the cell is presented. Chapter 4 also includes an investigation of different type of sink sources and how this method handles the "weak" sink problems. The method is deployed for challenging multi-layer models.

Chapter 5 summarizes the main conclusions and recommendations for this research. Limitations regarding the work that have been identified so far are also discussed.

Chapter 2

Background

2.1 Groundwater Modeling

2.1.1 Overview

The increased uses of groundwater resources throughout the world for development purposes require an expansion of our knowledge of groundwater systems and their interactions with the ecosystem. Hydrologists often use groundwater models as a tool to increase their understanding of a groundwater system and to predict its potential future responses. Likewise, groundwater models are used by regulators to support protection of the subsurface, such as during site remediation and well construction, in order to determine the standards for the quantity and quality of groundwater resources. One of the most common applications of groundwater models is the prediction of particle pathlines, such as advective particle tracking used to monitor contaminant transport or to delineate capture zones for wells. Information regarding particle movement in the subsurface system may help practitioners to assess whether or not contaminant can travel to nearby wells or rivers. The quality of these predicted pathlines depends upon 1) the validity of the conceptual model, 2) the accuracy of the numerical model, and 3) the accuracy of the tracking scheme.

One of the first steps in groundwater flow modeling is to establish the conceptual model of the subsurface system and to determine the modeling purpose. The definition of a conceptual model is the representation of the subsurface system in terms of physical hydrogeological features and their hydraulics (Wels *et al.,* 2012). By starting with an adequate groundwater conceptual model, hydrologists may increase the predictive performance of their numerical groundwater models (Yang, 2009). A good conceptual model must satisfactorily reflect the user's concept of the natural system, accurately describe the essential processes affecting groundwater flow, and be able to provide acceptable results in

comparison to observation data when translated into a mathematical model. Conceptual models for groundwater flow typically include the hydrogeologic setting of the field system, model domain, model boundaries, hydrostratigraphic units and hydraulic properties, and key components of groundwater-surface water interaction. A desirable conceptual model should be developed with a sufficient level of simplicity while still retaining its complexity to be able to represent the physical elements and hydraulic behavior of the systems. Simplifying assumptions are often necessary because there are rarely sufficient data regarding the groundwater parameters in order to completely reconstruct the groundwater system. As the users' understanding of the groundwater system improves, the conceptual model should be updated continually by incorporating any new observational information. A conceptual or mathematical model may be inadequate due to incomplete information of the model, incorrect assumptions regarding the parameters, ignorance of key processes influencing the model, and poor understanding of physical and chemical processes in the field subsurface system.

In order to be used for prediction, the conceptual model must typically be translated into a mathematical model, which may be either a simple analytic model or a more complex numerical model. Analytical solutions to investigate groundwater flow were first used in early 1960s to simulate hypothetical small drainage basins (Toth, 1963), helping to support key ideas about the nature of local and regional groundwater flow. Numerical models were exercised for the first time in late 1960s, e.g., Freeze and Witherspoon (1966), who used such models to simulate steady state regional flow patterns in layered aquifer system. Unlike analytical solutions, these numerical models were able to simulate three-dimensional groundwater flow under heterogeneous and anisotropic conditions. As part of the Regional Aquifer System Analysis (RASA) program in 1978, the application of groundwater flow models to large scale aquifer system was regularly performed by the U.S. Geological Survey (Sun and Johnson, 1994). During this program, 25 important American aquifer systems were intensively studied in order to further understand the regional aquifer system's responses to natural and human stresses, and the compilation of a national groundwater atlas. The RASA program used computer-based numerical groundwater flow models such as USGS 3D finite difference model and USGS MODFLOW (McDonald and Harbaugh,

1988) to simulate the effects of groundwater development and land use changes. Since its original publication in 1984, USGS MODFLOW has been extensively used as a three-dimensional finite-difference groundwater model. It has since been joined by other numerical codes for simulating groundwater, each with its own capability and advantages, to simulate groundwater flow, transport, and geochemical reactions (e.g. MODFLOW (Harbaugh, 2005), SEAWAT (Guo and Langevin, 2002), FEFLOW (Diersch, 2013), HydroGeoSphere (Therrien *et al*., 2010)), or variably-saturated flow and transport (e.g. SUTRA (Voss and Provost, 2008), R-UNSAT (Lahvis and Baehr, 1997)).

2.1.2 Particle Tracking & Contaminant Transport

In determining well capture zones, hydraulic containment, and simulating the advective part of contaminant transport, it is important to conduct advective particle tracking simulation. Particle tracking is essentially tracing the position of an advected parcel of fluid through a fluid velocity field over time. The definition of a particle in this thesis is anything from a single molecule to a parcel of water; there is no parameter associated with the particles in the model other than its coordinate location. By releasing hypothetical particles in groundwater simulation, the movement of the particles and their end point can be calculated from groundwater velocities. In some cases, it may be desirable to track hundreds to thousands of hypothetical particles. Particle tracking can determine where the particle will end up after a specific period of time by conducting forwards-in-time simulation, or determine the particle original point by conducting backwards-in-time simulation.

The primary transport mechanism of chemicals in high permeability aquifers is advection, which is fluid transport due to the fluid's bulk motion. Many applications have been developed to only account for this advective component of transport (Konikow *et al.,* 1996), and thus assume the dispersion and diffusion mechanism to be negligible. Advective flow velocities are obtained from Darcy's Law:

$$v_i = -\frac{K}{\theta}\frac{\partial h}{\partial x_i} \tag{2.1}$$

Where v_i is advective velocity in the ith direction [L/T], K is hydraulic conductivity of the aquifer [L/T], θ is porosity [-], and $\frac{\partial h}{\partial x_i}$ is the hydraulic gradient in the ith direction, a change in hydraulic head per change in length [L/L]. The general approach for advective velocity in particle tracking is that given a velocity field and current particle location, the future location of the particle can be calculated over a time period via integration.

Groundwater contamination may occur due to substance discharge from industries, urban activities, agricultural chemicals, or landfill leachate. This contamination may negatively impact the communities that use groundwater as their primary source of water. The migration of chemicals that enters the subsurface is controlled by three key transport mechanisms: advection, dispersion/diffusion, and retardation (Bedient *et al*., 1994). Dispersion is the spreading of molecules due to different flow paths of water. Diffusion is the movement of particles due to random molecular motion that is caused by thermal energy of the solute. Retardation is the act that causes particle movements to be delayed due to the difference in aquifer velocity. The partial differential equation that describes the fate of a contaminant in the three-dimensional, transient groundwater system is as follows (Zheng and Wang, 1999):

$$\frac{\partial(\theta C)}{\partial t} = \frac{\partial}{\partial x_i}\left(\theta D_{ij}\frac{\partial C}{\partial x_j}\right) - \frac{\partial}{\partial x_i}(\theta v_i C) + \sum R_n \tag{2.2}$$

Where C is the dissolved concentration [M/L^3], θ is the porosity of the subsurface medium [-], t is time [T], x_i is the distance along the respective Cartesian coordinate axis [L], D_{ij} is the hydrodynamic dispersion coefficient tensor [L^2/T], v_i is the seepage or linear pore water velocity related to the specific discharge or Darcy flux [L/T] that may be calculated from Equation 2.1, and $\sum R_n$ is the chemical reaction term [M/L^{3}T].

The second term on the right hand side of Equation 2.2 represents the influence of advection, and is dominant when the hydrodynamic dispersion coefficient tensor is small. By conducting an adequate particle tracking simulations, which only simulate advective transport, users may predict with some degree of confidence whether or not a contaminant source is likely to enter important features such as

rivers and wells effectively by solving a simplified version of Equation 2.2. Additionally, multiple numerical methods for solving Equation 2.2 (e.g., the method of characteristics) rely upon accurate particle tracking.

2.2 MODFLOW

MODFLOW (originally developed by the USGS in 1984) (McDonald and Harbaugh, 1984) is one of the most commonly used groundwater flow models. There have been four main versions of MODFLOW since 1984: MODFLOW-88, MODFLOW-96, MODFLOW-2000, and MODFLOW-2005.The first two major releases of MODFLOW, MODFLOW-88 and -96, were initially conceived solely as groundwater flow simulation codes. MODFLOW-2000 was later developed as an expansion to improve the calculations regarding the groundwater flow transport and parameter estimation. The core process in MODFLOW-2000 is the Groundwater Flow (GWF) Process, which is the part of the code that solves groundwater flow equation (McDonald and Harbaugh, 2003). The latest major release using structured grids is MODFLOW-2005, which improved the management of internal data and provided a means to define multiple grids in a single MODFLOW simulation. This version of MODFLOW is the most stable and well-tested version of the code, and is presently the primary MODFLOW version distributed by the USGS and used in practice. The basic packages in the current version of MODFLOW include block-centered flow, layer-property flow, horizontal flow barrier, river, recharge, well, drain, evapotranspiration, general-head boundary, strongly implicit procedure, preconditioned conjugate-gradient, and direct solver packages. Each of these packages handles different boundary conditions and/or contributes to the solving the resultant discrete system of equations.

The governing equation for three-dimensional groundwater flow through porous media with constant density can be described as follows (Rushton and Redshaw, 1979):

$$\frac{\partial}{\partial x}\left(K_{xx}\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_{yy}\frac{\partial h}{\partial y}\right)+\frac{\partial}{\partial z}\left(K_{zz}\frac{\partial h}{\partial z}\right)+W=S_s\frac{\partial h}{\partial t} \tag{2.3}$$

Where K_{xx} , K_{yy} , and K_{zz} are the hydraulic conductivity values in $x, y,$ and z direction [L/T]; W represents the volumetric flow rate for sink/source terms per unit volume of aquifer [1/T]; S_s is the specific storage of the porous media [1/L]; and t is time [T]. Equation 2.3 is the governing equation solved by MODFLOW, and can represent groundwater flow through heterogeneous, anisotropic media, with steady-state or transient flow, which may be influenced by sink or source terms. Under the conditions or assumptions where the media is homogeneous and isotropic ($K_{xx} = K_{yy} = K_{zz}$) with steady-state flow ($\partial h/\partial t = 0$) and no sink or source terms ($W = 0$), the left hand side of Equation 2.3 becomes the three-dimensional Laplace's equation.

Although MODFLOW is considered an international standard for simulating and predicting groundwater conditions, there are still some limitations of the program. For instance, the water in the model must have a constant density, dynamic viscosity, and temperature throughout the modeling domain. MODFLOW-2005 currently cannot simulate complex geological features or simulate steep hydraulic gradients; the rewetting and drying of cells is also an issue (Harbaugh, 2005). Furthermore, MODFLOW-2005 still cannot handle non-rectilinear-based grids. However, these limitations can be addressed by using specialized versions of MODFLOW, including: 1) MODFLOW-NWT, a standalone program to handle cases regarding drying and rewetting nonlinearities of the unconfined groundwater flow equation, 2) MODFLOW-LGR, a 3D finite-difference groundwater flow model with local grid refinement, and 3) MODFLOW-USG, the unstructured grid version of MODFLOW that is based on an underlying CVFD formulation, and can use arbitrary unstructured grids. The primary focus of this thesis is the particle tracking algorithm for MODFLOW-USG.

MODFLOW-USG was first released in 2012 as a completely rewritten version of MODFLOW (Panday *et al.*, 2013). The main purpose for the development of this MODFLOW version is to support a wide variety of structured and unstructured grids, such as nested grids, triangular-based grids, quad-based grids, and Voronoi Tessellations. This new flexibility in grid design is used to further focus resolution along rivers and around wells, and to better represent hydrostratigraphic units by subdiscretizing

individual layers. The core process in MODFLOW-USG remains the GWF Process, but extended to allow connections between a cell and an arbitrary number of other adjacent cells. One other new process within the MODFLOW-USG is the Connected Linear Network (CLN) Process, which is used to simulate the effects of multi-node wells and tile drains. Additional important packages incorporated into MODFLOW-USG include the Ghost Node Correction (GNC) Package that uses interpolated heads in the flow calculation between adjacent connected cells, and the Sparse Matrix Solver (SMS) Package to solve the system of flow equation, which is more complicated for the unstructured grids. Furthermore, MODFLOW-USG contains the Newton-Raphson formulation that is based on the formulation in MODFLOW-NWT as an optional process to handle problems regarding the drying and rewetting of cells by using the SMS Package. Current practical limitations of MODFLOW-USG include its incapability to support all of the packages that other MODFLOW versions do. These limitations may be addressed in the next version.

2.2.1 Structured Grids

There are various types of grids or cells that can be considered structured grids, in which the number of cell-connections is the same for all cells except along the boundaries. Figure 2.1 shows examples of structured grids presented in two-dimensional plan view. However, the structured grids discussed in this thesis are defined only as MODFLOW structured grids, which are the rectilinear-based grids; Grids A and B in the figure. Grids C through G in Figure 2.1 are considered non-rectilinear structured grids in groundwater flow modeling, which MODFLOW-2005 does not support. Rectilinear structured grids for discretization of the flow domain in standard MODFLOW require grid cell sides to align with the Cartesian coordinate system and grid spacing to be uniform in the cardinal directions. A conceptual model that consists of only structured grids is considered simple in terms of data management. However, its limitations include the inability to focus resolution along important features in the conceptual models such as rivers and wells. In order to increase the accuracy around those features, the size of the grids must

be refined in the both horizontal and vertical direction of the zones around the feature, hence impacting resolution across the whole modeled domain.

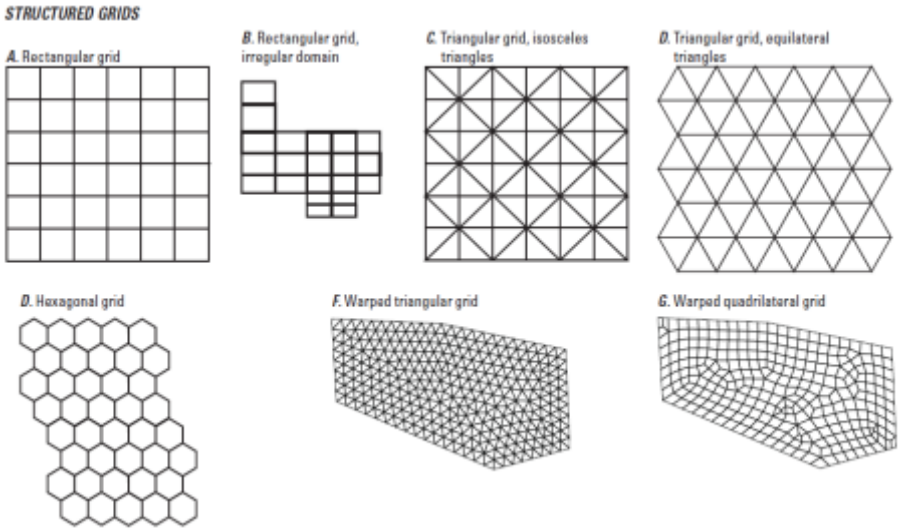


Figure 2.1: Examples of Different Types of Structured Grids (Figure from Panday *et al*., 2013)

The mesh refinement will result in higher computational time than necessary, because unimportant areas may be finely discretized. Figure 2.2 illustrates grid refinement around an important feature with a rectilinear grid. If, for example, the desired feature within the conceptual model is marked with the red dot, then it is more likely that the area around that feature will be refined in order to increase the accuracy of that area. The rows and columns of the whole conceptual model need to be refined as well in order to meet the restriction of the MODFLOW rectilinear structured grids, where each side of the cell must be touching one other cell at most, except along the boundary of the model.

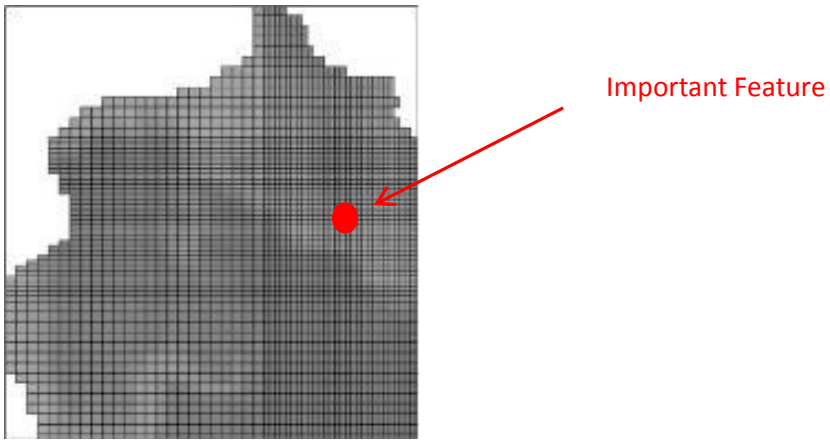


Figure 2.2: Non-Uniform Rectilinear Structured Grids of a Conceptual Model (Figure from Hesch, 2013)

2.2.2 Unstructured Grids

Due to the particular application of finite volume formulation in MODFLOW-USG, the construction of unstructured grids to represent the groundwater flow conceptual model becomes possible. This flexibility allows the grid cell geometry to be any shape: rectangular, triangular, nested, Voronoi Tessellations, or any combination thereof. Figure 2.3 depicts the unstructured grids with variable number of connections for each cell throughout the conceptual model. The unstructured geometry in MODFLOW-USG only apply in x and y direction; in cross-sectional view, the top and bottom cell faces are horizontal and the side faces are vertical, and thus the cells are prismatic in the vertical direction. However, since each side of the cells is not restricted to only be connected to one cell, the top and bottom side of each cell can also connect to multiple cells; nested grids may therefore be applied in the vertical direction.

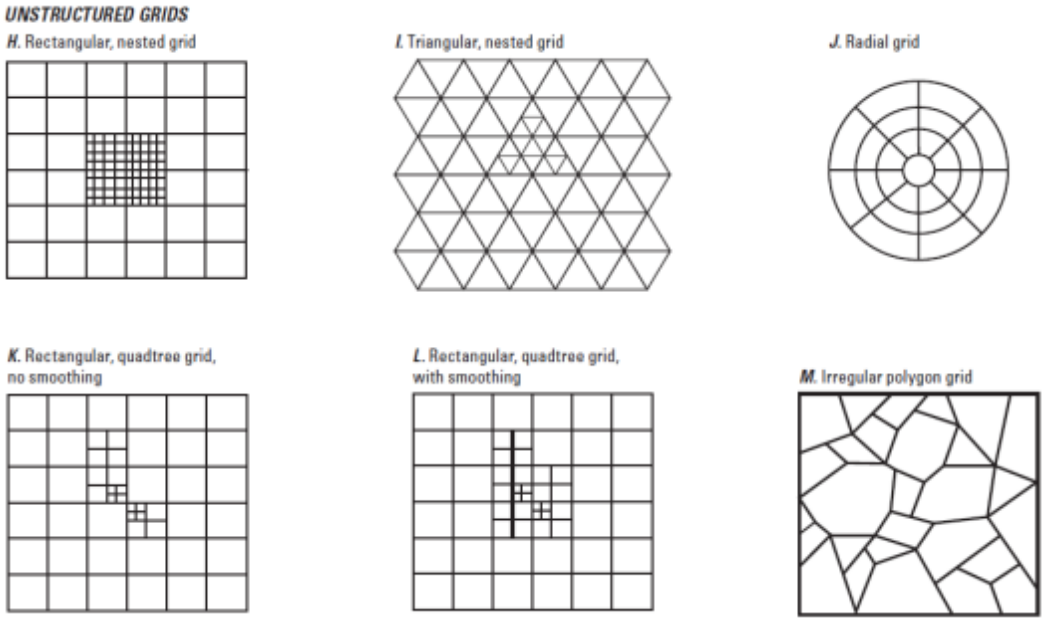


Figure 2.3: Examples of Different Types of Unstructured Grids (Figure from Panday et al., 2013)

Unstructured grids are desirable due to their ability to focus resolutions along and around important features like rivers and wells within the conceptual model without violating the standard CVFD requirements. All of the grids in Figures 2.1 and 2.3 meet the CVFD requirements for MODFLOW-USG. Every type of grid in Figures 2.1 and 2.3, except for the Radial Grid (J), is within the scope of this thesis,

and thus is intended to be handled by the Waterloo Method. The advantages of using unstructured grids in groundwater flow model include better resolutions of features without refining the rest of the conceptual model; limiting grid refinement only to areas of interest. Figure 2.4 shows the unstructured version of Figure 2.2 using Voronoi Tessellation. The areas of interest become easier to identify, allowing the modeler to better understand the conceptual model and its important features.

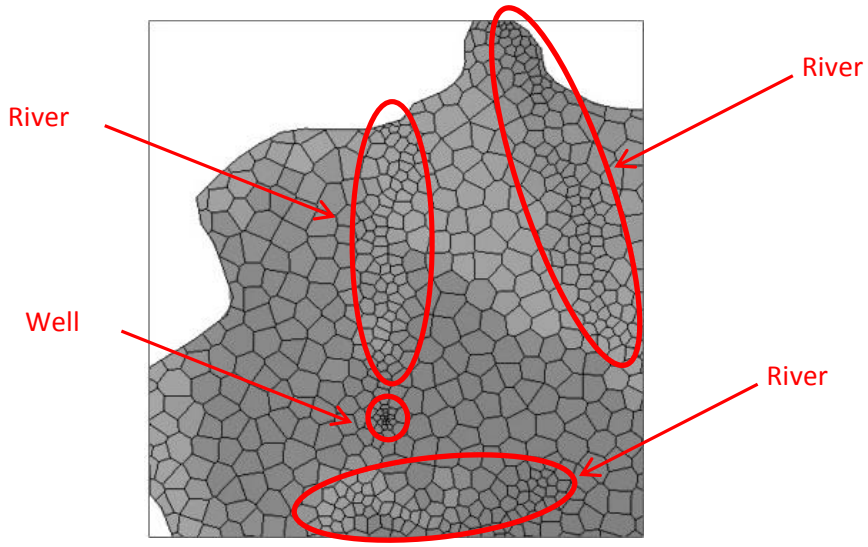


Figure 2.4: Voronoi Unstructured Grids of a Conceptual Model (Figure from Hesch, 2013)

The features, such as the rivers and the well, within the conceptual model in Figure 2.4 become more apparent when using unstructured grids. The discretization file for the unstructured grids in MODFLOW-USG is designed differently from the structured grids file. The current version of MODFLOW-USG does not require the coordinate of each node or cell, only the information regarding the cell connections. Groundwater flow modeling user interfaces, such as Visual MODFLOW Flex and Groundwater Vistas, may provide the Grid Specification File (.GSF) to determine the Cartesian coordinate of the vertices in each cell as well as the coordinate of the each cell's center.

2.3 Particle Tracking Algorithms

There have been numerous particle tracking methods developed over the years to account for the advective component in Lagrangian-Eulerian methods for simulating contaminant transport (Konikow *et al*., 1996; Zheng, 1990; Yeh *et al*., 1992; Clement, 1997), and also for basic particle tracking. The development of different particle tracking methods is necessary to handle special cases of steady or transient flow, velocity variations over each time increment, and different grid structures. Numerical methods, such as finite difference and finite element methods, use fundamentally different descriptions of the velocity field. Examples of particle tracking methods that exist for structured grids include the Pollock method (Pollock, 1988), a semi-analytical particle tracking method for use with velocities generated from block-centered finite-difference groundwater flow models: This method is used by multiple contaminant transport methods, including the 3D method of characteristics groundwater flow and transport model software (MOC3D) (Konikow *et al*., 1996); an algorithm for the simulation of coupled hydrological transport of single or multiple chemical species and their chemical reactions (Zheng, 1999), incorporated into the Modular 3D Multi-Species Transport Model for Simulation of Advection, Dispersion, and Chemical Reactions of Contaminants in Groundwater System (MT3DMS); Other algorithms include those that can account for changes in velocity during a time step in a complicated unsteady flow (Suk and Yeh, 2009; 2010). By far, the most popular particle tracking method for rectilinear structured grids is the semi-analytical Pollock method, which is incorporated into the MODPATH particle tracking utility for MODFLOW.

There have been numerous attempts over the past few years to develop a particle tracking algorithm for arbitrary unstructured grids. A new method for calculating flow streamlines or pathlines from a finite-volume flow solution was developed recently by Painter *et al.* (2012). This method uses unconstrained and constrained least squares methods on interior cells and boundary cells respectively in order to approximate cell-centered velocities. The reconstructed velocities are continuously interpolated in the streamline tracking calculation. The advantages of this method include simple implementation and

ability to fully support unstructured grids with arbitrary intercell connectivity. Furthermore, this method honors specified flux boundaries, including no-flow boundaries exactly, as long as control-volume nodes are placed on the domain boundaries. However, the method of Painter *et al*. (2012) may be insensitive to the number of discrete flux values available for each cell, even though it was tested and demonstrated using a flow solution based on the two-point flux approximation. Furthermore, the velocity field generated using this method does not respect the mass balance at cell faces or within the cell. In 2014, S.S. Papadopulos & Associates announced the release of mod-PATH3DU (Muffels *et al*., 2014), a particle path and travel-time simulator that can support a wide variety of structured and unstructured grids, which uses a kriging-based head interpolation approach and suffers from similar mass balance problems. More details regarding these other unstructured grids particle tracking algorithms are discussed later in this section.

2.3.1 MODPATH

The Pollock Method was first published in 1988 (Pollock, 1988) for semi-analytical particle tracking on rectilinear structured grids. The key assumption of the method is that each directional velocity components varies linearly within grid cell in its own coordinate system, which allows an analytical expression to be obtained describing the flow path within an individual grid cell (Pollock, 1988). Given the initial position of the particle as well as the cell geometry and the flows in through the cell faces, the particle coordinate along its pathlines within the cell and its travel time can be computed directly without numerical integration. An important application of this method includes tracing particle pathlines through any multidimensional flow field that is generated from a block-centered finite-difference groundwater flow model, such as MODFLOW. One limitation of using this method is that it cannot be extended to non-rectilinear cells. The method is here described in detail, because it is used for benchmarking of the new method. Figure 2.5 illustrates a rectilinear finite-sized cell and the inflows and outflows across its faces.

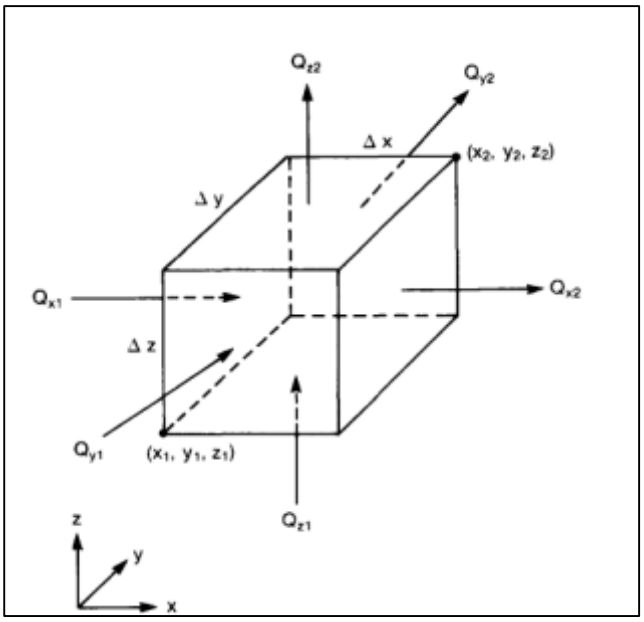


Figure 2.5: The Orientation of Rectilinear Finite-Difference Cells and the Cell Face Flows (Figure from Pollock, 1988)

The calculation regarding the linear velocity interpolation functions in x-direction is as follows:

$$v_{xp} = A_x(x_p - x_1) + v_{x1} \tag{2.4}$$

$$A_x = \frac{v_{x2} - v_{x1}}{\Delta x} \tag{2.5}$$

where v_{xp} is the current particle velocity in x-direction [L/T]; A_x is a constant that correspond to the component of the velocity gradient within the cell [1/T], which can also be described as $\frac{\partial v_x}{\partial x}$; x_1 represents the x-coordinate location of the face that is perpendicular to the x-direction at $x = x_1$ [L]; x_p is the particle coordinate location in terms of x-direction [L]; v_{x1} and v_{x2} are the groundwater velocities at faces x_1 and x_2 respectively [L/T]; and Δx is the dimensions of the cell in the x direction ($\Delta x = x_2 - x_1$) [L]. The linear particle velocity and the velocity gradient constant (A_x) are calculated in an identical manner for the y- and z-directions.

After obtaining the particle velocity and the velocity gradient constant, the time required for the particle to reach a particular face in the cell needs to be calculated in the x, y, and z directions:

$$\Delta t_x = \left(\frac{1}{A_x}\right) \ln\left(\frac{v_{x2}}{v_{xp}}\right) \tag{2.6a}$$

$$\Delta t_y = \left(\frac{1}{A_y}\right) \ln\left(\frac{v_{y2}}{v_{yp}}\right) \tag{2.6b}$$

$$\Delta t_z = \left(\frac{1}{A_z}\right) \ln\left(\frac{v_{z2}}{v_{zp}}\right) \tag{2.6c}$$

The first step is to determine the face in which the particle leaves the cell; whether it is the x-face, y-face, or z-face. Equations 2.6a, 2.6b, and 2.6c are used to solve for the time it would take for the particle to reach each potential exit plane. The smallest value of Δt obtained from the equations corresponds to the actual exit face. The actual location of the particle exit point or particle location at a given time within the cell can then be calculated using the following formula:

$$x_p(t_2) = x_1 + \left(\frac{1}{A_x}\right)\left[v_{xp}(t_1)\ \exp(A_x\ \Delta t) - v_{x1}\right] \tag{2.7}$$

Where, $x_p(t_2)$ is the particle location along the x-axis at time t_2 [L]; Δt is the minimum travel time for the particle to exit either x, y, or z face $\left(\Delta t = \min\left(\Delta t_x,\ \Delta t_y,\ \Delta t_z\right)\right)$ [T]. The location for a particle at a given time within the cell for y- and z-axis can also be computed in a similar manner. Equations 2.6 and 2.7 can be applied to determine both the particle exit location and the time required for the particle to exit the cell. The location and the time are then used as the starting time and coordinate of the particle for the next cell until it reaches a boundary, an internal sink/source, or satisfies another termination criterion.

The Pollock method is implemented in the MODPATH algorithm (Pollock, 1994), which have been officially released as the particle tracking method for MODFLOW. Applications of MODPATH include capture zone delineation and the evaluation of hydraulic containment of pump-and-treat system. MODPATH version 1 was first released in 1989 to calculate pathlines of particles in steady-state flow for MODFLOW-88. The most notable advantage of using MODPATH as a particle tracking algorithm is fast computational time; MODPATH is currently the fastest particle tracking algorithm available for finite-

difference simulations. However, a disadvantage of MODPATH includes its restriction regarding rectilinear structured grids.

In 2015, a new extension of MODPATH was developed to handle rectangular unstructured grids (Pollock, 2015), such as shown in Figure 2.3 (H), (K), and (L). The new extension is able to sufficiently handle rectangular-based structured grids as well as rectangular-based unstructured grids such as nested grids and quad-based grids. The method is extended for two-dimensional square grid cells that have a maximum 2:1 ratio of grid cells at adjacent lateral faces. Even with the new extension, the limitation of MODPATH regarding rectilinear cells still remains.

2.3.2 Tracking on Unstructured Grids

Due to the limitations in the Pollock method, it is desirable to develop a particle tracking algorithm that can handle arbitrary structured and unstructured grids with the same high degree of accuracy. Creating particle pathlines in unstructured grids is problematic due to the calculation requirement regarding a continuous velocity field. The kriging-based approach in mod-PATH3DU is one of the only particle tracking algorithms that can handle arbitrary unstructured grids. Another example of particle tracking method for unstructured control-volume grids is the method by Painter *et al.* (2012). These two methods are the only two known methods for tracking particles in unstructured grids. mod-PATH3DU (Muffels *et al.*, 2014) was developed in 2014 by S.S. Papadopulos & Associates to calculate the three dimensional flow pathlines of purely advective solute particles for structured and unstructured grids supported by MODFLOW-USG. There are currently two particle tracking schemes contained within this software: Pollock method, and the SSP&A method. The Waterloo method developed for this thesis is incorporated into the next version of mod-PATH3DU; the users are able to choose which of the three particle-tracking methods is most suited for their need. The Pollock method in this program is applicable to rectilinear structured grids. The SSP&A method is grid independent as it uses local universal kriging interpolation of a MODFLOW hydraulic head solution and calculates velocity vectors from Darcy's law using the

resultant head changes. The use of universal kriging to interpolate heads in the vicinity of a particle is applicable for velocity calculations in x- and y-direction; the SSP&A method currently uses the Pollock Method to calculate particle velocity in z-direction.

The numerical integration of particle paths in mod-PATH3DU is performed using a fourth-order Runge-Kutta scheme; the option of using Euler scheme for the particle pathlines generation is also available. The limitations regarding the applicability of the SSP&A method for particle tracking include the assumptions in the underlying tracking scheme, interpolation method, and the limitations in the groundwater flow model such as boundary and discretization effects. Furthermore, the SSP&A method cannot respect no-flow boundaries, the law of refraction, and the local velocity field does not respect mass balance. The symptoms of the particle tracking errors using the universal kriging interpolation depend on grid discretization, severe heterogeneity, and the proximity to certain boundaries. The main input files for mod-PATH3DU include files from MODFLOW-USG and files from third party software, such as grid-specification files. More information regarding the required input and output files by this program is found in the User's Guide for mod-PATH3DU (Muffels *et al*., 2014).

The particle tracking method that was introduced by Painter *et al*. (2012) is also able to trace particle pathlines on unstructured control-volume grids. As mentioned earlier in this section, Painter *et al*. approach is able to use constrained and unconstrained least squares methods on boundary cells and interior cells respectively. The cell-centered velocities that are approximated using these methods can be interpolated to any point in the domain of interest. This method has been compared to the Pollock method in structured grids and claimed that the overall comparison between the two results is reasonably good. Several advantages of the method by Painter *et al*. include simple implementation and able to fully support unstructured grids with arbitrary intercell connectivity. Unlike the SSP&A method, the Painter *et a*l. approach honors specified flux boundaries, including no-flow boundaries, but only when control volume nodes are placed on the domain boundaries. The method originally assumes constant Darcy's velocity in each cell to develop its equations. This assumption may result in discontinuous velocities at

cell faces, which can lead to inadequate particle tracking due to errors in mass balance. However, the discontinuities may be partially handled through a simple smoothing procedure.

2.4 Analytical Solutions to Laplace Equation

The Waterloo Method is based upon reconstructing the cell velocity field using local analytical solutions to the Laplace and Poisson equations, which are often encountered in fluid mechanics, elasticity, and heat and mass transfer theory. As will be shown later, the Laplace equation can result from vertical integration of the MODFLOW governing equation within a cell. The two-dimensional Laplace equation for discharge potential $\Phi(x, y)$ in Cartesian coordinates is:

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0 \tag{2.8}$$

This governing equation for sub-grid groundwater flow may be derived via application of continuity and Darcy's law within a homogeneous, isotropic cell over a single time step. Note that Equation 2.8 has no direct dependency on time, only on the two-dimensional spatial variables x and y. Thus, the Laplace Equation above describes the steady state flow condition, and is solved subject to some boundary conditions. The analytical solution to Laplace Equation is one of the main foundations for the Waterloo method.

2.4.1 Dupuit-Forchheimer Approximation

The Dupuit-Forchheimer approximation to a three-dimensional groundwater flow model suggests that flowlines are predominantly horizontal and lateral velocities do not vary over the aquifer depth (Haitjema, 1995); the hydraulic head is likewise presumed not to vary in the vertical direction. The main purpose of this approximation is generating three-dimensional streamlines using two-dimensional horizontal model by assuming that the resistance in vertical direction is nonexistent. This approximation was formulated by Jules Dupuit and Philipp Forchheimer in the late 1800s in order to simplify the equations for groundwater

flow to be able to obtain analytical solutions more easily. Here, the Dupuit-Forchheimer assumption will be invoked in each cell to simplify the three-dimensional groundwater flow equation into the more analytically tractable Poisson equation in each cell. This approximation may be used in computer models that are based on numerical or analytical approaches (Strack, 1984).

2.4.2 Discharge Potential and Taylor Series Function

For generating potential flow solutions, it is useful to work in terms of a complex discharge potential, expressed in complex coordinates. The use of z symbol for the rest of this thesis is referred as the complex coordinate $z = x + iy$ and no longer used to describe vertical direction in Cartesian coordinate, unless stated otherwise. A special property of any function of z alone is that its real and imaginary parts automatically satisfy the Laplace equation. In other words, this property includes $f(z)$, but it does not include $f(z, \bar{z})$, where $\bar{z}$ is the complex conjugate of z. The complex potential $\Omega(z)$ can be written as follows:

$$\Omega(\mathrm{z}) = \Phi(\mathrm{z}) + i\Psi(\mathrm{z}) \tag{2.9a}$$

$$\nabla^2\Omega = \nabla^2\Phi + i\nabla^2\Psi = 0\,, \qquad \Omega = \mathrm{f(z)} \tag{2.9b}$$

The function Φ is here a surrogate for head in the groundwater flow equation. The stream function, denoted as Ψ, has contours that is always perpendicular to the potential contours; i.e., lines everywhere tangent to the flow direction. A complex discharge function can be defined as:

$$W = -\frac{d\Omega}{dz} = -\frac{\partial\Phi}{\partial x} - i\frac{\partial\Psi}{\partial x} = Q_x - iQ_y \tag{2.10}$$

where z is the complex coordinate; and $-\frac{\partial\Psi}{\partial x}$ can also be described as $\frac{\partial\Phi}{\partial y}$ such that the discharge function W has its real and imaginary part of the discharge vector components. Using the discharge function, the discharge vector can be evaluated. Discharge vectors can be used in groundwater modeling to determine

the sub-grid velocity in each cell that can be useful for determining particle pathlines. The key point of the discharge potential in Equation 2.9a is that any Ω which is a function of complex coordinate along z satisfies the Laplace equation. For example, any function $f(z)$ can be used as a candidate solution to the complex Laplace equation. The two functions of complex coordinate z discussed in this thesis include the complex Taylor series and complex logarithmic function:

$$\mathrm{f(z)} = \sum_{n=0}^{\infty} a_n z^n \tag{2.11a}$$

$$\mathrm{f(z)} = \frac{Q}{2\pi} \ln(z) \tag{2.11b}$$

An example of complex Taylor series used to solve the complex potential due to a circular inhomogeneity described by (Janković & Barnes 1999) is as follows:

$$\Omega(\mathrm{z}) = \Phi(\mathrm{z}) + i\Psi(\mathrm{z}) = \begin{cases} \sum_{n=0}^{\infty} a_n z^n & |z| < 1 \\ -\sum_{n=0}^{\infty} \overline{a_n} z^{-n} & |z| \geq 1 \end{cases} \tag{2.12}$$

Where Ω is the complex potential function; Φ is the discharge potential function; and Ψ is the associated stream function; z is the location of interest in global coordinates in complex form $z = x + iy$; a_n is unknown complex coefficient. Equation 2.12 is applicable for simulating two-dimensional flows through large numbers of circular inhomogeneities in the Analytic Element Method (AEM).

The complex Taylor series and complex logarithmic functions shown in Equations 2.11a and 2.11b are here used as "building blocks" to assemble local exact solutions to the groundwater flow equation in each cell. By determining potential and stream function for each cell in a conceptual model, the sub-grid velocity distribution can be determined. The horizontal velocity at any point within a cell is required for tracking particles.

Chapter 3

Methodology

The Waterloo method is here developed to accurately track particles through various types of structured and unstructured grids in heterogeneous, anisotropic subsurface media, with steady-state or transient flow that may be influenced by sink or source terms; essentially filling the gaps that MODPATH cannot. The input files that are crucial for this method are mainly the cell geometry files and cell-by-cell flow files, as well as other necessary input files required for mod-PATH3DU. The development of Waterloo method prototyped in MATLAB and later transferred into C++ for incorporation into mod-PATH3DU.

3.1 Governing Equation

The governing equation for three-dimensional groundwater flow through porous media with constant density, similar to Equation 2.3 in Chapter 2, is given as follows (Rushton and Redshaw, 1979):

$$\frac{\partial}{\partial x}\left(K_x \frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial h}{\partial z}\right) + W = S_s \frac{\partial h}{\partial t} \tag{3.1}$$

As is mentioned in Chapter 2, this equation is valid for transient or steady state groundwater flow through heterogeneous, anisotropic media. Equation 3.1 is derived via application of continuity and Darcy's law.

The sub-grid flow field is generated by invoking the Dupuit-Forcheimer approximation in each cell, whereby head is treated as uniform in the vertical direction and resistance to flow in the vertical direction is treated as negligible. The basic condition for the Dupuit-Forchheimer approximation is for the length of a flowline in a groundwater flow model to be larger in comparison to the aquifer thickness; this basic condition is often met in groundwater flow modeling. By invoking the Dupuit-Forchheimer approximation of the MODFLOW governing equation (Equation 3.1), and assuming no source and sink term within the cell, the three-dimensional equation can be transformed into a two-dimensional equation

by discretizing time derivation and vertically integrating over the saturated thickness of the aquifer, simplifying Equation 3.1 into:

$$\frac{\partial}{\partial x}\left(K_x b \frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y b \frac{\partial h}{\partial y}\right) = \left(-(q_{top} - q_{bot}) + S\frac{\Delta h}{\Delta t}\right) \tag{3.2}$$

Where K_x and K_y [L/T] are the hydraulic conductivities in the x and y directions; h [L] is the hydraulic head; Δh is the variation in hydraulic head over the time step; b is the saturated thickness of the cell [L]; and S [-] is either equal to $S_s b$ (the storativity) or S_y (the specific yield), depending upon whether the cell is confined or unconfined. The fluxes q_{top} and q_{bot} [L/T] are the fluxes through the top and bottom of the cell, respectively, are obtained from vertically integrating Equation 3.1, and are treated as positive for upward flows. Note that in each cell, for each MODFLOW time step, K_x, K_y, q_{top}, q_{bot}, $\Delta h/\Delta t$ and S are constant and uniform. Also note that the storage term, $S\,\Delta h/\Delta t$, is treated as a constant over the MODFLOW time step and therefore will be treated in an analogous manner to the vertical flux.

For convenience, a Discharge potential, Φ [L^3/T] is here used as a surrogate for groundwater head. The discharge potential used in this thesis is given by Strack (1989):

$$\Phi = \begin{cases} \frac{1}{2}K(h - z_b)^2 & \text{(unconfined)} \\ KH(h - z_b) - \frac{1}{2}KH^2 & \text{(confined)} \end{cases} \tag{3.3}$$

Where $K = K_x = K_y$ is laterally isotropic hydraulic conductivity [L/T] and H is the uniform constant saturated thickness in cases of confined flow [L]. By defining a discharge potential, Equation 3.3 can be consistently treated for both confined ($b = H$) and unconfined ($b = h - z_b$, where z_b is the cell bottom elevation) conditions. By combining Equations 3.3 and the left hand side of 3.2, the discharge potential can be obtained. The two-dimensional integrated discharge can be defined by vertically integrating the specific discharge and applying Darcy's law to obtain the following expression:

$$Q_x = bq_x \ = b\left(-K\frac{\partial h}{\partial x}\right) \equiv -\frac{\partial \Phi}{\partial x} \tag{3.4a}$$

$$Q_y = bq_y \ = b\left(-K\frac{\partial h}{\partial y}\right) \equiv -\frac{\partial \Phi}{\partial y} \tag{3.4b}$$

Where q_x and q_y [L/T] are the specific discharges in x- and y-direction respectively; Q_x and Q_y [L^2/T] are the integrated discharge; and b is the saturated thickness. Using the discharge potential form from Equation 3.3, the governing equation shown in Equation 3.2 for two-dimensional Dupuit-Forchheimer flow can be simplified into the two-dimensional form of Poisson Equation in terms of the discharge potential:

$$\nabla^2\Phi = -\mu \tag{3.5}$$

Where $-\mu$ is the entire right hand side of Equation 3.2, and $\nabla^2\Phi$ is the entire left hand side of Equation 3.2. Note that Equation 3.5, the Poisson equation, is related to the two-dimensional Laplace equation as shown in Equation 3.2; solutions may be generated by superimposing solutions to the Laplace equation with any particular solutions to Equation 3.5. Due to the Dupuit-Forchheimer assumption, the vertical velocity at any point within the cell can be calculated separately or even using another method.

3.2 Local Problem Definition

The first step of the Waterloo method is the reconstruction of sub-grid velocity distribution on a cell-by-cell basis for each MODFLOW time step. There are two types of time steps discussed here: The time step used for numerical integration of particle pathlines, called the tracking time step; and the time step used MODFLOW in calculation of transient flows and heads, called the MODFLOW time step. In all cases, the tracking time step is always smaller than the MODFLOW time step. Reconstruction at the sub-grid-cell level simplifies the particle tracking problem to that of a relatively simple boundary value problem within the cell, which satisfies specified flux conditions along all sides, and Equation 3.5 internal to the

cell. The four important parameters that need to be populated for the preliminary calculations in Waterloo method are: 1) the cell geometry; 2) the time-averaged flow rates in through the sides, top, and bottom of the cell for each MODFLOW time step; 3) the flow rates of any wells within the cell; 4) the saturated thickness at the start and end of the MODFLOW time step; and 5) the tracking duration.

The key algorithm parameters to be determined in the Waterloo method are the total control points (M) and the order of approximation (N) of each cell. These parameters are necessary in order to determine the accuracy of the velocity reconstruction resultant and resultant travel times. M is specified control points along the perimeter of the cell. These points along the perimeter of the cell control the accuracy of the calculation regarding sub-cell velocity reconstruction. The second crucial parameter for the preprocessing step of the algorithm is the order of approximation (N). Both M and N are used to determine the accuracy of the solution to the local boundary value problem (BVP), which is further discussed in a later section in this chapter. The basic rule regarding these two parameters is that the number of total control points needs to be larger than or equal to two times the value for order of approximation ($M \geq 2N$). This ensures more equations than unknowns in the system of equations used for solving the BVP. The optimal values for these parameters are chosen to increase the accuracy and/or reduce the run-time of the simulation. Further details regarding these parameters are discussed in the errors section of this chapter.

3.2.1 Cell Geometry

In order to apply the Waterloo method, the coordinate of each vertex of every cell needs to be known. MODFLOW-USG does not explicitly provide information regarding the Cartesian coordinate of each node or cell, only the discretization file that contains information regarding the cell connection (DIS file for rectilinear structured grids, and DISU file for unstructured grids). The grid specification (GSF) file contains the Cartesian coordinate of each vertex and the connections between those vertices in a cell to make the polygon. These GSF files are not required by MODFLOW-USG, but are generated by third

party software such as Groundwater Vistas or Visual MODFLOW Flex. Cell geometry construction is possible by combining information from the discretization and grid specification files. The first step that the Waterloo method does regarding the cell geometry construction is to create the primary parameters. For example, the following parameters will generate the two-dimensional cell shown in Figure 3.1:

Table 3.1: Primary Parameters to Generate Example 3.1

Parameter Description	Symbols	Example Values
Number of faces	n_{faces}	4
Vertex coordinates of the cell (Arranged clockwise)	$[x_i \quad y_i]$	$\begin{bmatrix} 1000 & 1000 \\ 1000 & 1100 \\ 1100 & 1100 \\ 1100 & 1000 \end{bmatrix}$
Cartesian coordinate of the cell's center	$[x_c \quad y_c]$	$[1050 \quad 1050]$
Order of Approximation	N	20
Total control points	M	40

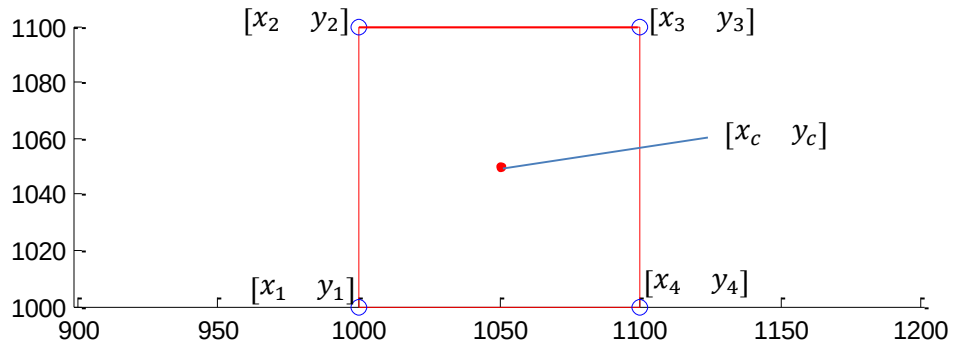


Figure 3.1: An Example of a Cell Geometry Constructed by the Algorithm

Figure 3.1 is an example of a rectilinear cell that has 4 sides and touching 4 other cells, hence 4 cell connections. However, in a situation where it is a quad-based grid, where one side of the cell may be touching two or more cells, the number of vertices for this cell and the number of faces may be more than 4. The Waterloo method only works when the number of faces and the number of vertices match with number of cell connections. Note that it is important for the algorithm that the vertices are arranged in

clockwise direction. However, it is unimportant which vertex is constructed first. The reason for this is because the other processing steps regarding the cell geometry, such as the use of M to specify how far along the perimeter each point is, are only valid when $[x_i \quad y_i]$ is arranged clockwise; arranging $[x_i \quad y_i]$ counterclockwise or in a random manner may cause the algorithm to produce erroneous results. The rearrangement of the vertices is done internally and is compatible with any prearrangement of vertices from any type of grid specification file. Since the total number of control points is already checked to be two times larger than the order of approximation, the next step in the algorithm is to calculate the secondary parameters, defined from the primary parameters.

Table 3.2: Secondary Parameters Calculated from Primary Parameters in Example 3.1

Parameter Description	**Symbols**	**Example Values**
Vertices in complex coordinate	z_i	$\begin{bmatrix} 1000+1000i \\ 1000+1100i \\ 1100+1100i \\ 1100+1000i \end{bmatrix}$
Length of each side of the cell [L]	L_i	$\begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}$
Complex coordinate of the cell center	z_c	$1050+1050i$
Cell perimeter [L]	P	400
Cell area [L^2]	A	10000
Maximum radius between cell center and each cell vertex [L]	R	70.71

The z_i parameter is obtained directly from the $[x_i \quad y_i]$ parameter. Using z_i, the length of each side of the cell can be computed easily even for cells with irregular geometry. The radius between the cell center and each cell vertex is computed using z_i and z_c in a similar manner for calculating L_i. The next preprocessing step for the algorithm regarding the cell geometry is to determine control point locations along sides of the cell; this step is to calculate the cell control parameters:

Table 3.3: Cell Control Parameters Calculated using the Primary and Secondary Parameters

Parameter Description	Symbols
The side control that specifies how far along the perimeter of the cell each point is	$\vec{S}_{ctrl}[1, M]$
The complex side control points specify the complex coordinate of $\vec{S}_{ctrl}$	$\vec{z}_{ctrl}[1, M]$
The local complex side control points specify the localized $\vec{z}_{ctrl}$	$\vec{Z}_{ctrl}[1, M]$

The $\vec{z}_{ctrl}$ must be transformed into $\vec{Z}_{ctrl}$ such that the problem is recast in a local coordinate system. The equations to calculate $\vec{S}_{ctrl}$, $\vec{z}_{ctrl}$, and $\vec{Z}_{ctrl}$ are as follows:

$$S_{ctrl}^{i} = \frac{(i - 0.5)}{M} P \quad , i = 1{:}\, M \tag{3.6a}$$

$$z_{ctrl}^{i} = \frac{\left(S_{ctrl}^{i} - \sum_{k}^{j-1} L_k\right)}{L_k}\left(z_{j+1} - z_j\right) + z_j \quad , i = 1{:}\, M\,, \qquad j = \text{side with } i \tag{3.6b}$$

$$Z_{ctrl}^{i} = \frac{\vec{z}_{ctrl}^{i} - z_c}{R} \quad , i = 1{:}\, M \tag{3.6c}$$

The $\vec{S}_{ctrl}$ is set such that none of the point is located at the vertex of the cell. $\vec{z}_{ctrl}$ is just the complex coordinate of each $\vec{S}_{ctrl}$ point in the cell. Note that the number of faces parameter only includes the side faces; the value of this parameter does not include the top and bottom face. $\vec{Z}_{ctrl}$ is a local coordinate system chosen such that the center coordinate of the "new localized cell" is located at [0 0]. Schematic of an irregular shape cell that contains details regarding the primary, secondary, and cell control parameters is shown in the next sub-section.

3.2.2 Time-Averaged and Well Flow Rates of the Cell

The flow rates for every cell can be obtained from the cell-by-cell flow file, one of the MODFLOW-USG output files (the CBC file for the rectilinear structured grids and CBB file for the unstructured grids).

Since the cell vertices need to be rearranged in clockwise direction, the flows through each side of the cell also need to be rearranged accordingly to match with the vertices. The dimension of the flows obtained from the cell-by-cell flow file is L^3/T. The sign convention for the flows in through the sides of the cell (Q_i) is negative for the flows leaving the cell, and positive for the flows entering the cell. The sign convention for the vertical flows (Q_{vert}) for the cell is positive in upward direction and negative in downward direction. Example 3.2 is provided in this sub-section to show how the algorithm handles a cell with irregular geometry. The primary parameters of example 3.2 include:

Table 3.4: The Primary Parameters of Example 3.2

Parameter Symbols	Example Values
n_{faces}	5
$[x_i \quad y_i]$	$\begin{bmatrix} 253.0483 & 53.0072i \\ 251.4244 & 49.7577i \\ 247.8321 & 50.2979i \\ 247.2358 & 53.8814i \\ 250.4596 & 55.5558i \end{bmatrix}$
$[x_c \quad y_c]$	[250 52.5]
N	20
M	50

The secondary and cell control parameters for example 3.2 can be computed using the primary parameters shown in Table 3.4. The schematic of example 3.2 is shown in Figure 3.2.

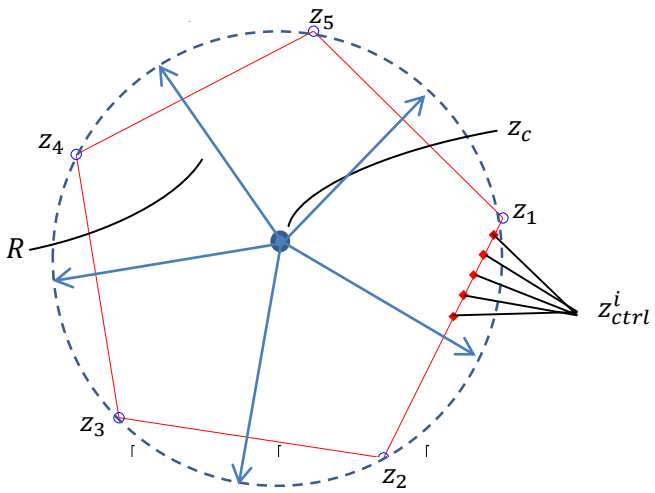


Figure 3.2: Schematic of Example 3.2 that Contains Primary, Secondary, and Cell Control Parameters

The flow parameters are shown in Table 3.5 below:

Table 3.5: Flow Parameters for Example 3.2

Parameter Description	Symbols	Example Values
The time-averaged flow rates in through the sides of the cell; negative outward, positive inward. [L^3/T] (This parameter must be arranged in accordance to the cell vertices)	Q_i	$\begin{bmatrix} -25 \\ -25 \\ 25 \\ 15 \\ 20 \end{bmatrix}^T$
Cumulative vertical inflows [L^3/T] (Negative upward, positive downward)	Q_{vert}	-30
Well flow rates [L^3/T] (Negative for pumping well, positive for injecting well)	Q_{well}	-40

As shown in Table 3.5, the value for well flow rates is negative for pumping well (i.e., the water leaving the cell), and vice versa for injecting well. The total net inflow to cell must be matched by a corresponding change with volume (i.e., for any given time step, the sum of Q_i, Q_{well}, and Q_{vert} that are evenly distributed out to the cell perimeter must be equal to zero) in order to satisfy the mass balance. It is assumed that, given no other information, fluxes are uniformly distributed along each side of the element. This assumption may later be relaxed to handle special cases; however it is the easiest approach, which also ensures continuity between individual cells without the knowledge of adjacent cell flows. Figure 3.3 represents the flow distributions and symbol conventions for example 3.2.

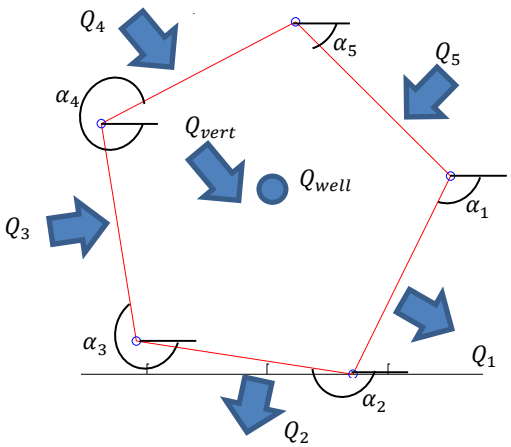


Figure 3.3: Flow Distribution for Example 3.2 that Includes Cell, Vertical, and Well Flow Rates, and Angle of Each Side

The Q_{vert} shown in Table 3.5 includes the flows through the top and bottom of the cell, and storativity. The calculation of Q_{vert} is similar to the $-\mu$ term in Equation 3.5. The value for the change in storage is purposely left out from the table and assumed zero since example 3.2 is a steady state flow case. Details regarding the storativity are discussed later in this chapter. It is crucial to ensure the mass balance of the cell regarding the flows such that:

$$\left(\sum_{i=1}^{n_{faces}} Q_i\right) - Q_{vert} + Q_{well} = 0 \tag{3.7}$$

The cell-by-cell flow information from MODFLOW should ensure the mass balance of each cell; failing to satisfy the condition in Equation 3.7 may cause the algorithm to produce an erroneous result. The vertical flux (q_{vert}) [L/T] of the cell is computed such that $q_{vert} = \frac{Q_{vert}}{A}$. The information regarding the wells provided by MODFLOW only includes the flow rates and the cell(s) in which the well(s) is/are located. MODFLOW does not provide any information regarding the coordinate location of each well, instead it assumes that any well exists within a cell is located at the center of the cell. Thus, example 3.2 also uses this assumption. The information regarding the well flow rates for the pumping and injecting wells is provided by MODFLOW in the well package file.

After ensuring that the sum of corrected inflows satisfies the mass balance, the next step is to distribute the cell, vertical, and well flow rates along the perimeter of the cell; the parameters associated with this step are called the flow control parameters, which contains specified flux along all sides of the cell. The flow control parameters include the normal flows distributed along each of the z_{ctrl}. The cell inflows are assumed to be distributed normally along the cell's sides. The schematic of the side angles are provided in Figure 3.3. The cell side angles (α_i) are obtained by using the argument of each side of the cell such that $\alpha_i = \arg(side_i)$. The description of the flow control parameters and the cell side angles are shown in Table 3.6.

Table 3.6: Description of Flow Control Parameters and Cell Side Angles

Parameter Description	Symbols
The angle for each side of the polygon calculated in radian (e.g. $\alpha_1 = \arg\big(\mathrm{Im}(z_2 - z_1) - i\ \mathrm{Re}(z_2 - z_1)\big)$	$\alpha_i[1, n_{faces}]$
The normalized cell flows	$Q_\eta^{cell}[1, M]$

The calculations regarding the normalized cell, vertical, and well flows are necessary in order to determine the flow pattern inside the cell that contains the potential and stream functions. The equation to compute the normalized cell flows is as follows:

$$Q_{\eta\, i}^{cell} = \frac{Q_j}{L_j} \tag{3.8}$$

Where $i = 1{:}\,\mathrm{M}$, and $j = 1{:}\, n_{faces}$, and j is the side corresponding to control point i The determination of $Q_{\eta\, i}^{cell}$ is crucial for calculating the particle velocity inside the cell. The calculations regarding normalized vertical and well flows are discussed in the next section.

3.2.3 Saturated Thickness

The saturated thickness reported in MODFLOW calculation is the relative saturated thickness for the cell; i.e., it is the current saturated thickness normalized by the height of the cell, which value varies between 1 and 0. The coordinate of the top of water table is also calculable and done internally. The saturated thickness for each MODFLOW time step is necessary to both determine the sub-grid horizontal velocity distribution within the cell and to calculate the vertical velocity for transient flow case where the water table position may increase or decrease over time. Figure 3.5 shows an example of the saturated thickness that is lower than the top of water table. The example depicted in Figure 3.5 shows a cell that is 60% full. Thus, the relative saturated thickness reported by MODFLOW will be 0.60; a fully saturated cell will have saturated thickness of 1.00. The saturated thickness is used to determine the storage term of the cell as shown in Table 3.5 and to calculate the particle velocity at any point within the cell.

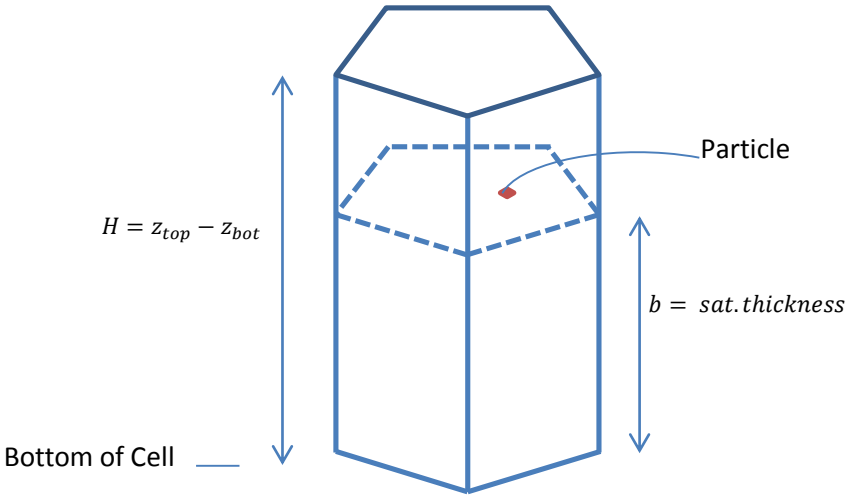


Figure 3.4: An Example of Water Table Located Below the Top of the Cell

In transient flow case, the saturated thickness is especially important. Information regarding this matter is discussed in the transient flow section.

3.3 Solution Approach

Here, the local solution to the Poisson equation, shown in Equation 3.5, is generated via superposition of the potential functions for the cell, vertical, and well flows. The complex Taylor series is a flexible and exact solution to Equation 3.5 without the distributed source term and is augmented by additional terms, which exactly simulate the influence of wells, vertical recharge, leakage, and changes in storativity. As mentioned earlier in Chapter 2, the complex potential form in Equation 2.9a can be solved using Taylor series approximation. The form of the solution is given as a complex potential:

Complex Flow Field Function | Vertical Flux Function | Well Function

$$\Omega(Z) = \Phi(\mathrm{Z}) + i\Psi(Z) = \sum_{n=0}^{N} a_n Z^n + \frac{q_{vert}}{2}\mathrm{Re}(RZ)^2 + \frac{Q_{well}}{2\pi}\log(R|Z - Z_w|) \quad (3.9)$$

Where $Z = \frac{z - z_c}{R}$ is the local complex coordinate; $z_c = x_c + iy_c$ is the cell center ; $z = x + iy$ is the global complex coordinate; and R is the maximum radius of a circle centered at this point which fully encircles the cell. N is termed the order of the approximation. A well with pumping/injecting rate Q_{well} is located

at local well coordinate $Z_w = \frac{z_{well} - z_c}{R}$. Note that this solution approach generates both the discharge potential and stream function within the cell, though the stream function is, strictly speaking, only defined if q_{vert} and Q_{well} are zero. Using the flexible coefficients of the Taylor series (a_n), a local flow solution which satisfies the governing flow equation exactly and precisely matches the boundary flux distribution along cell interfaces can be generated. The vertical flux function and well function are specified such that they satisfy the vertical flow distribution of uniform influx and a point source respectively; they do not satisfy the side boundary conditions, but they do satisfy the governing Equation 3.5. The coefficients a_n are determined by matching the normal flow rates along the sides of the cell at a set of M control points along the cell perimeter. This leads to a set of M equations for N unknowns which is solved in a least squares sense as done by Janković and Barnes (1999). The details regarding coefficient calculation are explained in the next sub-section. Equation 3.9 is necessary to create the flow pattern inside the cell for potential and stream lines (the real value for potential contours and imaginary value for stream function contours). In order to determine the flow pattern inside the cell, the complex flow field function, vertical flux function, and well function need to be calculated. The complex flow field function calculation depends on the order of approximation described in previous section. Thus, the only unknown parameters in Equation 3.9 are the coefficients a_n.

3.3.1 Identifying Coefficients

The coefficients are determined by matching the normalized flow rates along the boundaries of the cell; the total volumetric flows through each side are therefore also matched. The first step for finding the Taylor series coefficients is to calculate the normalized Taylor flow rates (Q_η^{Taylor}). To obtain this parameter, the algorithm needs to use the combination of normalized flows for the cell, vertical, and well terms. Along each side, the inflows from the three terms in Equation 3.9 must be equal to the normalized cell flows Q_η^{cell} such that:

$$Q_\eta^{Taylor} = Q_\eta^{cell} - Q_\eta^{vert} - Q_\eta^{well} \tag{3.10}$$

The Q_η^{cell} is already obtained from the previous section. The normalized vertical (Q_η^{vert}) and well flows (Q_η^{well}) can be calculated using the vertical (Q_{vert}) and well flow rates (Q_{well}). In order to transform the vertical and well flows, which are determined by the functional form of Equation 3.9, into normal flows, the angle (α_i) for each side of the cell is needed. The description of the side angle α_i is available in Table 3.6. Using these parameters, the Q_η^{vert} and Q_η^{well} can be calculated as follows:

$$Q_{\eta\, i}^{vert} = \mathrm{Re}(-q_{vert} * R * \mathrm{Re}(Z_{ctrl}^i) * (\cos(\alpha_j) + i\sin(\alpha_j)) \tag{3.11a}$$

$$Q_{\eta\, i}^{well} = \mathrm{Re}\left(-\left(\frac{Q_{well}}{2\pi * R * (Z_{ctrl}^i - Z_w)}\right) * \left(\cos(\alpha_j) + i\sin(\alpha_j)\right)\right) \tag{3.11b}$$

Where $i = 1{:}M$. Note that the "i" next to the $\sin(\alpha_j)$ in the equations above is the imaginary number $\sqrt{-1.0}$. Using Equations 3.8, 3.11, and 3.10, the normalized Taylor flows along the boundary of the cell can be determined. These parameters are essential in determining the Taylor series coefficients (a_n). After obtaining Q_η^{Taylor}, the normal flows may be integrated along the cell perimeter such that:

$$\int_0^{S_{ctrl}} Q_\eta^{Taylor}\, ds_{ctrl} = \int_0^{S_{ctrl}} Q_\eta^{cell} - Q_\eta^{vert} - Q_\eta^{well}\, ds_{ctrl} = \Psi^{Taylor}(s) \tag{3.12a}$$

$$\Psi_i^{Taylor} = 0.5\left(Q_{\eta\, i}^{Taylor} + Q_{\eta\, i+1}^{Taylor}\right)\left(S_{ctrl}^{i+1} - S_{ctrl}^{i}\right) + \Psi_{i-1}^{Taylor}, \qquad i = 1{:}M \tag{3.12b}$$

The integral of Q_η^{Taylor} in Equation 3.12a may be evaluated using the trapezoid rule depicted in Equation 3.12b. The stream function needed to be generated by the Taylor series Ψ_i^{Taylor} is now known; The problem now simplifies to finding the coefficients a_n which generate this specified stream function along the cell boundary. After obtaining the Taylor stream function, the Taylor series coefficient can be computed by solving an overspecified system of equations:

$$\mathrm{Im}\left(\sum_{n=0}^{N} a_n\,(Z_{ctrl}^i)^n\right) = \Psi_i^{Taylor}, \qquad \text{for each } i = 1{:}M \tag{3.13}$$

After determining the Taylor stream function, the next necessary parameter for calculating the Taylor series coefficients by using Equation 3.13 is the unit stream function $\left(\Psi^{unit}\right)$ in the left hand side of Equation 3.13, that is determined at each control point along the boundary defined as the stream function generated if the coefficient $a_j = 1$ and all other $a_i = 0$. Using the local control point coordinates Z_{ctrl}, the Ψ^{unit} matrix with the size of M by $2N$ can be calculated. The first step is to use the formula similar to the complex flow field function depicted as the first part of Equation 3.9 to find the imaginary part of the Taylor series.

$$\Psi_{i,j}^{unit} = \mathrm{Im}\left(1\left(Z_{ctrl}^{i}\right)^{j}\right) \text{ , for } j = 0 \text{ to } N, \qquad i = 1 \text{ to } M \tag{3.14a}$$

$$\Psi_{i,j+N}^{unit} = \mathrm{Im}\left(\mathrm{i}\left(Z_{ctrl}^{i}\right)^{j}\right) \text{ , for } j = 0 \text{ to } N, \qquad i = 1 \text{ to } M \tag{3.14b}$$

Note that the "i" next to Z_{ctrl}^{i} in Equation 3.14b is an imaginary value. By obtaining the unit stream function, The system of equations in Equation 3.13 can be reassembled into a matrix form such as $\vec{A} \cdot \vec{x} = \vec{b}$. The matrix assembly of Equation 3.13 is as follows:

$$\begin{bmatrix} \Psi_{1,1}^{unit} & \Psi_{1,2}^{unit} & \cdots & \Psi_{1,N}^{unit} & \Psi_{1,N+1}^{unit} & \cdots & \Psi_{1,2N}^{unit} \\ \Psi_{2,1}^{unit} & \Psi_{2,2}^{unit} & \cdots & \Psi_{2,N}^{unit} & \Psi_{2,N+1}^{unit} & \cdots & \Psi_{2,2N}^{unit} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Psi_{M,1}^{unit} & \Psi_{M,2}^{unit} & \cdots & \Psi_{M,N}^{unit} & \Psi_{M,N+1}^{unit} & \cdots & \Psi_{M,2N}^{unit} \end{bmatrix} \begin{bmatrix} \mathrm{Re}(a_1) \\ \mathrm{Re}(a_2) \\ \vdots \\ \mathrm{Re}(a_N) \\ \mathrm{Im}(a_1) \\ \mathrm{Im}(a_2) \\ \vdots \\ \mathrm{Im}(a_N) \end{bmatrix} = \begin{bmatrix} \Psi_1^{Taylor} \\ \Psi_2^{Taylor} \\ \vdots \\ \Psi_M^{Taylor} \end{bmatrix} \tag{3.15}$$

By following Equation 3.13, the Taylor series coefficient vector can be determined as shown in Equation 3.15. The first half of the unknown coefficient vector consists of the real part of the coefficients, and the second half consists of the imaginary part. The unit stream function matrix for equation 3.15 is obtainable using Equations 3.14a and 3.14b. The system of equations is solved using a least squares algorithm (e.g. mod-PATH3DU uses Eigen Library for C++ to solve this). Essentially, it uses inverse matrix calculation such as $[a_n] = \left[\Psi^{unit}\right]^{-1}[\Psi^{Taylor}]$. After obtaining the coefficient vector, this parameter can now be transformed into complex Taylor coefficient vector by matching the real and imaginary part together such

that $a_i = Re(a_i) + i\,\mathrm{Im}(a_i)$ is a complex parameter. This complex Taylor series coefficient a_n can now be inserted into Equation 3.9 to compute the potential and stream function of the cell, hence creating the flow pattern inside the cell. Note that Z_w is the well local coordinate; when the location of the well is assumed to be at the center of the cell, this parameter is zero. Furthermore, if there is no vertical or well term within the cell the vertical and well functions will also return zero as Q_{vert} and Q_{well} are both zeros.

Figure 3.5 represents a local flow problem that is solved semi-analytically using a complex Taylor Series and shows the capability of the Waterloo method to define the stream and potential function within a cell by reconstructing local flow field such that the flows in through the sides of the cell are uniformly distributed. The calculation of complex discharge potential is essential in determining the errors of the cell.

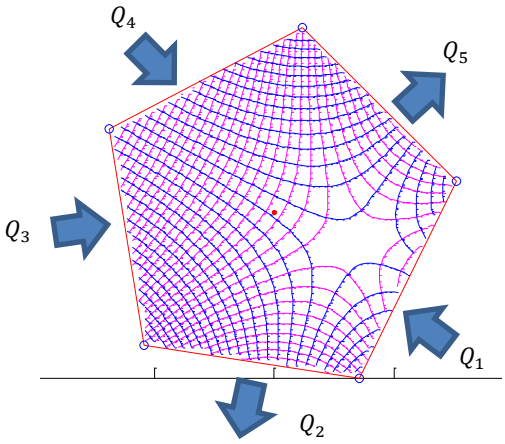


Figure 3.5: The Local Flow Problem, Containing Flows through Each Side, and Potential and Stream Lines within the Cell.

Figure 3.5 only includes the flow in through the sides of the cell with no vertical flux and well flow rates term. The vertical flux and well function only return real value as shown in Equation 3.9. Therefore, if there is a vertical flux and well terms within a cell, the stream lines for the flow pattern are undefined.

3.3.2 Complex Flow Velocities

The horizontal complex flow velocities are essential for creating particle pathlines using the Waterloo method. By differentiating the complex discharge potential with respect to the complex coordinate z, the

velocity at any point may be calculated; essentially differentiating Equation 3.9 with the complex coordinate to obtain a similar result as Equation 2.10 in previous chapter:

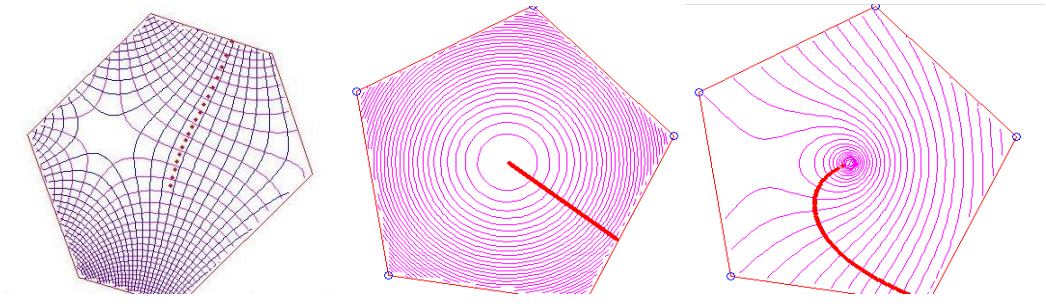
$$Q_x - iQ_y = -\frac{\partial \Omega}{\partial z} = \underbrace{-\frac{1}{R}\sum_{n=0}^{N} n a_n Z^{n-1}}_{\text{Complex Flux Field Function}} + \underbrace{q_{vert} Re(RZ)}_{\text{Vertical Flux Function}} \underbrace{- \frac{Q_w}{2\pi} \frac{1}{R(Z - Z_{well})}}_{\text{Well Flux Function}} \tag{3.16}$$

Here, Q_x and Q_y [L^2/T] are the integrated discharge components, related to velocities v_x and v_y [L/T]. By dividing the fluxes by the saturated thickness and the porosity of the cell, the horizontal velocities at any point within the cell are obtainable:

$$v_x = \frac{\mathrm{Re}\left(-\frac{\partial \Omega}{\partial z}\right)}{b(t_{MOD})\theta} = \frac{Q_x}{b(t_{MOD})\theta} \tag{3.17a}$$

$$v_y = \frac{\mathrm{Im}\left(\frac{\partial \Omega}{\partial z}\right)}{b(t_{MOD})\theta} = \frac{Q_y}{b(t_{MOD})\theta} \tag{3.17b}$$

Where θ is the cell porosity; and t_{MOD} is the MODFLOW time step. The saturated thickness in Equations above represents the saturated thickness of the cell at the start of each MODFLOW time step (it is also referred to as b^n); the value is constant for steady state flow. After calculating the coefficients for the cell, vertical recharge, and well, the horizontal velocity distributions any point within the cell over a specific MODFLOW time step can be obtained by solving Equations 3.17a and 3.17b. Note that because of the Dupuit-Forchheimer assumption, the lateral velocities are presumed to be uniform in the vertical direction. This is consistent with Pollock's method. The particle pathlines distribution in an unstructured cell is apparent in Figure 3.6.



(a) (b) (c)

Figure 3.6: Particle pathlines and head/stream function distributions generated for a cell with no vertical flux term (a), a cell with vertical flux evenly distributed out to the cell perimeter (b), and a cell with a well (c).

Figure 3.6(a) shows particle pathlines with potential and stream lines within the cell generated with no vertical and well fluxes, and hence the streamlines are defined in the image. Figure 3.6(b) shows the particle movement within a cell with only vertical flux that is evenly distributed out to the cell perimeter; the particle is originated at the center of the cell and exited the cell through one of its side. Figure 3.6(c) depicts a cell with a pumping well in a steady state flow case. The particle starting point is at one of the corners of the cell and ended up getting captured by the well. A particle horizontal velocity can be computed by solving Equations 3.16, 3.17a, and 3.17b, using the parameters obtained earlier in previous sections. The pathline integration may be performed using either Euler or High-Order Runge-Kutta schemes that are dependent on the chosen time step or space step. Details regarding these schemes are discussed later. In the algorithm, the horizontal velocities are calculated using Taylor series coefficients and other parameters which will change if the particle enters another cell (or at the beginning of a new MODFLOW time step for a transient flow case). As mentioned earlier, the calculations regarding horizontal velocities within a cell are independent of the calculations regarding the vertical velocities. Therefore, even with transient or steady state flow case, the cumulative coefficients of the cell and other parameters are not affected by the parameters to calculate vertical velocities; this is due to invoking of the Dupuit-Forchheimer approximation.

3.4 Vertical Velocity

The method to generate the vertical velocity is different from the method for the horizontal velocities of the cell, and is based upon mass balance considerations. The μ parameter in the algorithm consists of vertical flows from the top and bottom of the cell, and changes in cell storage (i.e., $Q_{vert} = \left(Q_{top} - Q_{bot}\right) - S\ \frac{\partial h}{\partial t_{MOD}}$). The vertical velocity, v_z, is generated from mass balance considerations in a manner similar to that of Strack (1985). The position of the particle over time is determined via a mass balance on the water below the particle elevation, relying upon the idea that water added from above the particle location will be stacked on top of the particle and will not impact the particle position, but water added or

removed from below the particle will raise or lower the particle. Figure 3.8 depicts the mass balance schematic regarding the vertical case.

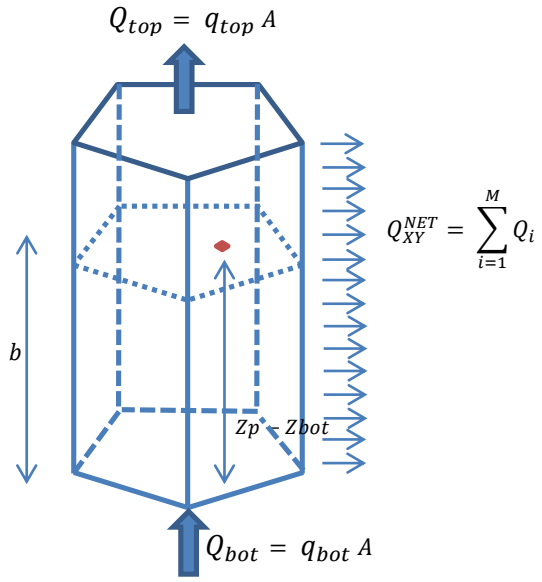


Figure 3.7: Mass Balance Schematic Regarding Vertical Case. Unlike the Horizontal Flows, the Vertical Flows is Marked Positive on Upward Direction.

A mass balance on the water below the particle leads to the equation for vertical velocity for the Waterloo method is expressed as follows:

$$v_z(z_p, t) = \underbrace{\frac{Q_{bot}}{\theta A}}_{\text{Bottom Vertical Flux}} + \frac{z_p - z_b}{\theta A\left(b^n + (t - t_n)\frac{\Delta b}{\Delta t}\right)} \underbrace{\sum_{i=1}^{n_{faces}} Q_i}_{\text{Horizontal Net Inflow}} \tag{3.18}$$

Where z_p is the particle elevation, A is the area of the cell footprint, Q_i is the volumetric flow through the i[th] of n_{faces}, b^n is the saturated thickness at the start of the Modflow time step, which begins at time t_n , and Δb is the change in saturated thickness over the time step. The calculation of $v_z(z_p, t)$ is independent of the calculation to obtain the horizontal velocities in x and y direction. The approach is valid for any cell geometry and may be used for both steady state and transient simulations, provided a steady flow regime is assumed over each model time step. Pollock method also uses linear interpolation in the vertical direction, but the way this algorithm handles the steady state and transient cases in terms of the particle vertical direction is different than the Pollock method, as shown in equation 3.18. As shown in Figure 3.7,

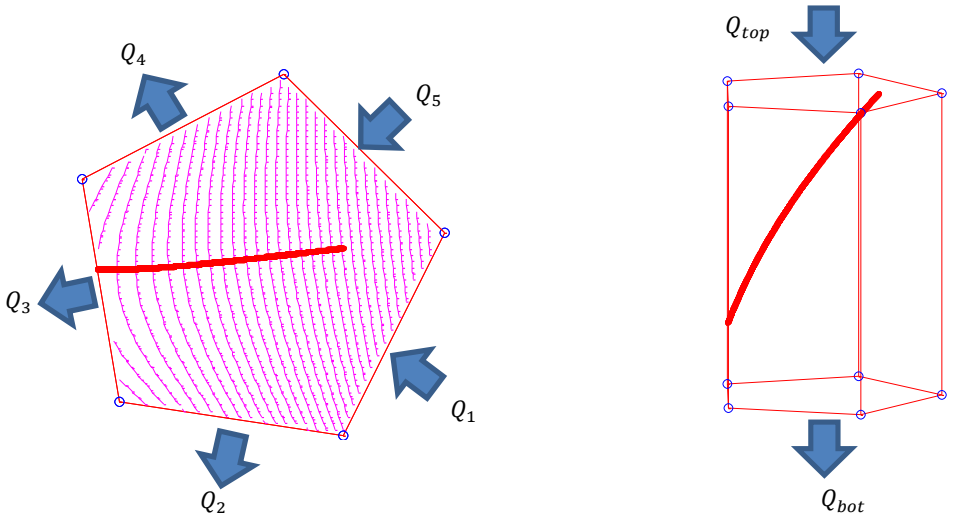
the vertical flux part of Equation 3.18 only includes the bottom vertical flux Q_{bot} because flows in through the top of the cell will always be on above the particle location, whereas the flows from the bottom of the cell will always be below the particle location, and thus satisfying the preliminary assumption regarding the vertical velocity calculation.

3.4.1 Steady State

The requirement to solve the steady state flow problem is that the saturated thickness of the cell is unchanged over the whole time step. When the change in saturated thickness is zero, Equation 3.18 can be simplified into the following:

$$v_z(z_p, t) = \frac{Q_{bot}}{\theta A} + \frac{z_p - z_b}{\theta A(b^n)} \sum_{i=1}^{n_{faces}} Q_i \tag{3.19}$$

After calculating the vertical velocity of the particle, it is then moved to the next location until the particle pathlines distribution is created within the cell. The estimation of pathlines of the particle is discussed in the next section. Figure 3.8 represents an example of a pathline within a cell with steady state flow case.



$$Q_i = [25.4528;\ -17.4955;\ -12.1743;\ -17.9065;\ 22.1235\,],$$
$$Q_{top} = -25\,,\ Q_{bot} = -25$$

Figure 3.8: Example of Steady State Flow Case, Where the Particle Exits the Cell through One of the Sides

The example problem in Figure 3.8 has zero horizontal net flow and zero change in storage. The image on the left shows the particle tracking in the plan view and the image on the right shows the tracking in the cross-section view. The particle exits the cell through one of the sides. The units for the flows in through the sides, top, and bottom of the cell are [L^3/T]. The horizontal net flow term that is used for calculating the vertical flow is the horizontal cell flow rates parameter (i.e. Q_i). The particle tracking algorithm for the vertical velocities in steady state may be treated similarly with the Pollock method because each cell must be treated, assuming that the bottom boundary of the cell is not connected to multiple cells.

3.4.2 Transient Flow

Transient flow occurs in a groundwater flow system when there is a change in water storage with time due to water accumulation or removal. In MODPATH, the transient flow is treated as a series of steady-state flow periods. However, the water table presents a problem in a transient flow system as it behaves as a moving boundary. MODPATH deals with this problem by assuming that the water table moves in discrete jumps from one time step to the next. Similar to the steady state flow problem, the key assumption for the transient case is that only the flows below the particle location will impact the particle position. Four example cases for a falling water table are shown in Figure 3.9:

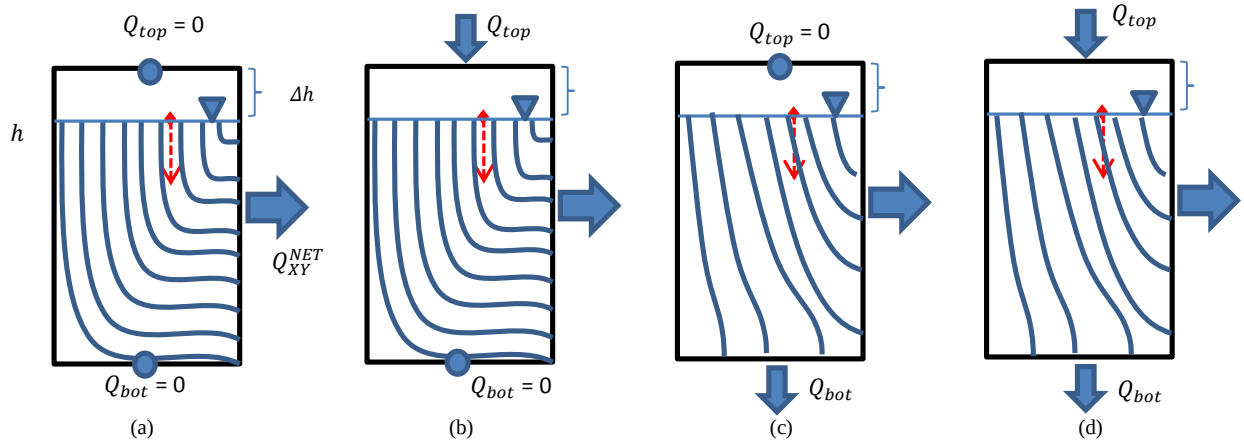


Figure 3.9: The Four Transient Cases, from the Left Side: Case (a) Falling Head with No Recharge and No Vertical Leakage, (b) Falling Head with Recharge and No Vertical Leakage, (c) Falling Head with No Recharge and Vertical Leakage, and (d) Falling Head with Recharge and Vertical Leakage; Streamlines are Approximated and for Rough Visualization Only

While dealing with these cases, the assumption made regarding steady state where $\Delta b = 0$ is no longer valid. Figure 3.9 shows the four cases for the vertical flux where the storage of the cell changes over time. Case (b) assumes that there is no vertical leakage (Q_{bot} =0) but there is a vertical recharge from the top of the cell. Depending on the value of vertical recharge and horizontal leakage of the cell, the water table can decrease, increase, or stay the same. However, as stated in the previous assumption, an increase in water table will not affect the particle movement in the absence of other flows. Therefore, the vertical movement of the particle will only be directly affected when the water table is decreased. Case (c) is similar to Case (a) but with a vertical leakage within the cell. This vertical leakage will directly affect the change in storage and thus the vertical movement of the particle in the cell. Case (d) is similar to Case (c) but with an additional vertical recharge within the cell. The vertical recharge will affect the change in storage within the cell, but it will not directly impact the vertical movement of the particle. Figure 3.10 shows an example of particle pathlines in a transient flow case.

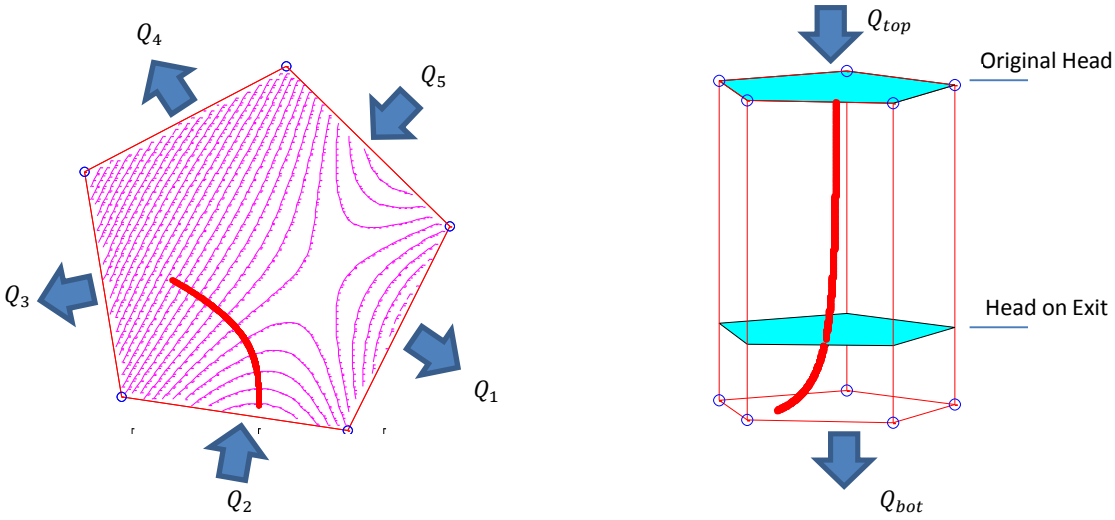


Figure 3.10: Example of Transient Flow Case, Where the Particle Exits the Cell through the Bottom

The example within the figure contains $Q_{vert} = -50\frac{L^3}{T}$. The pathlines of the particle within the cell show that the particle exits the cell through the bottom face. The top blue part of the right image of Figure 3.10 shows the initial water table at the beginning of the time step. The bottom blue part is the water table elevation when the particle exits the cell. The end water table elevation does not represent the elevation at the end of a MODFLOW time step, just the elevation when the particle exits the cell. Here, upward

vertical flow direction is considered positive while downward flow direction is negative, but net horizontal flow direction into the cell is positive while flow going out of the cell is considered negative. Over each time step, the average vertical flux is constant but the water table elevation varies linearly in time. The cross sectional area of the cell and the average porosity are assumed to be constant throughout any time step. Another important assumption is that the increase in water table within the cell due to the water source from above the particle will not directly affect the movement of the particle.

3.5 Pathline Integration

After obtaining the horizontal and vertical velocities for the cell, there are different schemes available to create the particle pathlines in each cell. By using pathline integration, the particle's path as it moves through the flow model may be determined. In mod-PATH3DU input files, the user has the option of choosing one of two types of pathlines distribution to track a particle within an unstructured cell: Euler or fourth-order Runge-Kutta schemes (Zheng, 1989; Zheng, 1992). The basic idea of forward and backward Euler scheme is that given the initial position of a particle and time step or space step, the next position of the particle may be determined. Using the Euler method, the particle pathlines may be generated with either time or space step. Each particle pathline obtained with the Euler scheme is determined by using either fixed space step or fixed time step. Figure 3.11 represents the particle pathlines in a cell that vary spatially and temporally.

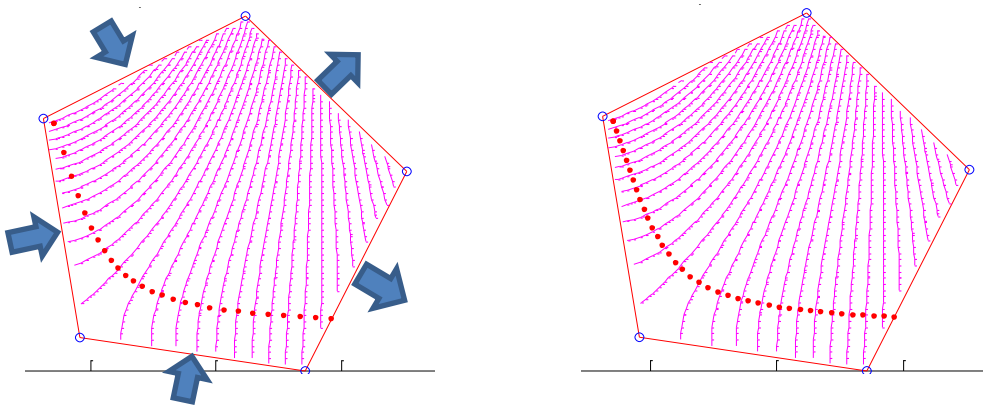


Figure 3.11: Particle Pathlines Integration with Euler Scheme for Fixed Time Step (Left) and Fixed Space Step (Right)

The left image of Figure 3.11 depicts the particle pathlines integration using fixed time step option and the right image uses fixed space step option. As it is shown in the right image, the distance between one particle position and the next is similar. Depending on the chosen time step or space step, the time and the location where the particle exits the cell may vary significantly. The advantage of using the fixed space step over the fixed time step option is that the particle pathlines do not get affected by the velocity variation in the conceptual model and are therefore more accurate even for large head gradients. However, it may require more computation if the fixed space step value is too small. The Euler method for calculating the next position of a particle using fixed time step and space step is as follows:

$$x_p^{n+1} = x_p^n + v_x^n \, \Delta t, \qquad y_p^{n+1} = y_p^n + v_y^n \, \Delta t \tag{3.20a}$$

$$x_p^{n+1} = x_p^n + \frac{v_x^n}{|v_n|} \Delta x, \qquad y_p^{n+1} = y_p^n + \frac{v_y^n}{|v_n|} \Delta y \tag{3.20b}$$

Where x_p^{n+1} is the next particle position in x-coordinate; x_p^n is the current particle position in x-coordinate; v_x^n is the current particle velocity in the x-direction; $|v_n|$ is the absolute value of the current complex horizontal velocity of the particle such that $|v_n| = \sqrt{{v_x}^2 + {v_y}^2}$; Δt is the fixed time step; and Δx is the fixed space step. Using the Euler scheme, the next particle position can be computed using either Equations 3.20a or 3.20b.The first-order Euler tracking scheme is simple and easy to simulate, but the end result of the pathlines may not be accurate if the time step or space step chosen during a simulation is too high. Therefore, using a higher order scheme for tracking particles may be more desirable.

To increase the accuracy of the calculations regarding the particle tracking relative to Euler scheme, the fourth-order Runge-Kutta scheme may also be implemented. The basic principle of this scheme is to move the particle from its initial position over a time interval by combining the results from several trial steps. The equations and full steps of fourth-order Runge-Kutta scheme are available in mod-PATH3DU User's Guidelines. The information gathered from those steps may then be used to for a fourth-order Taylor series expansion. Using this scheme, the algorithm can automatically find a more

optimal particle pathlines within the conceptual model. The accuracy of the Runge-Kutta scheme depends on the tracking time step size. As with Euler scheme, if the time step is too large, the particle pathline may be inaccurate by diverting from the actual flow path. Similarly, if the time step is too small, the simulation may require a significant computational effort to move the particle over a given distance within the conceptual model. By comparing the results between the full time step trial and the half time step trial, the adaptive step size control can be implemented within the algorithm; i.e., tracking time step is taken twice, once as a full step and once as two half steps. Using this adaptive time step procedure, the time step for each iteration in the simulation may vary. This way, the particle pathlines may be generated with high accuracy and as little computational time as possible. The equation for the fourth order Runge-Kutta scheme that the Waterloo method uses for calculating the next position of a particle is as follows:

$$x_p^{n+1} = x_p^n + \frac{(k_1 + 2k_2 + 2k_3 + k_4)}{6} \tag{3.21a}$$

$$y_p^{n+1} = y_p^n + \frac{(l_1 + 2l_2 + 2l_3 + l_4)}{6} \tag{3.21b}$$

$$z_p^{n+1} = z_p^n + \frac{(m_1 + 2m_2 + 2m_3 + m_4)}{6} \tag{3.21c}$$

where

$$\begin{aligned} k_1 &= \Delta t\, v_x\left(x_p^n, y_p^n, z_p^n\right) \\ k_2 &= \Delta t\, v_x\left(x_p^n + \frac{k_1}{2}, y_p^n + \frac{l_1}{2}, z_p^n + \frac{m_1}{2}, t_p + \frac{\Delta t}{2}\right) \\ k_3 &= \Delta t\, v_x\left(x_p^n + \frac{k_2}{2}, y_p^n + \frac{l_2}{2}, z_p^n + \frac{m_2}{2}, t_p + \frac{\Delta t}{2}\right) \\ k_4 &= \Delta t\, v_x\left(x_p^n + k_3, y_p^n + l_3, z_p^n + m_3, t_p + \Delta t\right) \end{aligned} \tag{3.22}$$

$$\begin{aligned} l_1 &= \Delta t\, v_y\left(x_p^n, y_p^n, z_p^n\right) \\ l_2 &= \Delta t\, v_y\left(x_p^n + \frac{k_1}{2}, y_p^n + \frac{l_1}{2}, z_p^n + \frac{m_1}{2}, t_p + \frac{\Delta t}{2}\right) \\ l_3 &= \Delta t\, v_y\left(x_p^n + \frac{k_2}{2}, y_p^n + \frac{l_2}{2}, z_p^n + \frac{m_2}{2}, t_p + \frac{\Delta t}{2}\right) \\ l_4 &= \Delta t\, v_y\left(x_p^n + k_3, y_p^n + l_3, z_p^n + m_3, t_p + \Delta t\right) \end{aligned} \tag{3.23}$$

$$
\begin{aligned}
m_1 &= \Delta t\ v_z\left(x_p^n, y_p^n, z_p^n\right) \\
m_2 &= \Delta t\ v_z\left(x_p^n + \frac{k_1}{2}, y_p^n + \frac{l_1}{2}, z_p^n + \frac{m_1}{2}, t_p + \frac{\Delta t}{2}\right) \\
m_3 &= \Delta t\ v_z\left(x_p^n + \frac{k_2}{2}, y_p^n + \frac{l_2}{2}, z_p^n + \frac{m_2}{2}, t_p + \frac{\Delta t}{2}\right) \\
m_4 &= \Delta t\ v_z\left(x_p^n + k_3, y_p^n + l_3, z_p^n + m_3, t_p + \Delta t\right)
\end{aligned}
\tag{3.24}
$$

Equations 3.21a, b, and c, 3.22, 3.23, and 3.24 above are taken directly from mod-PATH3DU User's Guide (Muffels *et al.*, 2014). t_p in the equations above is the tracking time of the particle, and Δt is the tracking time interval. By combining the information associated with the several trial steps in the fourth order Runge-Kutta scheme, the particle may be advanced from its initial position to the next. In comparison to the Euler scheme and Pollock's method linear interpolation, the fourth order Runge-Kutta scheme may generally be more computationally intensive. However, the Runge-Kutta method is applicable to many cases, including any velocity interpolation scheme.

The advantages of using Euler scheme is that the user has more freedom in choosing fixed time or space step between the particle locations. However, it requires more knowledge about the groundwater model and the cell-by-cell flows. Using the fourth-order Runge-Kutta scheme with the adaptive time step size control procedure, the particle pathlines may be accurately generated, while still maintaining a reasonable computational time for the particle tracking simulation. In comparison with the Pollock semi-analytical method (Pollock 1994), the Runge-Kutta scheme may require intensive computational time with the possibility of numerical truncation errors. However, the Runge-Kutta method is generally more applicable to any velocity interpolation scheme. Using the Waterloo method in mod-PATH3DU software, the users may choose between the Euler and fourth-order Runge-Kutta schemes during their simulation by modifying the input file. The integration of pathlines is one of the sources of errors within the algorithm. Selecting the time step (or space step) for the pathlines distribution may determine where and when the particle may end up in the conceptual model.

3.6 Sources of Errors

As mentioned earlier in this chapter, there are several sources of velocity field reconstruction errors in Waterloo method: (1) the primary errors are truncation errors, where there are not enough terms in the Taylor series to represent the flow field; and (2) fitting errors, where discontinuities in the boundary conditions of the cell cannot be fitted properly; these errors are both boundary errors. In addition, tracking errors may be caused by insufficient temporal discretization used in pathline integration. These boundary and tracking errors are controlled by the total number of control points (M), the order of approximation (N), and the fixed time or space step chosen for the pathline integration. The errors obtained from the fixed time or space step parameter only affect the pathline integration of the particle tracking. However, the number of control points and the order of approximation chosen for each cell determine the ability to generate the complex velocity field within the cell.

The accuracy of the particle tracking depends on the tracking step size; too large of step size parameter may divert the actual flow path of the particle. A large fixed time or space step may affect the total time required for particle to exit the cell, and the location of the particle exit point. However, small time step may increase computational time of the simulation. Therefore, the algorithm implements the adaptive step size control procedure (Zheng, 1989; Zheng, 1992; Zheng, 1994; Zheng and Bennett, 2002). Using the adaptive time step, the tracking errors due to fixed time or space step may be decreased significantly.

The error in boundary condition includes a combination of truncation error and Gibbs phenomenon. The net impact of fitting and truncation error within the cell may be calculated using the following equation:

$$\in = \frac{1}{M} \sum_{i=0}^{M} \frac{\left| Q_{spec}^{i} - Q_{\eta}^{i} \right|}{\max \left| Q_{spec} \right|} \tag{3.25}$$

Where $\in$ is the normalized absolute average error within the cell, evaluated at points in between control point location; M is the total number of control points along the cell perimeter; Q_{spec} is the specified

normal integrated discharge obtained from the total corrected inflows at every side of the cell per length of each side of the cell; and Q_η^i is the normalized integrated discharge at each evaluation point i. Equation 3.25 is a measure of the fitting error obtained by comparing the calculated normalized flux and the specified normalized flux from the total corrected inflows. Figure 3.12 shows examples of the particle tracking simulation with different values of order of approximation and total number of control points. As mentioned earlier in the previous section, the basic rule regarding the values of order of approximation and total number of control points is that $M \geq 2N$. If the total control points is less than 2N, the flow pattern inside the cell will not be created and the pathline integration will be absent from the cell. For simplicity purposes, the total control points are set at least two times the order of approximation for each example. Thus, the only variable that is directly changed for these examples is the order of approximation of the cell.

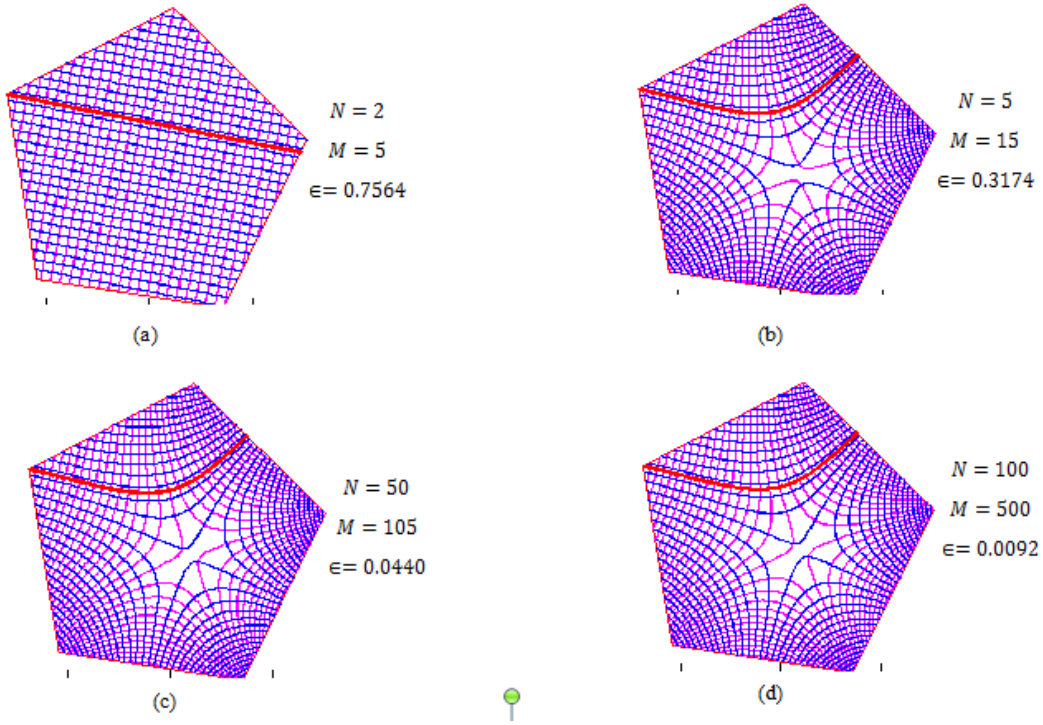


Figure 3.12: Examples of the particle tracking within a cell with different values for the total number of control points and order of approximation of the cell.

The simulations in Figure 3.12 use a fixed time step of $\Delta t = 0.001$. There is no vertical or well flux within the cell for simplicity purposes. As shown in Figure 3.12, changing the order of approximation

affects both the error and the complex flow fields within the cell, and hence the total time and exit location of the particle tracking. It is desirable to determine an optimum value for both the total number of control points and the order of approximation for the cell in order to ensure the smallest value for the average velocity errors within the cell as well as reasonable computational time for the simulation. Figure 3.13 depicts the boundary values of each example shown in Figure 3.12 arranged downward from simulation (a) to (d).

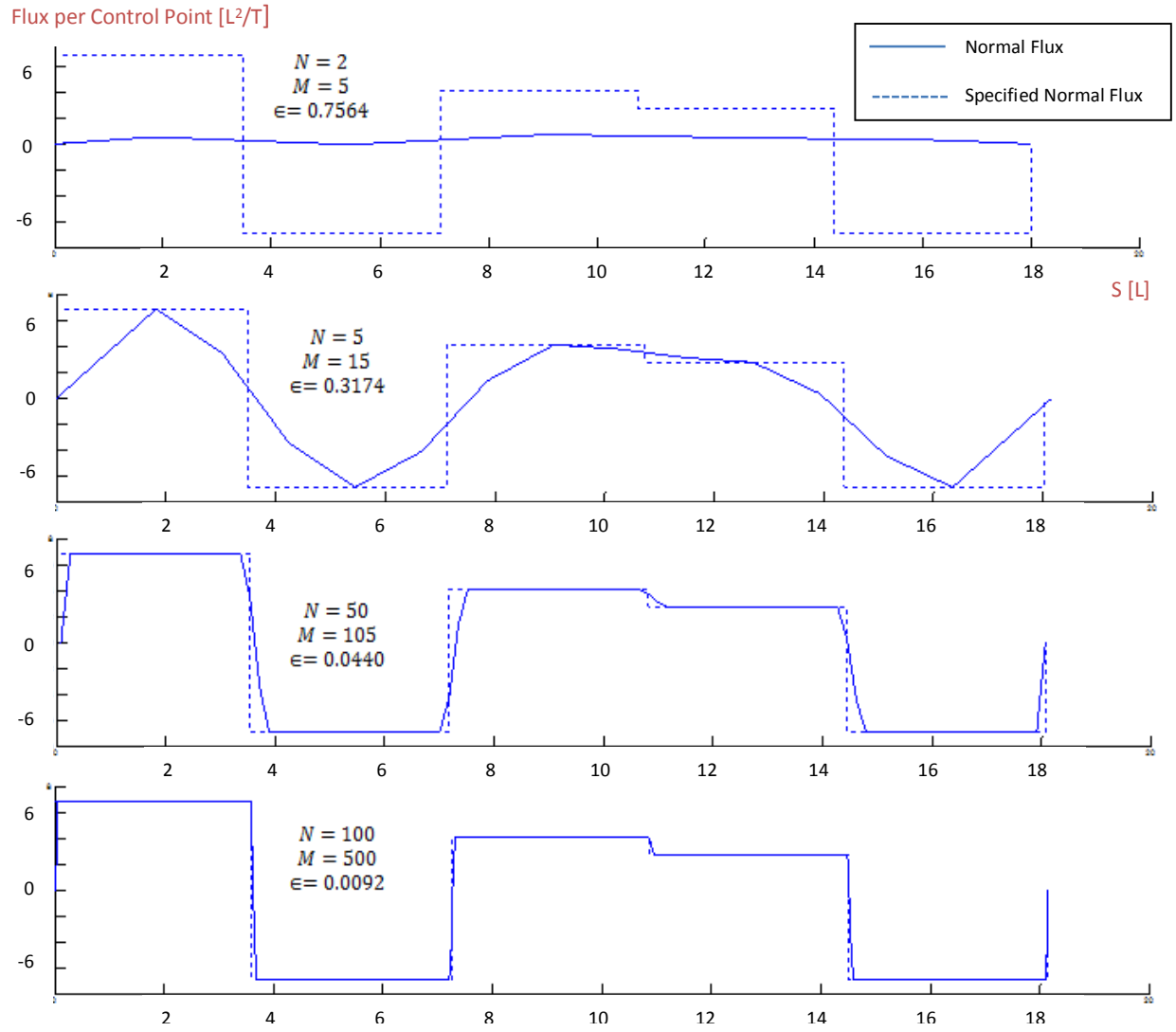


Figure 3.13: The Specified and Normal Fluxes at Cell Boundaries for the Simulations in Figures 3.12

When determining on how to reduce errors, the goal is to match the normal fluxes obtained from the coefficients in the algorithm that is represented as the dotted blue lines with the specified normal flux from the input files that is represented as the solid blue lines in Figure 3.13. By choosing a large value

for the total number of control points and order of approximation within the cell, the error may be decreased. However, choosing large values for these parameters means increasing the computational cost of the particle tracking simulation.

The other type of error, as part of boundary error, that is encountered in the algorithm is the one called Gibbs phenomenon, where the values of a partial sum expansion of a function appear to be persistently overshot or undershot near a jump discontinuity, compared to the values in the original function (Raeen, 2008). The fitting or boundary error of a cell can be reduced by choosing an optimum M and N of the cell; no matter how large these values are the fitting error may not be completely eliminated from any simulation. Using an adaptive time step or even a very small fixed or space step does not eliminate the tracking error for the cell. Figure 3.14 shows an example of Gibbs phenomenon induced symptoms in the particle tracking algorithm of the Waterloo method:

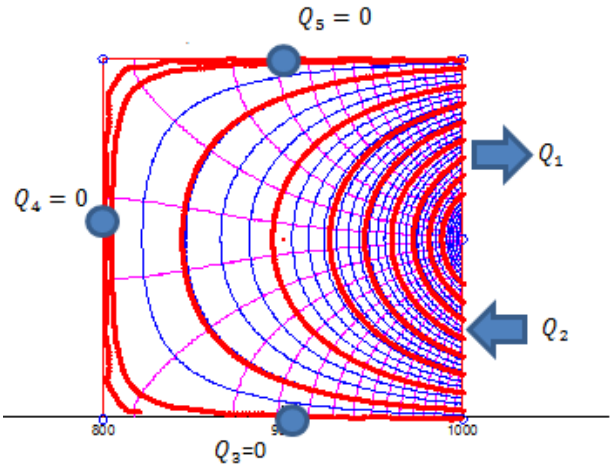


Figure 3.14: Example of Gibbs phenomenon during particle tracking using Waterloo method

The Gibbs phenomenon in Figure 3.14 is shown near the top, bottom, and left side of the cell, where the flows in through those sides are zero. The phenomenon is part of truncation error of particle tracking for this algorithm and it may affect the total time travel and particle end point location if the travel path of the particle is within the zone of the phenomenon. The solution to this issue is not within the scope of this thesis and may be addressed in the future.

3.7 Summary of Methodology

To summarize this chapter, a flow chart is provided to better understand the steps for determining the particle pathlines using the Waterloo method. The flow chart is provided in Figure 3.15. The first thing that the user needs to do is populate all of the necessary parameters in order to start the preprocessing procedure for the algorithm. The parameters are available from MODFLOW input files and the grid specification file from the third party mesh-generating software (.DISU, .GSF, and .CBB files). The order of approximation and the total number of control points need to be specified. These two internal parameters determine the spatial and temporal accuracy of the particle tracking, as well as the computational time for the whole simulation. These two parameters are included in the primary parameters of the cell as shown in Table 3.1. The secondary parameters (Table 3.2), the cell and flow control parameters (Tables 3.3 and 3.6), and the unit stream function are needed in order to determine the Taylor series coefficients for the cell. These coefficients are essential in determining the flow pattern inside the cell (Figure 3.5), as well as the horizontal complex velocities for the cell. Calculating the flow pattern inside the cell for stream and potential lines using Equation 3.9 is not necessary in order to determine the velocity of the particle. The calculation of vertical velocity of the particle is independent of the calculation of horizontal velocities. After obtaining the velocities of the particle, it may then be moved to the next location within the cell. By repeating the step mentioned above, the algorithm can either use Euler or fourth-order Runge-Kutta scheme to create the particle pathlines for the conceptual model. The adaptive time step for the fourth-order Runge-Kutta scheme is recommended because it can significantly reduce the tracking error of the particle. The particle(s) movements are written into the output file of mod-PATH3DU, as well as the total time travel of each particle through the conceptual model.

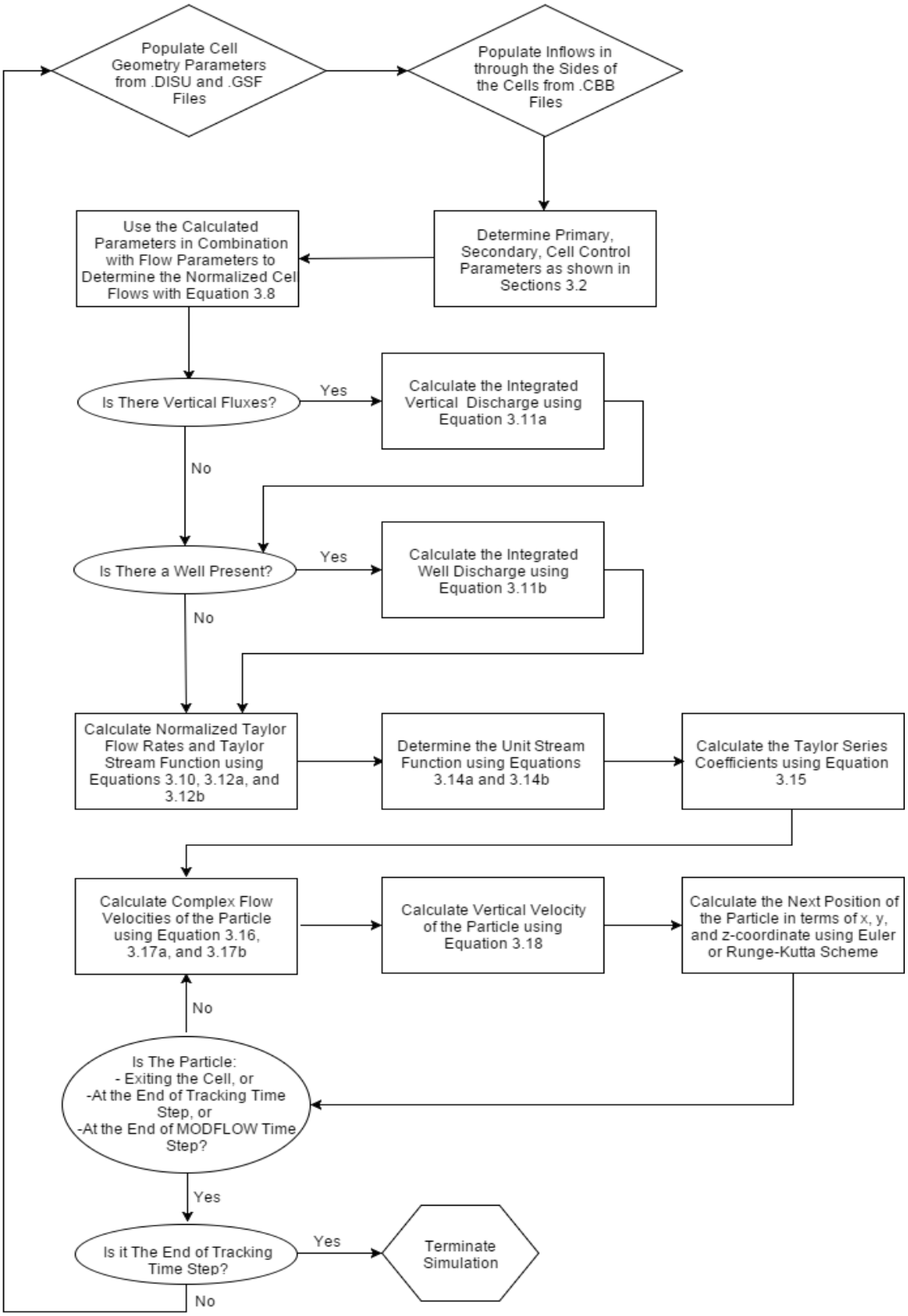


Figure 3.15: The Flow Chart of Waterloo Method's Methodology

Chapter 4

Results and Discussions

In this chapter, various test cases are used to thoroughly illustrate and assess the effectiveness of the method described in Chapter 3. Impact of cell geometry, truncation error, overspecification ratio, and tracking time step are assessed. The Waterloo method is contrasted and compared to other available methods.

4.1 Comparison to Pollock Method

As mentioned earlier in this thesis, the Pollock method (Pollock, 1988) is a semi-analytical particle tracking method that has been widely used. The key characteristic of this method is that given the initial position of a particle anywhere within the cell, the next position of the particle and the time travel between them can be computed directly without numerical integration. Furthermore, the Pollock method determines the particle pathlines within a cell by using linear interpolation of the velocities sourced from the cell-by-cell flows that are generated from MODFLOW. However, the Pollock method is still limited to track particles within rectilinear grids, even with the new extension (Pollock, 2015) for handling nested grids and quad-based grids. Here the Waterloo method is benchmarked against the Pollock method using only rectilinear grid cells. Figure 4.1 represents example of a case that will be handled by the Waterloo method and the Pollock method with the same geometry, inflows in through the sides of the cell, and particle starting point; example 4.1. Figure 4.1 shows the geometry and cell inflows of the case example that will be run by the Waterloo method and the Pollock method. The dx value for this example is $x_2 - x_1 = 1.5$; the dy value is $y_2 - y_1 = 1.0$; and the porosity for this example 4.1 is 0.3.

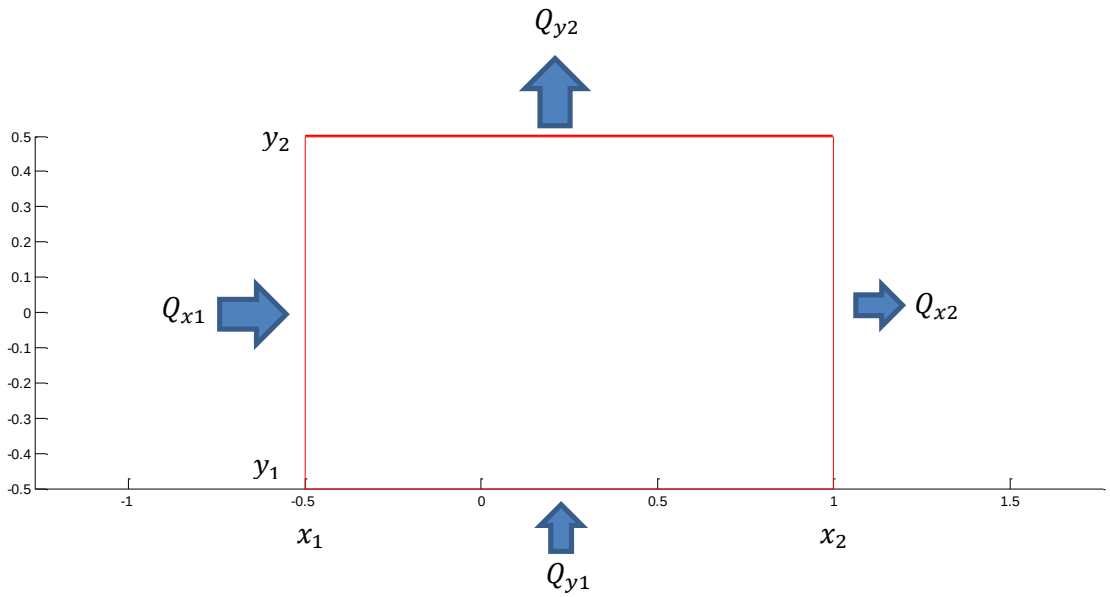


Figure 4.1: An Example of a Case with the Geometry and Cell Inflows of a Cell

The primary and flow parameters of example 4.1 are reported in Table 4.1.This is a simple example that is included in this chapter to benchmark the Waterloo method against the Pollock method. These primary parameter values were chosen to create a simple two-dimensional rectilinear cell with fluxes from all of the cell's sides.

Table 4.7: Primary and Flow Parameters to Generate Example 4.1

Parameter Symbols	**Example Values**
n_{faces}	4
$[x_i \quad y_i]$	$\begin{bmatrix} -0.5 & -0.5 \\ -0.5 & 0.5 \\ 1 & 0.5 \\ 1 & -0.5 \end{bmatrix}$
$[x_c \quad y_c]$	$[0.25 \quad 0]$
N	30
M	80
θ	0.3
$Q_i[\mathrm{L}^3/\mathrm{T}]$	$\begin{bmatrix} 13 \\ -13 \\ -2 \\ 2 \end{bmatrix}^T$

The sign convention of the horizontal inflows for the Waterloo method is different from the Pollock method; positive for flows going into the cell, and negative for flows going out of the cell. The particle pathline for a starting point of $(x_p \quad y_p) = (-0.25 \quad -0.5)$ is shown in Figure 4.2 for both methods.

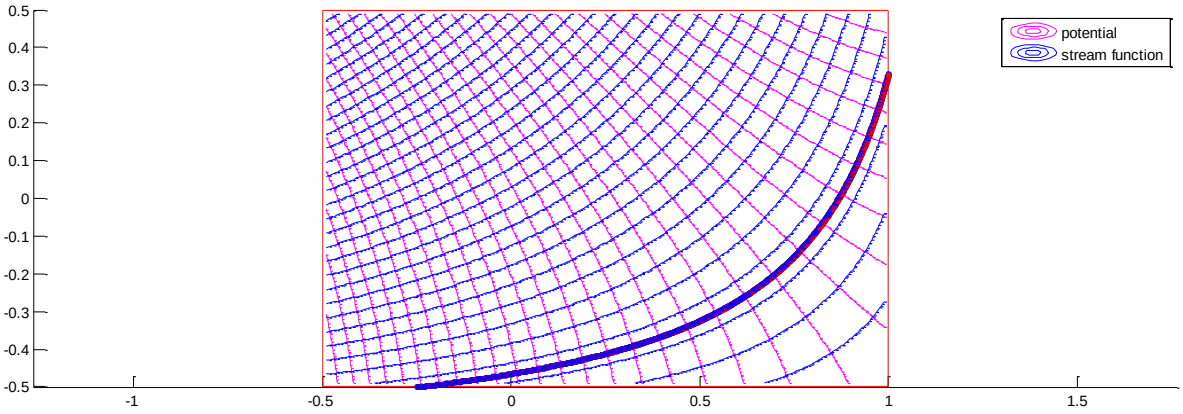


Figure 4.2: The Particle Pathlines Generated using the Waterloo Method (Red Pathline) and the Pollock Method (Blue Pathline) in a Rectilinear Cell using Similar Parameters

Figure 4.2 shows the particle pathlines generated using the Waterloo method and the Pollock method. The Waterloo method uses a fixed integration time step of $\Delta t = 0.0001$, and the Pollock method is plotted at intervals of the same Δt. As shown in Figure 4.2, the difference between the Waterloo and Pollock method particle pathline is not visually discernible. Since the pathlines for both methods are very close together, one seems to overlay the other. The red pathline is generated using the Waterloo method, and the blue pathline is generated using the Pollock method, using a fixed time step. The normalized absolute average error for this simulation is 0.0231. The particle end result for the example shown in Figure 4.1 is represented in Table 4.2.

Table 4.8: The Particle Exit Point and the Time Travel for the Waterloo Method and the Pollock Method for Example 4.1

Example 4.1	Pollock	Waterloo	% Difference
Xp exit	1	1	0%
Yp exit	0.3333	0.3312	0.63 %
Travel Time	0.0704	0.0706	0.28 %

As shown in Table 4.1, the particle exit position in terms of the x-axis is the same because both particles exit through the same face. In terms of y-axis, the difference between both methods is less than 1%. Similarly, the travel time of the simulation for the particle from the starting point to the exit point has less than 1% difference. The particle exit location and the time travel between them depend on the order of approximation, number of control points per side, and the algorithm fixed time step. The fixed space step option is not implemented for this or other examples in this section. By changing one or more of these primary parameters, the particle exit location and the travel time between them for the two methods may vary significantly. The parameters of example 4.1 are chosen solely to assess whether or not the method works; a very small fixed time step, and large N and M are used to minimize the boundary and tracking errors within the cell.

Another example to compare the end result between the two methods is shown in Figure 4.3. Example 4.2 shows two of the sides within the cell having no flow in through them.

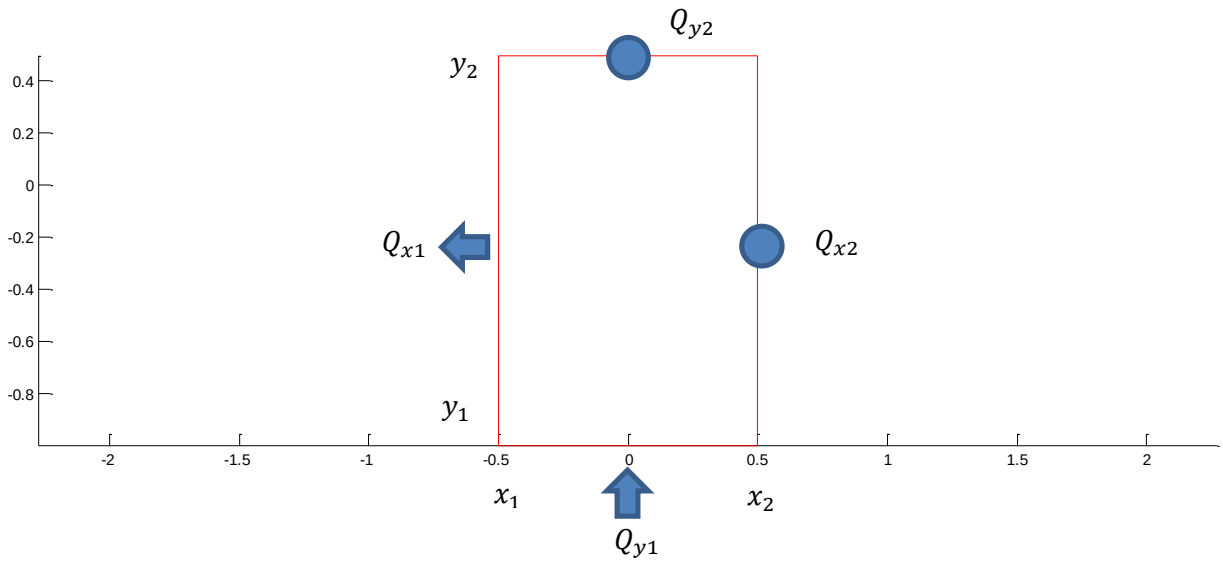


Figure 4.3: An Example of a Special Case Handled by the Waterloo Method and the Pollock Method

This type of example represents the hardest special case for rectilinear geometry because it has the highest boundary error due to Gibbs phenomenon; by being able to solve this case, the method is shown to solve the other cases with various other horizontal cell-by-cell inflows. Similar to example 4.1, example 4.2

only contains horizontal inflows; there is no vertical and well term within the cell. This is to keep the simplicity of the example and to show the horizontal velocity distribution created by the Waterloo method. The primary parameters for example 4.2 are slightly different than those of example 4.1 just to show some variations in the cell geometry. The primary and cell flow parameters for example 4.2 is shown in Table 4.3The examples shown thus far are to compare the horizontal particle pathlines between the two methods; i.e. the inflows in through the top and bottom of the cell are zero.

Table 4.9: The Primary and Cell Flow Parameters of Example 4.2

Parameter Symbols	**Example Values**
n_{faces}	4
$[x_i \quad y_i]$	$\begin{bmatrix} -0.5 & -1 \\ -0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & -1 \end{bmatrix}$
$[x_c \quad y_c]$	$[0 \quad -0.25]$
N	30
M	80
θ	0.3
Q_i	$\begin{bmatrix} -3 \\ 0 \\ 0 \\ 3 \end{bmatrix}^T$

Similar to Example 4.1, both methods use a very small fixed time step of $\Delta t = 0.0001$, with a large order of approximation and total control points of the cell to reduce any boundary and tracking error in the cell. The flow pattern inside the cell and the particle pathlines for both methods are represented in Figure 4.4.

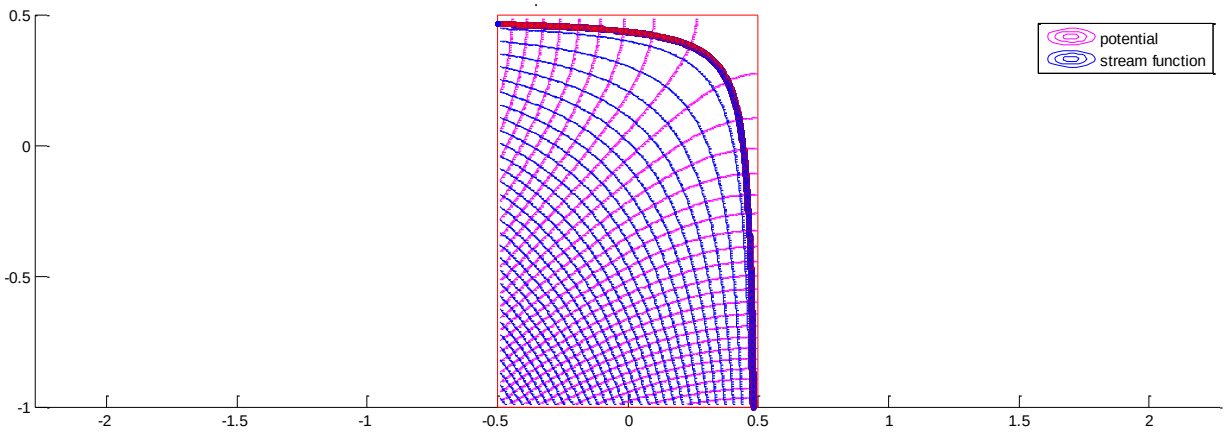


Figure 4.4: The Flow Pattern and Particle Pathlines of Example in Figure 4.3 Generated using the Waterloo Method (Red Pathline) and the Pollock Method (Blue Pathline)

The particle starting point for both methods is $(x_p \quad y_p) = (0.48 \quad -1)$. The normalized absolute average error for this simulation is 0.0278. This error is shown to measure the how accurate the flow field for the cell is. Similar to example 4.1, the difference between the two pathlines is very small that one seems to overlay the other. The particle end result for example 4.2 shown in Figure 4.4 is represented in Table 4.2:

Table 4.10: The Particle Exit Point and the Time Travel for the Waterloo Method and the Pollock Method for Example 4.2

Example 4.2	Pollock	Waterloo	% Difference
Xp exit	-0.5	-0.5	0%
Yp exit	0.4700	0.4704	0.08 %
Travel Time	0.5868	0.5884	0.27 %

As mentioned earlier in this section, the end result for the Waterloo method highly depends on the parameters such as algorithm fixed time step, order of approximation, and number of control points. However, the users may manually determine the fixed time step between each particle location if they are using Euler scheme; the fourth-order Runge-Kutta scheme in the algorithm is applied using an adaptive

time step. Since the differences between the particle end results obtained using the Waterloo method and the Pollock method are very small on both examples 4.1 and 4.2, it is safe to say that the Waterloo method can handle the horizontal particle tracking in rectilinear cells within a comparable accuracy to the Pollock method. However, the Pollock method still has much lower computational cost compared to the Waterloo method.

The last example that is presented in this section for the comparison between the Waterloo method and the Pollock method is depicted in Figure 4.5. Example 4.3 shows the comparisons between the two methods after adding vertical flows. The dx value for this example is $x_2 - x_1 = 1.0$; the dy value is $y_2 - y_1 = 1.0$; dz value is $z_2 - z_1 = 1.0$; and the porosity for this example 4.3 is 0.3. The value for the porosity can be any value between, but not including, 0.0 and 1.0. Similar to previous examples in this section, the parameters are chosen to be simple and easily replicable. Because of the Dupuit-Forchheimer approximation that allows the calculation for horizontal velocities to be independent of the calculation for vertical velocity, the primary parameters for example 4.3 are reported in two-dimensions. The number of faces for this example is 4; the top and bottom faces of the cell are not necessary in order to determine the horizontal velocities of the cell.

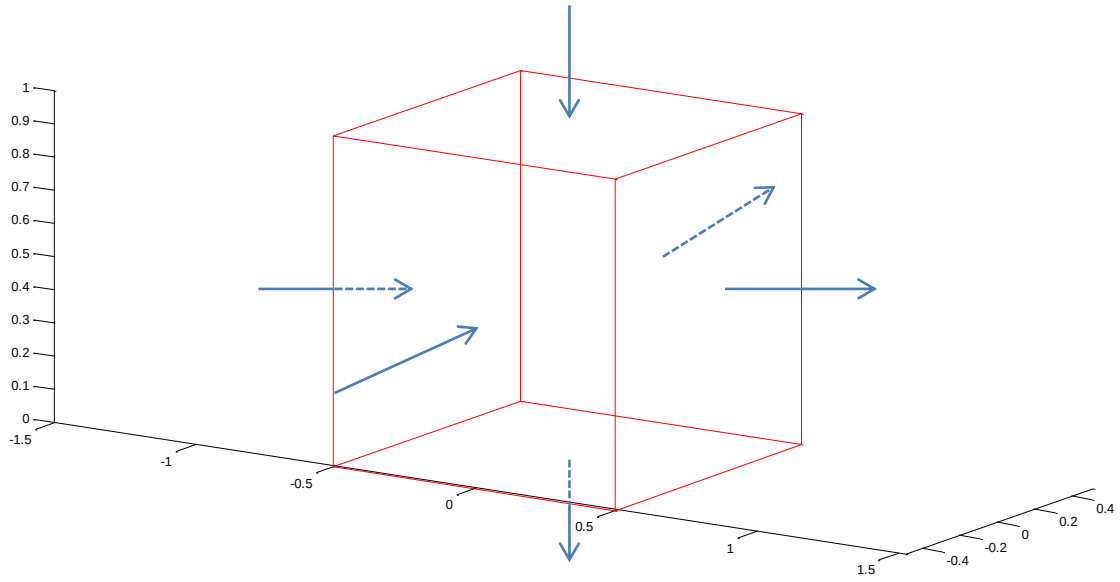


Figure 4.5: Example of a Case with Vertical Velocity Distribution to be handled by the Waterloo Method and the Pollock Method

The last example shown in Figure 4.5 is for the comparison between the two methods when dealing with vertical velocity within the cell. The calculation regarding the vertical velocity at any point within a cell for the Waterloo method is presented in the methodology section, and is independent of the calculations for the horizontal velocity. The primary and flow parameters for example 4.3 is shown in Table 4.5. These parameters are slightly different than the previous examples in this section to show more cell and flow variations to be handled by the two methods.

Table 4.11: The Primary and Flow Parameters of Example 4.3

Parameter Symbols	**Example Values**
n_{faces}	4
$[x_i \quad y_i]$	$\begin{bmatrix} -0.5 & -0.5 \\ -0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}$
$[x_c \quad y_c]$	$[0 \quad 0]$
N	20
M	80
θ	0.3
Q_i	$\begin{bmatrix} 5 \\ -2 \\ -3 \\ 4 \end{bmatrix}^T$
z_{top}	1
z_{bot}	0
Q_{top}	-3
Q_{bot}	-7

The case assumes steady-state flows such that the water table within the cell does not change over time. z_{top} and z_{bot} are the Cartesian coordinate of the top and bottom of the cell respectively. Note that the

number of faces is stated as 4; this is because this parameter does not include the top and bottom face, in order to determine the horizontal velocities of the cell. The flow pattern and the particle pathlines integration of example 4.3 is shown in Figure 4.6.

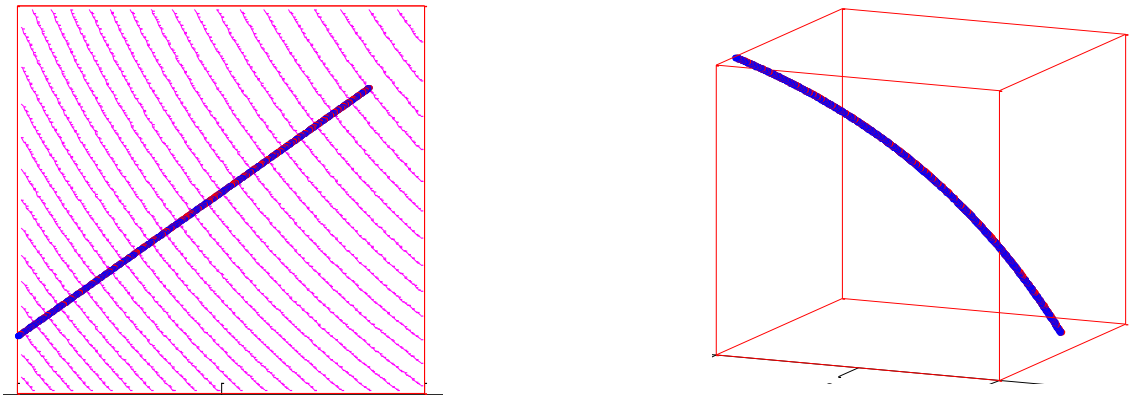


Figure 4.6: The Particle Pathlines Distribution in a Cell with Vertical Velocity Generated using the Waterloo Method (Red Pathline) and the Pollock Method (Blue Pathline) for Example 4.3; in Plan View (Left) and 3D (Right)

In example 4.3, the starting point of the particles is $(x_p \quad y_p \quad z_p) = (-0.5 \quad -0.35 \quad 1)$. Since there is a vertical term in the cell, the stream functions in the cell are undefined. Note that the sign convention of the inflows parameter stated here uses the Pollock method standard; positive flows toward the right along x-axis; toward the front along y-axis; and upward along the z-axis. The Waterloo method flows sign convention is positive for flows into the cell and negative for the flows going out of the cell. The vertical flows for the Waterloo method are positive upward and negative downward. Similar to the other examples in this section, example 4.3 contains pathlines generated from both methods to be nearly identical. The particle end location and time travel between them for both methods are shown in Table 4.3.

Table 4.12: The Particle Exit Point and the Time Travel for the Waterloo Method and the Pollock Method for Example 4.3

Example 4.3	Pollock	Waterloo	% Difference
Xp exit	0.3634	0.3641	0.19%
Yp exit	0.2889	0.2896	0.24%
Zp exit	0	0	0%
Travel Time	0.0635	0.0636	0.16%

As shown in Table 4.3, the spatial and temporal differences between the two methods are still less than 1% for cells with vertical velocity. The particles for both methods exit the cell from the same face: the bottom of the cell, hence Zp exit % difference is 0. Note that this test indicates that the mass balance formulation used to calculate vertical velocity under steady conditions, as shown in Equation 3.19, and the vertical influx term used in generating the horizontal velocities are both valid and correctly implemented. The comparison between the Waterloo method and the Pollock method within a cell with a pumping or injecting well in it is not presented in this thesis. This is due to Pollock method not being able to handle "weak sinks" within a cell, where some particles entering a particular cell may enter the sink or source and some other particles may bypass the sink, leaving through a cell face. Details regarding the ability of the Waterloo method to handle "strong sinks" and "weak sinks" are discussed later in this chapter.

4.2 Applications to Moderately Complex Flow Fields

As mentioned earlier in this thesis, the strength of the Waterloo method includes its ability to handle complex flow fields on an unstructured grid. The calculations regarding sub-grid velocity distributions and pathline integration can be implemented without the direct use of hydraulic conductivity information, assuming the cells are horizontally isotropic. In comparison with other particle tracking algorithms, the Waterloo method is better able to handle unstructured grids with complex flow fields. Here, multiple test cases for systems with moderate to significant heterogeneity are reported. The heterogeneity in hydraulic conductivity induces local velocity variations which are difficult or impossible to handle using the methods of Painter *et al*. (2012) or SSP&A (Muffles et al 2013). Flow problems with little heterogeneity tend to exhibit locally planar flow, which is readily handled by these interpolation-based tracking algorithms.

4.2.1 Zero Hydraulic Conductivity Zones

When a particular cell has horizontal and vertical hydraulic conductivities that are much lower than the conductivities of the cells around it, or even zero hydraulic conductivity (zero-K), the particles tend to go around the low-K zones. Thus, the flow field around the zero-K zones tends to be more complex compared to other areas in the model that do not have zero-K zones. Figure 4.7 represents the first example of a conceptual model with zero-K zones spread throughout the model. The determination of the zones is completely randomized such that 30% of the cells have zero hydraulic conductivity and the rest of the cells have horizontal hydraulic conductivity of 0.1 m/d and vertical conductivity of 0.001 m/d. This test case also demonstrates the capacity of the method for multiple-cell tracking. The basic input files for Example 4.4 here are provided by Visual MODFLOW-Flex as part of its tutorial example (Schlumberger, 2014). The only differences between example 4.4 and the tutorial example of the software are that this example does not contain any well or river.

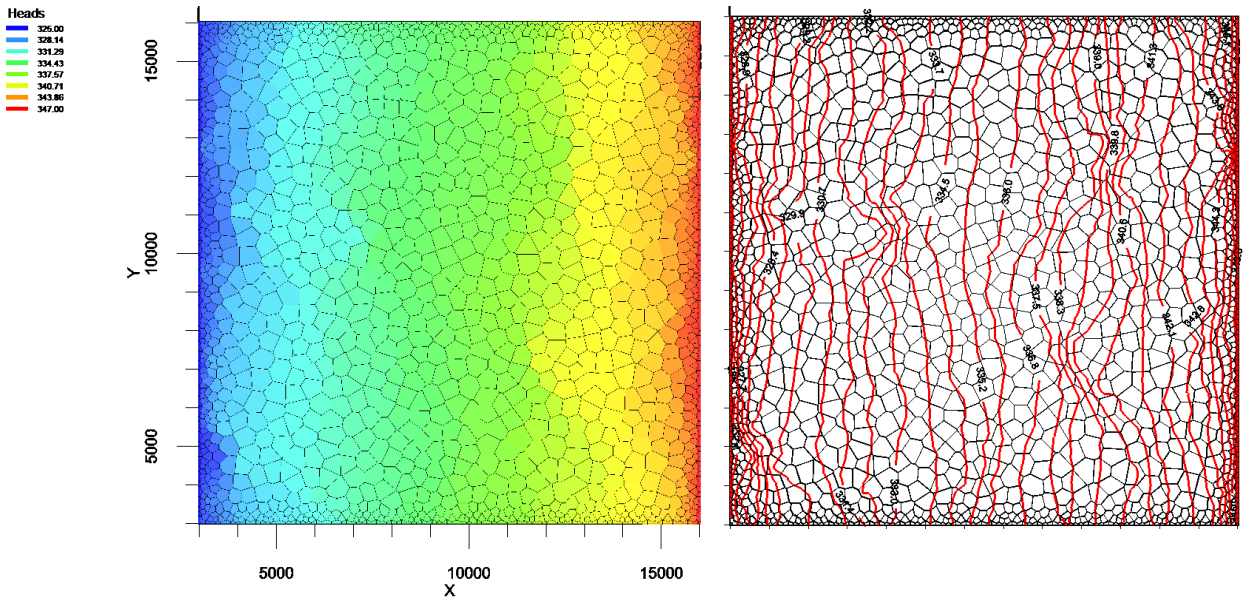


Figure 4.7: An Example of a Conceptual Model with Zero-K zones Featuring Colored Head Distribution (Left) and Head Contours (Right)

Figure 4.7 shows the conceptual model with unstructured grids that was generated using Visual MODFLOW-Flex with head contours generated using MODFLOW-USG. The left image of Figure 4.7 is

the head distribution of the model shown in color, and the right image if this figure shows the head contours of the model. The example above is a simple steady-state example with constant head on both sides of the model and no flows on top and bottom of the model; particles released on the right side of the model exit through the left side of the model. All of the basic values to regenerate example 4.4 are provided in the tutorial (Schlumberger, 2014). As shown in the figure, even with zero-K cells the heads distribution and the head contours do not clearly indicate which cell contains zero hydraulic conductivity. This implies that particle tracking method that uses head distribution may not be able to accurately create pathline integration through this type of conceptual model, which turns out to be the case. The example in Figure 4.7 consists of a two-layered model. The hydraulic conductivity zones for both layers are the same. The model is horizontally isotropic such that K_x and K_y are the same for each cell throughout the entire model. Figure 4.8 depicts the particle pathlines generated using the Waterloo method for the system configuration in Figure 4.7.

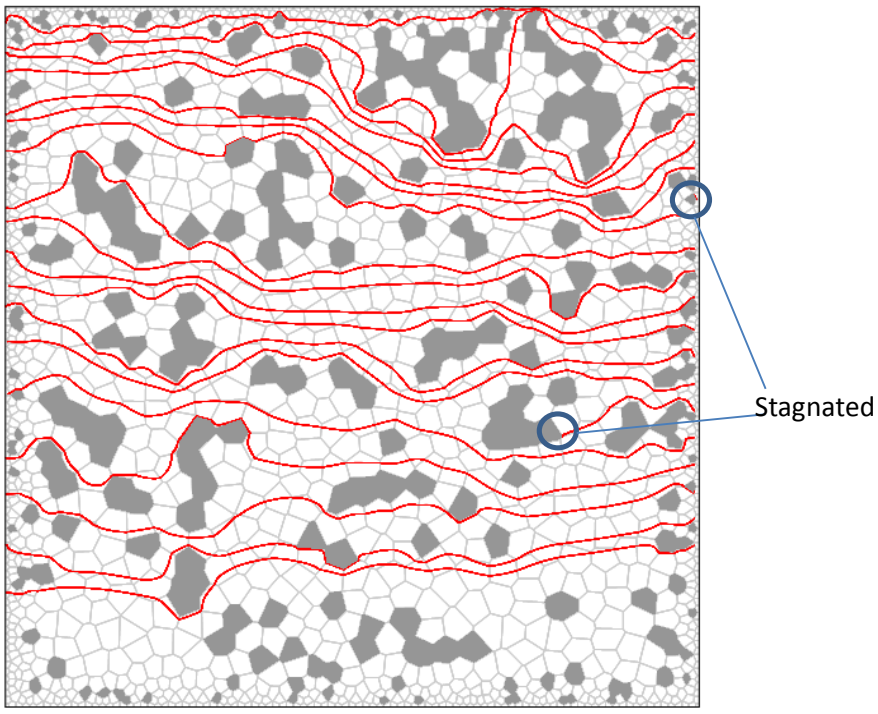


Figure 4.8: Particle Pathlines in Complex Flow Field for the Example in Figure 4.7 Generated using the Waterloo Method

The model has constant heads located on both sides of the model, where the flow goes from the right side to the left side, as shown in Figure 4.7 where the right side has constant head of 347 m and the left side has constant head of 325 m. There are a total of 3464 cells within the two-layered model with 1732 cells per layer. The domain of the model is depicted in Figure 4.7 as [x y] between [3000 and 16000] meters for both axes. In this particular case, there are 674 randomized zero-K cells with 337 zero-K cells per layer. The zones of zero hydraulic conductivity are randomized such that only 30% of the cells are zero-K cells. There are two layers in this model with 20 particles released from the right side of the model within the layer 1. However, the simulation is run using Euler scheme this time with fixed time step dt = 0.01 day. As shown in Figure 4.8, the 6^{th} and 14^{th} particles from the top do not start inside the zero-K cells, but they enter those cells and therefore reported as stagnant by the algorithm. The rest of the particles travel around the zero-K zones and exit the model through the constant head cells on the left. The reason for this accidental stagnation is because the time step chosen for the simulation is slightly too high. Therefore, by using that particular time step, the algorithm moves the particle too far using the velocity of the particle at that point. By decreasing the time step or using the adaptive time step procedure, this type of situation may be avoided. After using the adaptive time stepping for the high-order Runge-Kutta scheme in the algorithm (not shown here), these particles do not enter the zero-K cells. The purpose of example 4.4 is to show how the Waterloo method is able to handle this type of situation, and to additionally illustrate that choosing the correct pathline integration scheme and its time step is extremely crucial for the accuracy of the particle tracking simulation.

Another example regarding the particle pathlines traveling in complex flow field through a conceptual model with randomized zones of zero hydraulic conductivity is shown in Figure 4.9. This example shows the comparisons between the Waterloo method and the SSP&A method. This example 4.5 has similar basic parameters of example 4.4; the only differences are the hydraulic conductivity zones due to randomization. Each particle starting position is the same for both methods. Similar to example 4.4 in this section, the grid in example 4.5 is generated using Visual MODFLOW-Flex and the pathlines file is

created by mod-PATH3DU using the Waterloo method (Figure 4.9a) and the SSP&A method (Figure 4.9b).

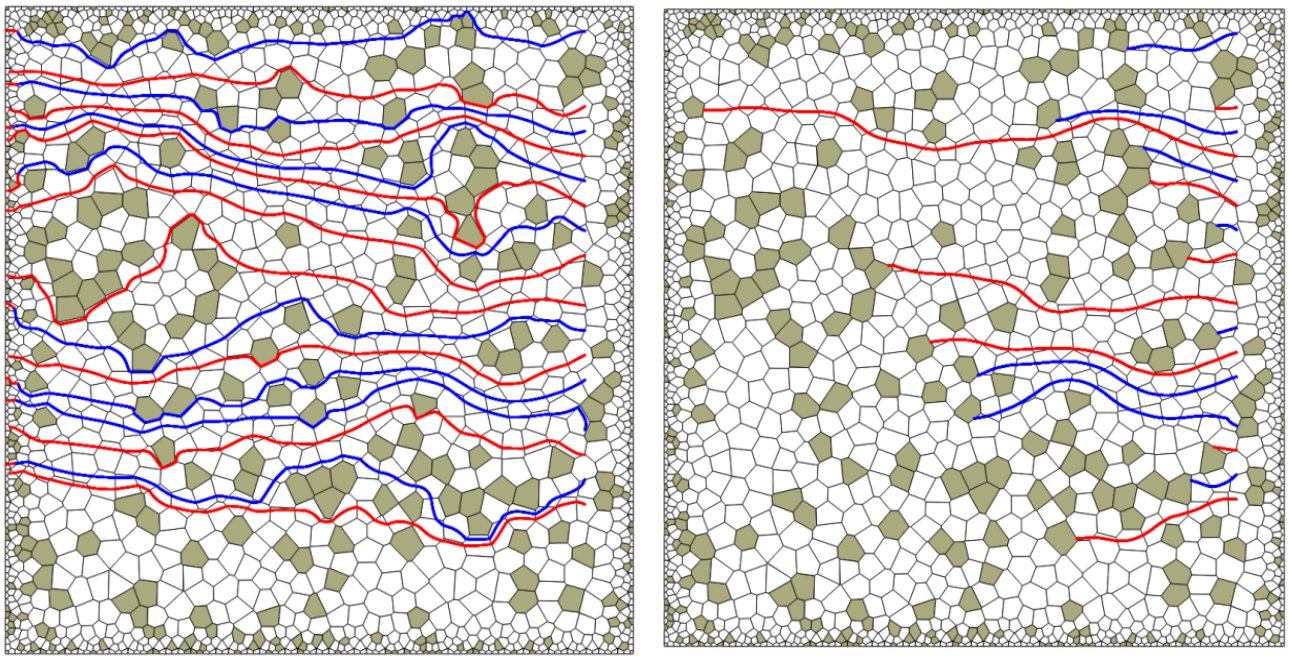


Figure 4.9: Another Example of Particle Pathlines Traveling through a Conceptual Model with Zero Hydraulic Conductivity Zones Generated by the Waterloo Method (a) and the SSP&A method (b)

There are 20 particles released at the right side of the model; 10 particles are released in the first layer and the other 10 particles are released in the second layer. There are only 16 particle pathlines that are generated in example 4.5 for both methods because the starting points of the four absent particles are inside the zero hydraulic conductivity zones, and thus immediately and validly reported as stagnant during the simulation. The red lines in Figure 4.9 represent the particle pathlines in the first layer, and the blue lines are the particle pathlines in the second layer for both methods. The cells marked in grey represent the cells with zero hydraulic conductivity. As shown in Figure 4.9 (a), the particle pathlines that are generated by the Waterloo method travel around the zones with zero hydraulic conductivity. This shows the capability of the Waterloo method to handle complex flow fields. However, the all of the particle pathlines generated by the SSP&A method erroneously enter the zero-K zones, as shown in Figure 4.9 (b). This is because the SSP&A method does not use the cell-by-cell flow to generate the

particle velocity within the cell; instead it uses head interpolation, which can lead to flows across zero-K interfaces.

The constant head zones at the sides of the model are both located in layer 1. Therefore, when the particles exit the model through the cells with constant head, they have to be in layer 1. The users must pay attention to the time step chosen when running the simulation using Euler scheme. The user must choose a very small time step in order to handle the flow field so the particles may not end up entering the cells with zero hydraulic conductivity and therefore treated as stagnant by the algorithm, as shown in example 4.4 for two of the particles. However, using such small time step may significantly increase the computational cost of the simulation. Thus, it may be better to use the adaptive time step procedure in the fourth-order Runge-Kutta scheme as implemented in example 4.5.

4.2.2 Heterogeneity

In a real subsurface system, the hydraulic conductivity of the porous media may vary from one area to another. Therefore, it is crucial for a particle tracking algorithm to be able to accurately handle heterogeneity; particularly in that it does not smooth out flow fields between cell centers, as both the SSP&A and Painter *et al.* (2012) methods do, and respects jumps in tangential flow along interfaces of changing hydraulic conductivity. As mentioned earlier in this thesis, for each time step in each cell, the hydraulic conductivities are constant and uniform. The methodology section of this thesis explains that the variation of hydraulic conductivities and hydraulic heads within each cell is combined and denoted as discharge potential. Since the every calculation that includes the discharge potential in this algorithm is done per cell-by-cell basis, the heterogeneity in the larger model does not directly affect the result of velocity distributions within a cell. However, this is not true with interpolation-based methods, which "smear" the local flow fields. Example 4.6 highlights the capability of the Waterloo method to sufficiently handle mild heterogeneity and is depicted in Figure 4.10. There are four different zones of hydraulic conductivity presented in the example; each is depicted with its own unique color. The basic

parameters of example 4.6 here use the information from Visual MODFLOW-Flex tutorial; e.g. ground, top layer, bottom layer, and boundary.

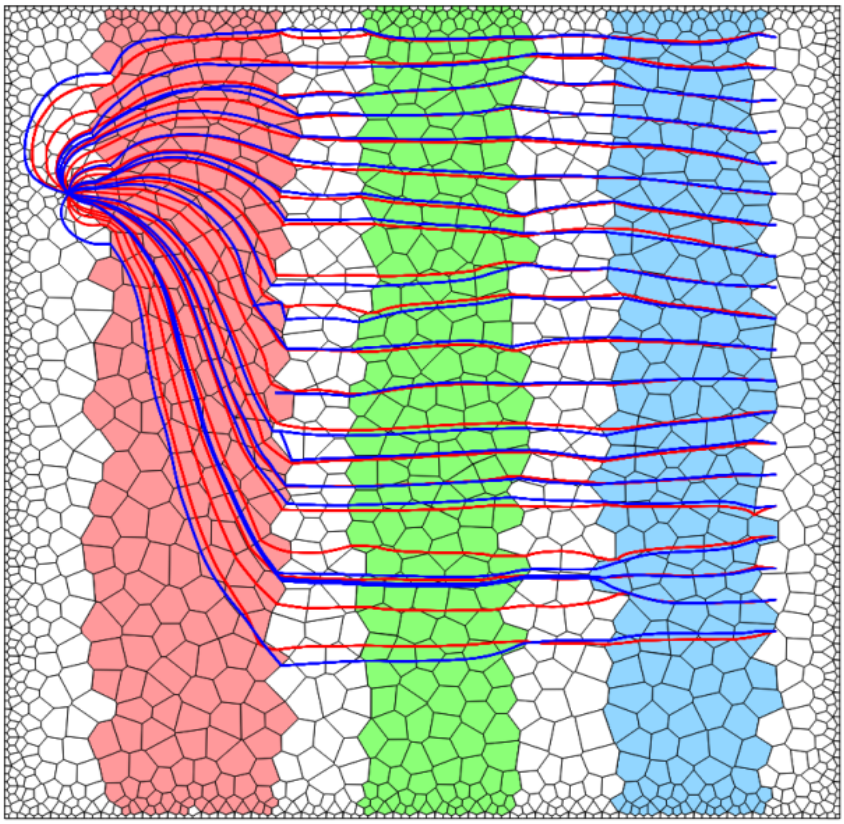


Figure 4.10: An Example of Particle Pathlines Traveling through Different Zones of Hydraulic Conductivity Generated using the Waterloo Method (Red Lines) and the SSP&A Method (Blue Lines)

Figure 4.10 shows the movement of particles through different zones of hydraulic conductivity. This particular example demonstrates the ability of the algorithm to handle heterogeneous simulations. The domain of example 4.6 here is the same as example 4.4. The background zone in the model has a hydraulic conductivity of 0.01 m/d, the blue zone is 0.05 m/d, the green zone is 0.1 m/d, and the red zone is 0.5 m/d. The model has constant head located at the right end side of the model (347 m) and the left, top, and bottom of the model are no flow boundaries. There is a pumping well located at the top left of the model at (x,y) = (4000,13000) meters. The pumping rate for this example is 20000 m^3/d. Therefore, the only location where the particles may exit is the well. There are 20 particles released from the right side of the model. Example 4.6 is simulated using both methods with fourth-order Runge-Kutta scheme with adaptive time step; the time step changes for each particle iteration. As is shown in Figure 4.10, the

particle pathlines travel through the lower hydraulic conductivity zones smoothly until it reaches the high-K zone. The red lines represent particle pathlines generated using the Waterloo method and the blue lines are pathlines generated using the SSP&A method. As shown in Figure 4.10, SSP&A particles number 7, 9, and 11 do not reach the well and get terminated erroneously in the red-K zone. The algorithm reports these particles as "terminated due to contradictory gradient". Furthermore, the terminated particle pathlines created by the SSP&A method experience "jumps" in pathlines when entering the red-K-zone. This may be due to the similar bugs that cause the "contradictory gradient" termination.

Even though the Waterloo algorithm does not directly use hydraulic conductivities in generating the particle pathlines, the cell-by-cell flows generated by MODFLOW-USG affect the velocity distributions of the particles within each cell. Other examples that highlight the capability of the Waterloo method to handle heterogeneity are presented in Figure 4.11. These examples contain more advanced features and conditions within the model such as rivers, wells, and randomized hydraulic conductivity. These examples are shown in this section to show how the pathlines integration generated by the Waterloo method are appropriately generated in the presence of other sinks such as rivers, and multiple wells within the domain. Furthermore, every cell in both of the models in Figure 4.11 has randomized horizontal and vertical hydraulic conductivity.

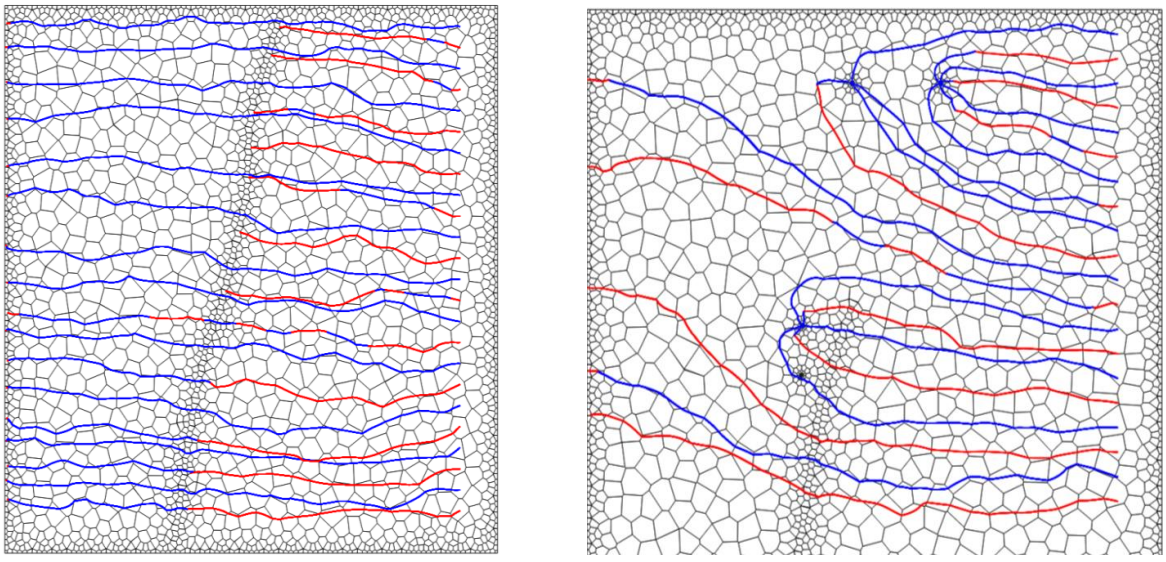


Figure 4.11: More Advanced Examples of Particle Pathlines in Heterogeneous Model with River (Left) and Wells (Right)

Each cell is horizontally isotropic with the randomized values of the horizontal hydraulic conductivities that vary between 10^{-1} and 10^{-4} m/d, and the randomized values of the vertical hydraulic conductivity that vary between 10^{-3} and 10^{-6} m/d. Due to these variations of hydraulic conductivity within each cell, the particles do not travel as smoothly as they normally would had the models been homogeneous. Due to this extreme heterogeneity within the model, there might be significant amount of vertical flows in many of the cells. Therefore, these examples show how the particles may travel to different layers due to the vertical flows, and how these "fluctuations" may affect the particle exit position. The domain size, constant head, and other basic parameters of the examples in Figure 4.11 are the same as example 4.4; these parameters are provided from Visual MODFLOW-Flex tutorial example. The left image of Figure 4.11 shows a conceptual model with a high conductance river going through the model. The right image consists of four wells and one river that has low conductance; all of the wells are pumping wells with one well at the bottom having very low pumping rate in comparison to the other wells in the image. The models have two layers. The left image contains 4094 cells and the right image contains 4138 cells. There are 20 particles released in each of the example; 10 in the first layer, and the other 10 in the second layer. The constant heads for both examples are similar; higher constant head in the first layer of the right end of the model and lower constant head in the first layer of the left end of the model. Furthermore, because of the randomized vertical hydraulic conductivity for each cell, the particles constantly move between the first and the second layer of the models.

As presented in the left image of Figure 4.11, the particles travel through the model from the right side to the left side. Since the river conductance is high, the particles that happen to travel in the first layer when they hit the river cells enter those river cells; the particles in the second layer in the area below the river cells bypass those cells and exit the model through the constant head boundary at the left end of the model. The particles that are depicted in the right image of Figure 4.11 travel through the model from the right side to the left side. The river conductance in this example is very low, and thus the particles that travel towards the river cells do not enter the cells even though some of them are in the first layer at the time. Some of the particles in the right image constantly switch layers due to the significant variation in

vertical hydraulic conductivity throughout the entire model. The bottom of each well is located in the second layer, and thus the particles would switch to the second layer before entering the wells. The particles that do not get captured by the wells of the river exit the model through the constant heads boundary located at the first layer of the left end of the model. Note that the Waterloo method knows nothing about the boundary conditions of the model and it does not have to know. These examples represent how the Waterloo method can track particles through different zones of hydraulic conductivity; especially the last two examples that show extreme cases of heterogeneity, where the horizontal and vertical hydraulic conductivities of every cell are randomized, but still keeping the model to be horizontally isotropic.

The Waterloo method can readily handle vertical anisotropy. This is due to the Dupuit-Forchheimer approximations that made the calculations for horizontal velocities to be independent of the calculations regarding the vertical velocities. Therefore, no matter how different the vertical hydraulic conductivity of one cell compared to its horizontal hydraulic conductivities, the velocity distributions of the particles, and thus the particle pathlines, are not affected. The examples presented in this section thus far show how the Waterloo method is able to handle moderately complex flow field in a conceptual model with unstructured grids. This ability is apparent when the Waterloo method is compared to the SSP&A method as shown in examples 4.5 and 4.6. This is one of the advantages of using the Waterloo method for tracking particles through arbitrary unstructured grids.

4.3 Sinks and Sources

Sinks and sources are used to define negative or positive contribution to a cell. MODPATH treats features like strong wells and rivers such that when a particle enters the cell containing these sinks, the pathlines stop at the boundary of the cell. However, the Waterloo method is able to efficiently handle sub-grid cell sinks and sources by explicitly representing the flow pattern within each cell. This proper treatment of "weak" sinks and sources (Zheng, 1994) is important because it may determine whether a particle should

be terminated (i.e. enters a river or gets pumped out of the subsurface system by a pumping well). This decision is crucial especially in advective contaminant transport to determine whether the contaminant may travel further in the groundwater system or be removed from the aquifer.

4.3.1 Strong Sinks and Weak Sinks

A strong sink is any sink strong enough such that all sides of the cell have inflows entering the particular cell; i.e. no water enters the cell through one side and discharges through another. As mentioned earlier, the Waterloo method considers the flows in through the sides, top, and bottom of the cells, generates the flow patterns, and calculates the velocities at any point within the cell. The cell, vertical, and well flows are obtainable via the cell-by-cell flow files provided by MODFLOW. Therefore, if a cell contains a strong sink, the movement of the particles inside the cell is implicitly handled by the algorithm.

The "weak" sinks and sources are those that do not result in complete inward or outward gradients along the faces of the cell which contains those sinks or sources (Zheng, 1994); i.e. water entering the cell with weak sink through one face may discharge through another face, depending on the flow pattern within the cell. Frequently encountered sinks include vertically oriented pumping wells, rivers, and constant head. An example of a cell that contains a "weak" well is presented in Figure 4.12. In this example, a simple conceptual model is discretized with an unstructured grid with constant head boundaries located at the end of both sides of the model with the flow going from left to right. The top and bottom end of the model contain no-flow boundaries. In the middle of the conceptual model, a pumping well is placed. The pumping well flow rate is chosen to create a weak sink, where some water entering the cell bypass that cell to go to the next cell. The aquifer in this example is homogeneous and isotropic. There are 13 particles released in this example within the cells near the well. The model is based upon a test case created by SSP&A to run a particle tracking simulation for the SSP&A method. Only a part of the cells around the wells is presented in the example to show more detailed pathlines integration.

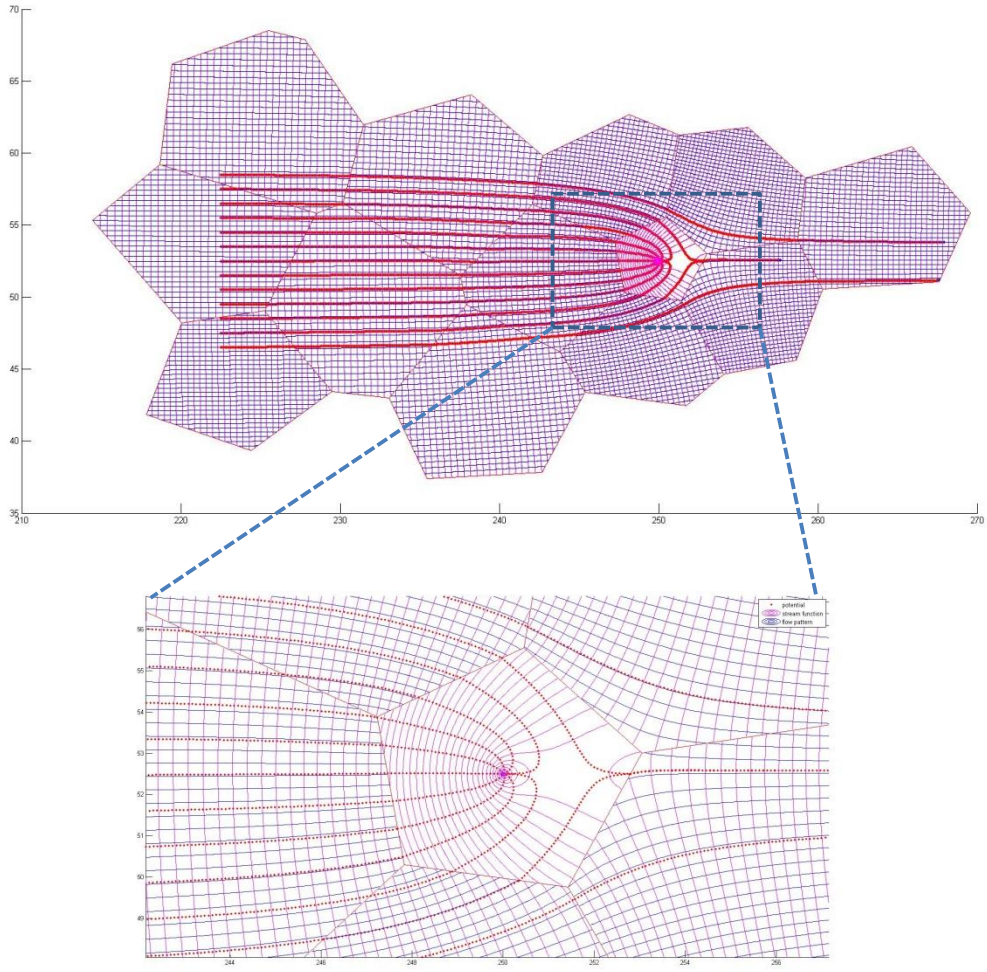


Figure 4.12: An example of a Part of a Model with a Well located in the Middle Representing a Weak Sink

As shown in Figure 4.12 above, the particles that enter the cell with a pumping well placed in it do not all get withdrawn by the well; two of the particles presented in the example above bypass the well and go to the next cell. The reconstructed head and stream function contours are shown in this figure for cells containing the pathlines. Rather than treating a cell with a sink as all strong sink, being able to determine whether a particular source is a weak sink or a strong sink may be better especially in a situation like the example in Figure 4.12, where some of the particles (or contaminant) bypass the well. Zheng (1994) introduced a procedure to handle weak sinks in a cell by using an approximate analytical solution to define the velocity distribution inside a cell that contains a weak sink or source. The reason for this is so the capture of the particles that enter the cell may be accurately determined. Pollock (1994) stated that there is no way to know whether a particle that enters a weak sink cell should discharge to the sink or pass

through the cell. However, this statement is only true if the sink distribution is unknown within the sink cell. In MODPATH, cells that contain a weak sink well may require special measures in order to generate correct particle traces and individual travel times, and hence capture zones (Abrams *et al*. 2012). However, the Waterloo method does not require any special calculation regarding this matter because of the particle sub-cell velocity created using the cell-by-cell flows.

The cases in Figure 4.11 can also be treated as examples of the capability of the Waterloo method to handle sinks and sources. The river in the left image of the figure may be considered strong sink; the particles that enter the first layer of cells along the river all get absorbed by the river. The river in the right image of Figure 4.11 is considered weak sink, where even the particles in the first layer of the cells along the river do not get absorbed. The pumping wells in the same image are considered strong sink such that all of the particles entering the second layer of a cell with a well in it enter the well. Another way to handle the problems with weak sinks in particle tracking, the users can further increase the resolution around the wells by adding more refined grids. However, by doing so, the computational cost of the simulation may increase, and even the more refined cell that contains the sink or source may still be a weak sink cell.

4.3.2 Multiple and Off-Center Wells

Most of the current particle tracking algorithms that uses MODFLOW files as its input files work with the assumption that any well that exists within a cell is located at the center coordinates of the cell. This assumption is implemented because the Well Package provided by MODFLOW does not contain any information regarding the Cartesian coordinate of the well. The files generated by MODFLOW regarding a well only provides information on the flow rates of the well, as well as in which cell the particular well is located. Furthermore, since MODPATH does not track particles inside a cell with a sink or source, it becomes irrelevant where the well is located within a cell. However, the exact location of a well inside a cell may be very important because it may determine whether the well in that cell is considered weak sink

or strong sink; by changing the location of the well inside a cell, the particle pathlines inside the cell may change significantly. A cell that contains a strong sink well when it is located at the center of the cell may become a weak sink well if it is located somewhere else within the cell. However, the Waterloo method is developed with a capability to handle wells that may not be located at the center of the cell. Currently there is no information regarding the coordinate of the well from the input files. Therefore, the algorithm currently works with the assumption that the Cartesian coordinate of any well within a cell is located at the same coordinate of the cell's center. Example 4.7 represents multiple, off-center wells that are located within a cell. The primary parameters of this cell are the same as example 3.2 in the previous section. As shown in Figure 4.13, the well is not located at the center of the cell. There are 10 particles that are released along the boundaries of the cell. The coordinate of the well is established manually, not obtained from any input file. As mentioned earlier, the equations for calculating the well flux function for the flow field within a cell and for particle velocity distributions assume that the complex well coordinate Z_{well} is the complex center coordinate of the cell $z_c = (x_c + iy_c)$.

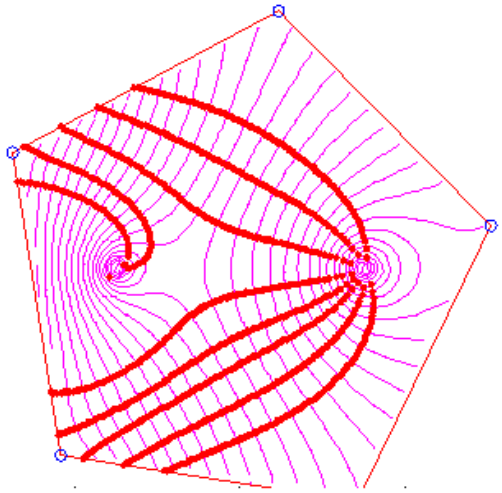


Figure 4.13: Particle Pathlines in a Cell with Multiple Wells; the Right Well has Twice the Pumping Rate of the Left Well

However, the complex well coordinate can be inserted into the equations to change the location of the well within the cell. When the input file from MODFLOW regarding the well package is modified in the future to have the coordinate of a well inside a cell, the algorithm can be slightly altered to include the

well coordinate. As presented in Figure 4.14, two of the particles enter the left well and the other particles enter the right well.

4.4 Examination of Solution Accuracy

The accuracy of the pathline integration in Waterloo method depends highly on the assumptions made and the parameters. By changing these parameters and assumptions, the total time required for the particles to exit the cell, as well as the particle exit location may change. As mentioned in the methodology section regarding the sources of error for the algorithm, by changing the order of approximation, total number of control points, and the integration algorithm fixed time or space step for each simulation, the particle positions may be altered. Details regarding the effects of the order of approximation and total number of control points to the total time travel of the particle are further discussed here. As shown in the examples thus far, the tracking error of the cell can be reduced by decreasing the time step. Furthermore, since an option to use adaptive time step is available in mod-PATH3DU software, choosing fixed time or space steps may not be desirable anymore due to high computational cost. However, the fixed time step for Euler scheme is still used here to explore the impact of fitting (or boundary) errors within a cell; this is because Euler scheme is less complicated and does not have a high computational cost for tracking particle in a single cell. The test case used for error analysis (example 4.8) is presented in Figure 4.14.

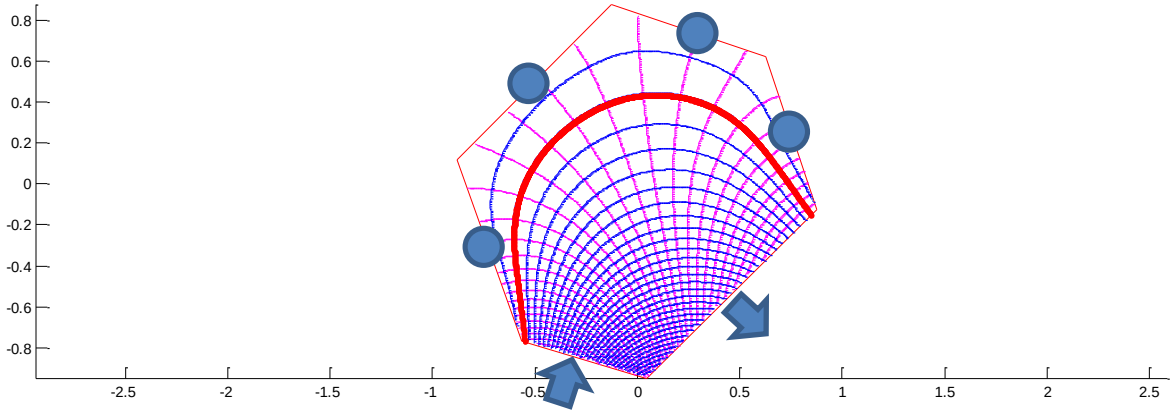


Figure 4.14: An Example of Particle Pathlines through a Hexagonal Cell with Four No Flow Boundaries

This case includes a six-sided polygon with 4 no-flow boundaries, no well, and no vertical flux term within the cell. The simulation is kept simple to better represent the sensitivity of the parameters that contribute to the source of errors in the algorithm; there is no vertical or well term in this steady-state cell. The primary and cell flow parameters of example 4.8 is presented in Table 4.7. Several simulations are run using the same cell geometry and inflows through the sides of the cell. The only parameters that change for each simulation are the total number of control points and the order of approximation of the cell. Example 4.8 uses fixed time step Euler scheme to create the pathlines.

Table 4.13: The Primary and Flow Parameters of Example 4.7

Parameter Symbols	**Example Values**
n_{faces}	6
$[x_i \quad y_i]$	$\begin{bmatrix} 0.0424 & -0.9482 \\ -0.5662 & -0.7677 \\ -0.8814 & 0.1186 \\ -0.1250 & 0.8750 \\ 0.6250 & 0.6250 \\ 0.8749 & -0.1248 \end{bmatrix}$
$[x_c \quad y_c]$	$[0 \quad 0]$
θ	0.3
Q_i	$[2.25 \quad 0 \quad 0 \quad 0 \quad 0 \quad -2.25]$

The particle starting point for each simulation is at $(x_p \quad y_p) = (-0.55 \quad -0.7725)$, chosen to be very close to one of the vertices, but not at the vertex, thus subject to the influence of Gibbs phenomenon. The fixed time step for example 4.8 is 0.001 for each simulation. The results of the simulations are shown in Figure 4.15. As shown in Figure 4.15, the total particle travel time of a particle in example 4.8 varies with the order of approximation and total control points of the cell. The smaller the values of these parameters are, the smaller the value for total particle travel time.

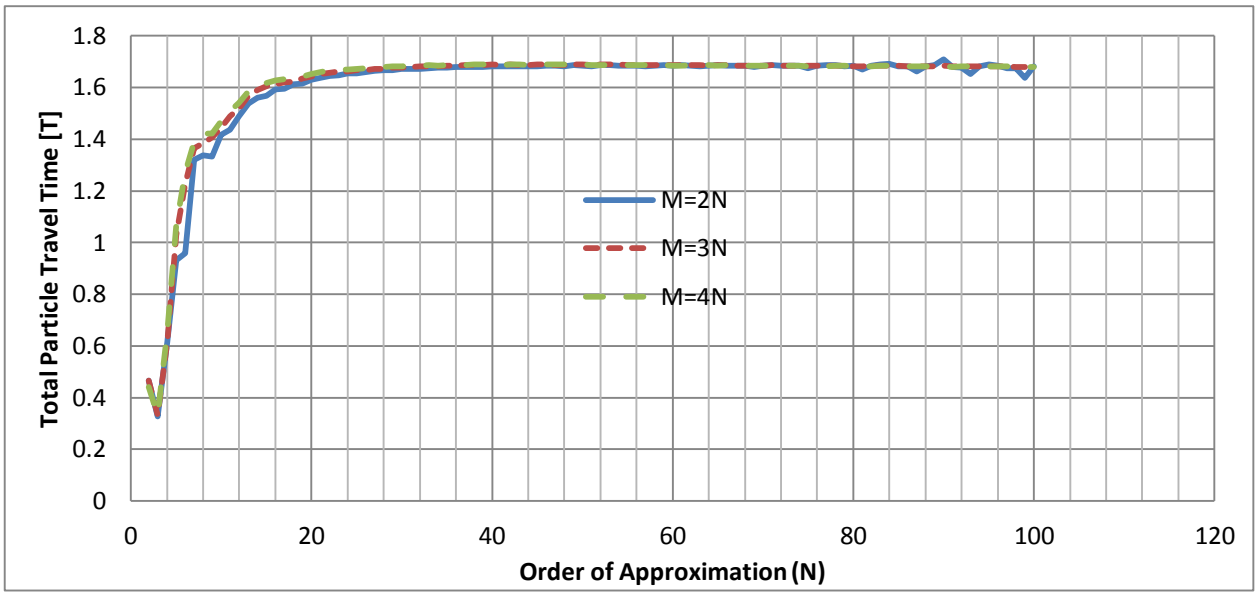


Figure 4.15: The Total Particle Travel Time with Different Order of Approximation and Total Control Points for Example 4.7

This does not mean that the particle reaches the exit point faster; it means that the horizontal velocity distribution within the cell is inaccurate and it affects the travel time of the particle to exit the cell, as well as the particle exit position. The total control points for the cell are changed accordingly with the change in order of approximation. Three types of simulation were conducted as shown in Figure 4.15: 1) M is kept as minimum to be 2N (solid blue line); 2) M is set to be 3N (dotted red line); and M is set to be high such that M=4N (dotted green line). The order of approximation parameter for all three simulations varies between 2 and 100. The reason for these simulations is not only to see how the total particle travel time gets affected by the order of approximation, but also the total control points of the cell.

As depicted in Figure 4.15, the red and green dotted lines have less fluctuation in comparison to the solid blue line. The fluctuation that exists for all three simulations might be partly due to the Gibbs phenomenon as the particle travel path is very close to the boundaries of the cell that contain no inflows. The particle total time travels for all simulations vary significantly when the order of approximation is less than 20, and they stabilize when N is greater than about 30. The red and green lines show less fluctuation regarding the particle time travel. This shows that the higher the total control points are, the more "stable" the particle time travel curves will be. The blue line in Figure 4.16 shows a more stable line

after N=30 but it shows fluctuation after N=80. This unusual fluctuation may be caused by the fluctuation in the normalized absolute average error within the cell. This phenomenon may occur using a pure collocation approach (i.e. M=2N). When the total control points parameter is set higher, as shown by the red and green lines in Figure 4.16, the fluctuation decreases but not completely gone. This is likely due to the presence of Gibbs phenomenon. The total time required for the particle to exit the point is approximately 1.68 [T]. The values for the normalized absolute average error of each simulation are depicted in Figure 4.16. The average and maximum errors are represented in this figure to show how the errors are affected by the order of approximation and total control points of the cell.

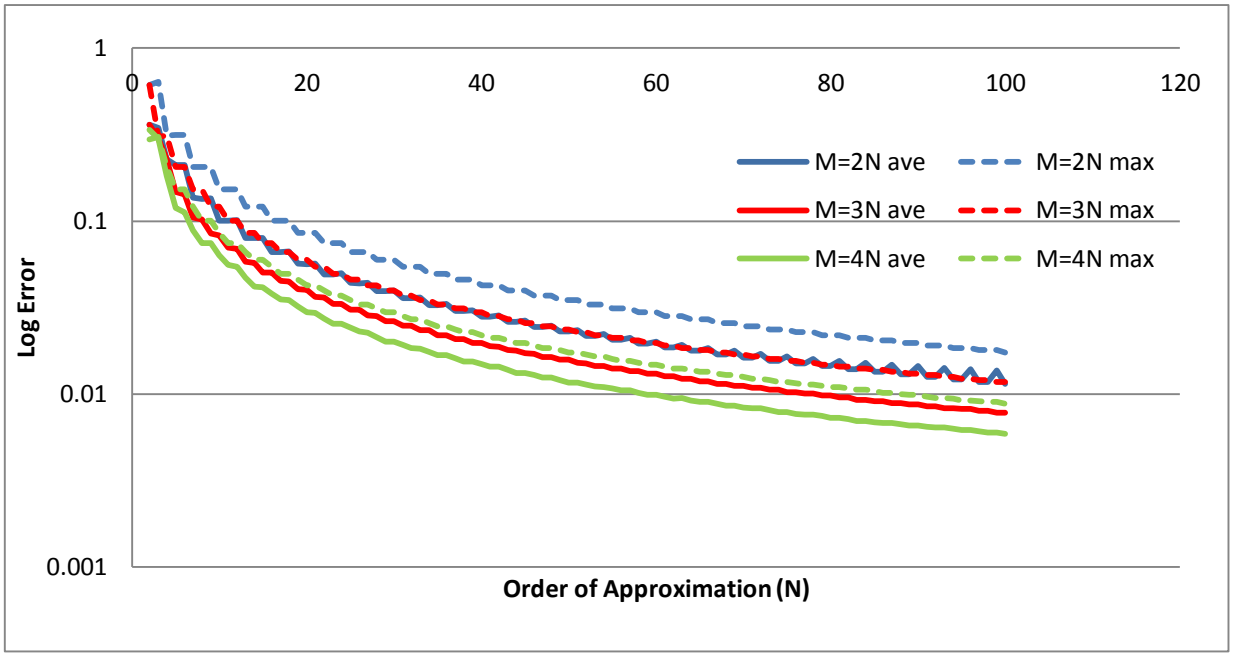


Figure 4.16: The Normalized Absolute Average and Maximum Error with Increasing N for Example 4.8

As shown in Figure 4.16, the solid lines represent the average error of the cell with increasing order of approximation and total control points. The dotted lines represent the maximum error; this maximum error represents the highest value of error for the control points along the boundary of the cell. The average and maximum error for each simulation of example 4.8 decrease with increasing order of approximation. When the total control points are further increased, the error may be significantly decreased. As shown in the figure above, the maximum error of the cell boundary for M=4N is lower for the average error for M=2N. This further shows that the order of approximation and the total control

points contribute significantly to the error of the cell, as well as the total time travel of the particle to exit the cell. As mentioned earlier in this section, there is a fluctuation regarding the total particle time travel when the order of approximation is high for M=2N. The average error for the cell when M=2N shows similar fluctuation when the order of approximation is high. By changing M and N to be extremely high the time it takes to finish the simulation increases considerably (e.g. computational time of the simulation for N=20, M=3N is about 5 seconds in unoptimized MATLAB code, and computational time of the simulation for N=100, M=3N is approximately 20 seconds). Note that each simulation is a particle tracking simulation for one particle in a single cell. Thus, overspecifying the order of approximation may result in considerable amount of computational cost for a large-scale simulation.The order of approximation is set internally within the mod-PATH3DU software to be relatively high for now; each cell contains N=40 and M=3N.

4.5 Other Test Cases

There are several test cases that are discussed in this section. The reason for simulating these test cases is to further show the capability of the Waterloo method to perform in complicated domains and to demonstrate how other tracking methods perform in these domains in comparison to the Waterloo method. Example 4.9 is taken from the quad-based grid example in paper by Pollock (2015). There are two simulations for this example: example 4.9a depicts how the Waterloo method and SSP&A method compare against the Pollock method in a rectilinear structured grid; example 4.9b represents how all three methods behave in rectilinear unstructured grid such as nested or quad-based grids. According to the paper, example 4.9a has maximum refinement of 2 x 2 in based grid cell with a total of 124 cells, and example 4.9b has maximum refinement of 8 x 8 in base grid cell and a total of 352 cells. The domain consists of 11 rows and 11 columns with uniform square cells of 200 feet on a side. No flow boundaries are specified along all four of the domain sides. Each nested child grid is 0.25 times the size of its parent's grid; e.g. if the parent grid is 200 x 200 ft, the child grid is 100 x 100 ft, and its "grandchild" grid is 50 x

50 ft. It is important to note for this comparison that the information regarding the particles starting position is not stated explicitly in the Pollock (2015) paper, and thus there is no way to know the exact particle starting position for each simulation. Therefore, the particle starting point is approximated to be as similar as possible with the figure in Pollock paper. The pathlines generated by the Waterloo and SSP&A methods are overlaid onto the image obtained from the Pollock paper. Therefore, this example is not presented to exactly show the difference in result between the three methods, rather it is just an approximation of how the three methods would behave under the same situation. Example 4.9a is shown in Figure 4.17.

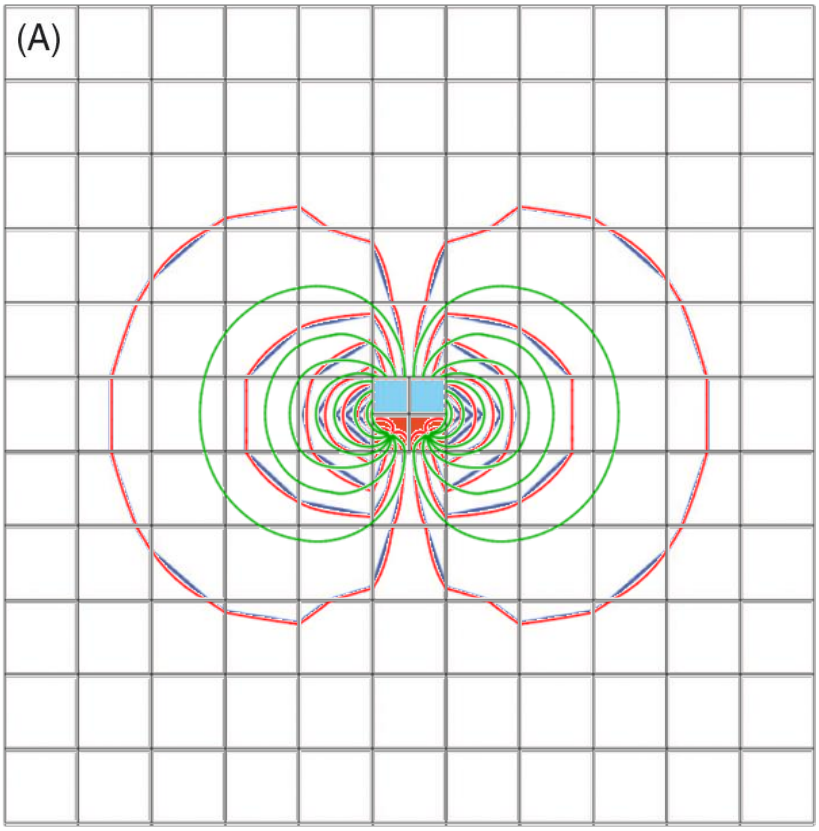


Figure 4.17: Example 4.9a, Particle Pathlines Generated in a Rectilinear Structure Grids using Three Methods: The Waterloo Method (Red Line), The Pollock Method (Blue Line), and The SSP&A Method (Green Line)

As mentioned earlier, the pathlines generated with the Waterloo and the SSP&A methods are overlain onto the image obtained from the Pollock paper (2015). As shown in Figure 4.17, the pathlines generated from the Waterloo method and the Pollock method are very similar, except in a cell that is directly connected to the two constant head cells. Although it is just an approximation, it further shows that the

results of these two methods are similar in a rectilinear structured grid. However, the SSP&A method produces pathlines that are very different than the other two pathlines. This is likely an interpolation artefact. Since the details regarding this example are not presented within the Pollock paper, some parameters, such as the particle starting locations, may be subject to approximation. The total time travel for the particles to reach the end point for the Pollock pathlines is not reported in Pollock (2015) paper. Thus, it is impossible to compare the total time travels of the particles between the three methods.

Figure 4.18 depicts the pathlines generated for example 4.9b using all three methods in a rectilinear unstructured grid; e.g. nested grids. Similar to Figure 4.17, this example contains pathlines generated using the Waterloo and the SSP&A methods to be overlaid on top of the image obtained from the Pollock paper.

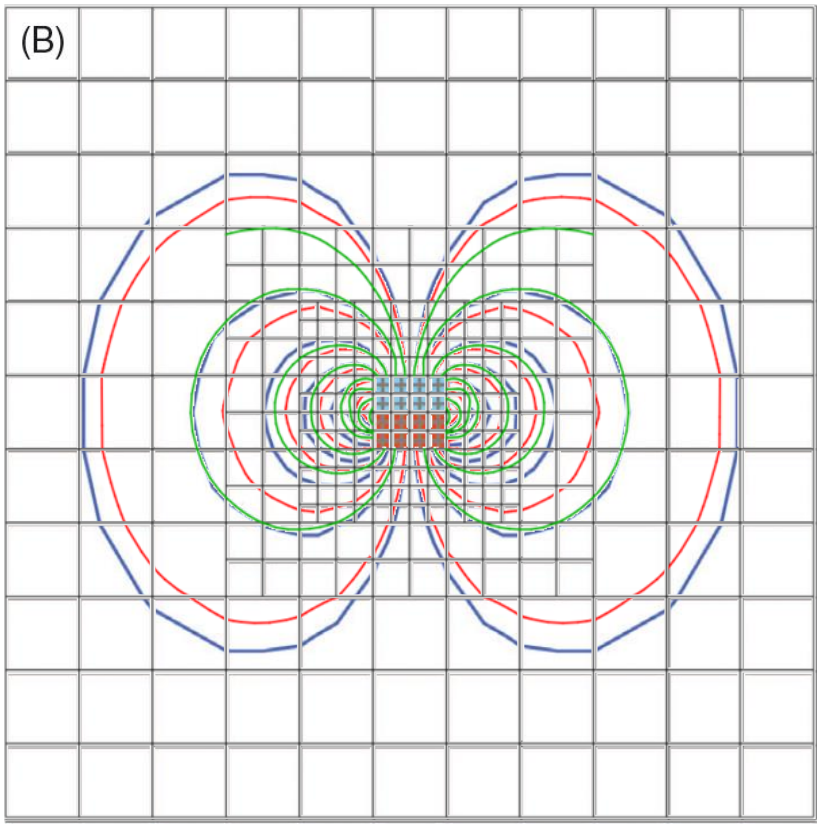


Figure 4.18: Example 4.9b, Particle Pathlines Generated in a Rectilinear Structure Grids using Three Methods: The Waterloo Method (Red Line), The Pollock Method (Blue Line), and The SSP&A Method (Green Line)

As shown in Figure 4.18, the pathlines generated using the Waterloo method is different than the Pollock pathlines. This may be due to the positioning of the particle starting location; slight difference in particle

starting position may result in significant difference in particle pathlines, particle end point location, and the travel time. However, the pathlines generated using the Waterloo and Pollock methods behave similarly in a rectilinear unstructured grid. However, the SSP&A method behaves differently than the other two methods, and two particles seem to terminate itself due to contradictory gradient, according to the output files. This again suggests that the Waterloo method can perform better than the SSP&A method.

Chapter 5

Conclusions and Recommendations

The primary challenge in tracking particles through arbitrary unstructured grids is the reconstruction of sub-grid velocity at any point within the cell. The Waterloo method is able to perform this reconstruction by solving local boundary value problems within an arbitrary polygonal cell given the side fluxes. Numerous test cases are investigated and presented in this paper in order to demonstrate the robustness of the Waterloo method. When the Waterloo method is compared to the Pollock method in rectilinear structured grids, the result shows differences in particles exit point and their time travel to be less than one percent, given a relatively large order of approximation and total control points with a small tracking time step. The robustness of the Waterloo method is shown by its ability to handle mild and extreme heterogeneity, vertical anisotropy, zero hydraulic conductivity zones, sinks and sources problems, and multiple and off-center wells. Most importantly, the Waterloo method is able to track particles through various types of unstructured grids with steady-state and transient flow. The Waterloo method is included within mod-PATH3DU software developed by the SSP&A.

There are currently two methods available to track particles through unstructured grids: the method developed by Painter *et al.* (2012); and the SSP&A method (Muffels *et al.* 2014). The method by Painter *et al.* uses unconstrained and constrained least squares methods on interior cells and boundary cells respectively in order to approximate cell-centered velocities. The SSP&A method uses local universal kriging interpolation of a MODFLOW hydraulic head solution to calculate the velocity vectors using the resultant head changes. The local interpolation-based velocity fields generated using these two methods do not respect mass balance. However, the Waterloo method constructs the velocity fields within a cell while respecting the cell-by-cell flow going in through the sides, top, and bottom of the cell while still respecting the mass balance and no-flow boundaries. The Waterloo method uses the same governing equation for three-dimensional groundwater flow as the one used in MODFLOW-USG. The sub-grid

flow field is generated by invoking the Dupuit-Forchheimer approximation in each cell, where the resistance to flow in the vertical direction is neglected. Using this assumption, the calculation regarding vertical velocity is independent of the calculation regarding the horizontal velocities of the cell. The reconstruction of sub-grid velocity distribution on a cell-by-cell basis for each MODFLOW time step requires the construction of cell geometry and the time-averaged flow rates in through the sides, top, and bottom of the cell. Information regarding the cell geometry such as vertex coordinates for each cell are provided by mesh generating software such as Groundwater Vistas or Visual MODFLOW-Flex through a grid specification file. The cell flows are provided by MODFLOW-USG. Using the information obtained from the input files, internal algorithm parameters may be populated. Using these parameters, the Taylor series coefficients can be determined. These coefficients are necessary in order to determine the flow pattern inside the cell and the complex flow velocities. The complex potential and the complex velocities are calculated using complex Taylor series. The particle pathlines integration can be computed using Euler or fourth-order Runge-Kutta scheme; the pathlines integration is necessary to determine the particle end point location and time travel between them.

There are two different types of errors within the algorithm: the boundary errors and the tracking errors. The boundary error is controlled by truncation errors, where there are not enough terms in the Taylor series to represent the flow field, and fitting errors, where discontinuities in the boundary conditions of the cell cannot be fitted properly. The parameters associated with the boundary errors include the order of approximation and total control points of the cell. These truncation errors may be reduced by increasing the value of the order of approximation and the total control points of the cell, but with the cost of higher computational time. These parameters are chosen internally to provide a relatively high accuracy in particle tracking with low computational cost. The tracking errors are caused by insufficient temporal discretization used in pathline integration. One solution to reduce tracking errors in the algorithm is to use the adaptive time step for the fourth-order Runge-Kutta scheme. The Gibbs phenomenon is another contributor to the boundary error. The appearance of significant impacts of Gibbs

phenomenon within a groundwater model appears to be rare in practice. The solution to handle the Gibbs phenomenon is not discussed in this paper; this is recommended for future work.

Three examples are provided to compare the particle pathlines generated using the Waterloo method and the Pollock method. Each example represents a unique case in rectilinear structured grids. The results for all three examples regarding the particle exit point and the total time travel show the differences to be less than one percent, i.e. conducting particle tracking using the Waterloo method produces similar results as the Pollock method in rectilinear structured grids. Five additional examples are presented to show the ability of the Waterloo method to handle complex flow field with zero-hydraulic conductivity zones, mild heterogeneity, and extreme heterogeneity. The pathlines of the particle generated using the Waterloo method are compared to the pathlines from the SSP&A method. The particle pathlines generated by the SSP&A method seem very unusual in some cases. This may be caused by internal bugs within the SSP&A algorithm and may be addressed in the future. However, it is clear that many of these issues are linked to uncorrectable issues with the head interpolation approach for velocity reconstruction. The Waterloo method can handle strong and weak sinks or sources without any additional method. Furthermore, the Waterloo method is able to track particles within a cell that contains multiple off-center wells. The algorithm can handle this cases with some slight changes if, in the future, MODFLOW is updated to be able to provide multiple and off-center wells within a cell.

The Waterloo method can readily handle vertical anisotropy. However, cases regarding lateral anisotropy are not discussed here because it is outside the scope of this thesis; one idea for the future release regarding this matter is to use coordinate transformation, such that the vertex coordinates of the cell are rearranged to fit the differences between the horizontal hydraulic conductivities, as done by Fitts (2006). Furthermore, the Waterloo method is not ready to handle unstructured grids in the vertical direction; e.g., the bottom or top face of a cell is touching more than one other cells.

References

Abrams, D., Haitjema H.M., and Kauffman L., (2012). "On Modeling Weak Sinks in MODPATH". *U.S. Geological Survey Staff – Published Research. Paper 589*

Barnes, R.J, Janković, I., (1999). "Two-dimensional flow through large numbers of circular inhomogeneities". *Journal of Hydrology, 226, pp. 204–210*

Bedient, P.B., H.S. Rifai, and C.J. Newell. (1994). "Ground Water Contamination: Transport and Remediation". *United States: Prentice Hall, Inc., Englewood Cliffs, NJ (United States).*

Bredehoeft, J. (2005). "The conceptualization model problem – surprise". *Hydrogeology Journal 13 (1), pp. 37–46.*

Clement, T.P., (1997). "RT3D (Version 1.0): A Modular Computer Code for Simulating Reactive Multispecies Transport in 3-Dimensional Groundwater Systems". *A Report for the U.S. Department of Energy.Richland, WA: Battelle Pacific Northwest Laboratory; p. 59.*

Craig, J.R., Read, W.W., (2010). "The Future of Analytical Solution Methods for Groundwater Flow and Transport Simulation", *XVIII International Conference on Water Resources*

Diersch, H.J., (2013). "FEFLOW: Finite Element Modeling of Flow, Mass and Heat Transport in Porous and Fractured Media". *Groundwater Modelling Centre*

Faust, C.R. and Mercer J.W., (1980), "Groundwater Modeling: Numerical Models", *GROUND WATER. Vol. 18, No.4.*

Fitts, C.R., (2006), "Exact Solution for Two-Dimensional Flow to a Well in an Anisotropic Domain", *Ground Water – Jan-Feb. 2006, Vol. 44, No. 1, pp 99-101*

Freeze, R.A., Witherspoon, P.A., (1966). "Theoretical Analysis of Regional Groundwater Flow: 1. Analytical and Numerical Solutions to the Mathematical Model". *Water Resources Research 2 (4), 641e656.*

Guo, W. and Langevin C.D., (2002). " User's Guide to SEAWAT: A Computer Program for Simulation of Three-Dimensional Variable-Density Ground-Water Flow". *U.S. Geological Survey, Techniques of Water-Resources Investigation 6-A7.*

Haitjema, H. M., (1995). "Analytic Element Modeling of Groundwater Flow". *Indiana University. Academic Press, Inc.*

Harbaugh, A. W. (2005). "MODFLOW-2005, The U.S. Geological Survey Modular Groundwater Model The Ground-Water Flow Process". *Technical report, U.S. Geologic Survey.*

Hazewinkel, M., ed. (2001), "Calculus", *Encyclopedia of Mathematics, Springer, ISBN.*

Hesch W., (2013), "Understanding MODFLOW-USG: A Comparison between Structured MODFLOW and UnStructured MODFLOW". *Waterloo Hydrogeologic*

Janković, I. and Barnes R.J., (1999). "High-order line elements in modeling two-dimensional groundwater flow". *J. Hydrol.*226, 211–223. *doi:10.1016/S0022-1694(99)00140-7.*

Janković, I. and Barnes, R.J., (1999). "Three-dimensional flow through large numbers of spheroidal inhomogeneities". *Journal of Hydrology, 226, pp. 224–233*

Konikow, L.F., (1992). "Ground-Water Models Cannot be Validated". *U.S. Geological Survey, Advances in Water Resources 15(1992) pp. 75-83.*

Konikow, L.F., Goode, D.J., Hornberger, G.Z., Reston, V.A., (1996). "A Three-Dimensional Method-of-Characteristics Solute–Transport Model (MOC3D)". *U.S. Geological Survey Water Resources Investigation Report 96-4267.*

Lahvis, M.A. and Baehr, A.L., (1997). "Documentation of R-UNSAT, a Computer Model for the Simulation of Reactive, Multispecies Transport in the Unsaturated Zone". *U.S. Geological Survey, Open-File Report 97-630.*

Maier, U. and Bürger, C. (2013). "An accurate method for transient particle tracking." *Water Resources Research, 10.1002/wrcr.20236, 3059-3063.*

McDonald, M.G., Harbaugh, A. W., (1984). "A Modular Three-Dimensional Finite-Different Groundwater Flow Model". *Technical report, U.S. Geologic Survey.*

McDonald, M.G., Harbaugh, A.W., (1988). "A Modular Three-dimensional Finite-Difference Ground-Water Flow Model". *Open-File Report 83e875. U.S. Geological Survey. Book 6.*

McDonald, M.G. and Harbaugh, A.W., (2003), "The History of MODFLOW", *GROUND WATER Vol 41 No. 2*

Muffels, C., Wang, X., Tonkin, M., Neville, C., (2014). "User's Guide for mod-PATH3DU: A Groundwater Path and Travel-Time Simulator". *S.S. Papadopulos & Associates, Inc.*

National Groundwater & Contaminated Land Centre (NGCLC). (2001). "Guide to good practice for the development of conceptual models and the selection and application of mathematical models of contaminant transport processes in the subsurface". *NGCLC report NC/99/38/2.*

Painter, S.L., Gable, C.W., Kelkar, S. (2012). "Pathline Tracking on Fully Unstructured Control-Volume Grids". *Computational Earth Sciences Group DOI 10.1007/s10596-012-9307-1. Springer Science +Business Media B.V. 2012*

Panday, S., Langevin, C.D., Niswonger, R.G., Ibaraki, M., Hughes, J.D., (2013). "MODFLOW-USG Version 1: An Unstructured Grid Version of MODFLOW for Simulating Groundwater Flow and Tightly Coupled Processes Using a Control Volume Finite-Difference Formulation". *U.S. Geological Survey, Techniques and Methods 6-A45*

Pollock, D.W., (1988). "Semianalytical computation of path lines for finite-difference models". *Ground Water 26 (6), 743–750.*

Pollock, D.W., (1989), "Documentation of computer programs to compute and display pathlines using results from the U.S. Geological Survey modular three-dimensional finite difference groundwater flow model". *U.S. Geological Survey Open-File Report 89–381, 188 p.*

Pollock, D.W., (1994). “Users guide for MODPATH/MODPATH-PLOT, version 3: A particle tracking post-processing package for MODFLOW, the U. S. Geological Survey finite-difference ground-water flow model”. *Open-file report 94-464, U.S. Geological Survey.*

Pollock, D.W., (2012). “User Guide for MODPATH Version 6 – A Particle-Tracking Model for MODFLOW”. *U.S. Geological Survey, Techniques and Methods 6-A41*

Pollock, D.W., (2015). “Extending the MODPATH Algorithm to Rectangular Unstructured Grids”. *U.S. Geological Survey*

Raeen, K., (2008), “A Study of The Gibbs Phenomenon in Fourier Series and Wavelets”, *University of New Mexico*

Rushton, K. and Redshaw, S. (1979). “Seepage and groudwater flow: numerical analysis by analog and digital methods”. *Wiley, New York, 1979. No. of pages: 339*

Schlumberger Water Services, (2014), “Visual MODFLOW Flex User’s Manual”.

Strack, O.D.L., (1984). “Three-Dimensional Streamlines in Dupuit-Forchheimer Models”. *Water Resources Research, Vol. 20, No. 7, pp. 812-822.*

Strack, O.D.L., (1985). “An application of determining streamlines in a Dupuit-Forchheimer model”. *Hydrological Science and Technology Journal 1, no. 1: 17–23.*

Strack, O.D.L., (1989). “Groundwater Mechanics”. *Englewood Cliffs, New Jersey: Prentice Hall. Out of print: Contact Strack Consulting Inc.*

Suk, H. and Yeh, G. (2009). ”Multidimensional Finite-Element Particle Tracking Method for Solving Complex Transient Flow Problems.” *J. Hydrol. Eng., 14(7), 759–766.*

Suk, H. and Yeh, G. (2010). "Development of particle tracking algorithms for various types of finite elements in multi-dimensions." *Computers & Geosciences, 10.1016/j.cageo.2009.09.011, 564-568.*

Suk, H. (2012). "Practical Implementation of New Particle Tracking Method to the Real Field of Groundwater Flow and Transport." *Environmental Engineering Science, 10.1089/ees.2011.0153, 70-78.*

Sun, R.J. and Johnson, R.H., (1994). “Regional Aquifer System Analysis Program of the U.S. Geological Survey”, *1978e1992. U.S. Geological Survey. Circular 1099*

Therrien, R., McLaren, R.G., Sudicky, E.A., Panday, S.M., (2010). “HydroGeoSphere: A Three-Dimensional Numerical Model Describing Fully-Integrated Subsurface and Surface Flow and Solute Transport”. *Groundwater Simulations Group*

Toth, J., 1963. “A theoretical analysis of groundwater flow in small drainage basins”. *Journal of Geophysical Research 68 (16), 4795e4812.*

United States Geological Survey (USGS), (2011), “Status of MODFLOW Versions and MODFLOW-Related Programs Available on USGS Web Pages”, *Office of Groundwater, U.S. Geological Survey*

Voss, C.I. and Provost, A.M., (2008). "SUTRA, a model for saturated-unsaturated variable density ground-water flow with solute or energy transport". *U.S. Geological Survey, Water-Resources Investigations Report 02-4231, Virginia, 270 pp.*

Wels, C., Mackie, D., Scibek, J., (2012), "Guidelines for Groundwater Modelling to Assess Impacts of Proposed Natural Resource Development Activities", *British Columbia Ministry of Environment, Water Protection & Sustainability Branch. Robertson GeoConsultants Inc. & SRK Consulting (Canada) Inc. Report No. 194001*

Yang X., Aber, J.T., Steward D.R., (2009). "Using Conceptual Groundwater Data Model to Model Groundwater Flow with PMWIN", *Kansas State University, Department of Civil Engineering.*

Yeh, G.T., Sharp-Hansen, S., Lester, B., Strobl, R., Scarbrough, J., (1992). "3DFEMWATER /3DLEWASTE: Numerical Codes for Delineating Wellhead Protection Areas in Agricultural Regions Based on the Assimilative Capacity Criterion". *Washington, DC: U.S. Environmental Protection Agency.*

Zheng, C., (1989), "PATH3D, A ground-water path and travel-time simulator, version 3.0 user's manual", *S.S. Papadopulos & Associates, Inc., Bethesda, MD.*

Zheng, C., Bethesda, M.D. (1990). "MT3D User's Manual: A Modular Three-Dimensional Transport Model for Simulation of Advection, Dispersion, and Chemical Reactions of Contaminants in Groundwater Systems". *S.S. Papadopulos and Associates*

Zheng, C., (1992), "PATH3D, A ground-water path and travel-time simulator, version 3.2 user's manual", *S.S. Papadopulos & Associates, Inc., Bethesda, MD.*

Zheng, C., (1994), "Analysis of particle tracking errors associated with spatial discretization", *Ground Water, vol. 32 no. 5: 821-828.*

Zheng, C. and Wang, P.P., (1999). "A modular three-dimensional multispecies transport model for simulation of advection, dispersion and chemical reactions of contaminants in groundwater systems", *MT3DMS User's Guide, The University of Alabama, Prepared for: U.S. Army Corps of Engineers.*

Zheng, C., and Bennett, G., (2002). "Applied Contaminant Transport Modeling", *2nd ed., John Wiley & Sons, New York.*

Zhou, Y. and Li, W. (2011). "A Review of Regional Groundwater Flow Modeling". *Geoscience Frontiers, 2(2), pp.205-214, April 2011.*