Non-Darcy Flow Through Porous Media

by

James Edward Bené, B.S.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

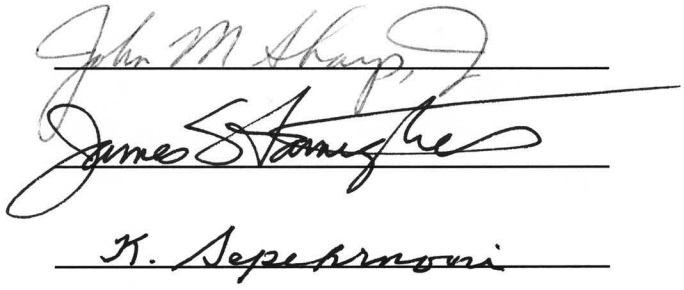
Master of Science in Geological Sciences

The University of Texas at Austin

May 2000

Non-Darcy Flow Through Porous Media

Approved by
Supervising Committee:



Dedication

Dedicated to the memory of my father, to my wife, Amy and all my family and friends who new me better than I knew myself.

Acknowledgements

I acknowledge the guidance of Dr. Jack Sharp who supervised this study. His assistance and support were invaluable to me.

Above all, I thank my wife, Amy for her encouragement. Her love and understanding have helped me to accomplish this goal. My achievements are a tribute to her sacrifices.

I also thank my mother, brother and my wife's family for all their support along my studies.

March 1, 2000

Abstract

Non-Darcy Flow Through Porous Media

James Edward Bené, M.S.Geo.Sci.

The University of Texas at Austin, 2000

Supervisor: John M. Sharp

Since its introduction, Darcy's law has been implemented as a mathematical tool that allows simple calculation and prediction of low velocity subsurface flows. However, turbulence, non-isothermal conditions, as well as other factors can create conditions where Darcy's law does not accurately describe the head and velocity distributions within a given porous matrix. Darcy's law has been widely applied to analytical and numerical modeling of fluid flow through porous matrix, regardless of the hydrogeologic setting. This study attempts to quantify the error incurred by these models through simultaneous numerical modeling of the mass continuity equation using Darcy's law as well as Forchheimer's relation. To this end, results from steady-state and transient Darcy-based and Forchheimer-based numerical models are presented in this study.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

In 1856, Henry Darcy made an extremely important contribution to the science of fluid flow through a porous media. In his report (Darcy, 1856) on the public fountains of Dijon, France stated that: "for identical sands, one can assume that the discharge is directly proportional to the head and inversely proportional to the thickness traversed." He made this now famous assertion following a series of experiments in which Darcy monitored the flow rate through a sand filled cylinder. His experimental apparatus consisted of a cylinder 2.5 m in height and 0.35 m in diameter packed with sand, which was sealed and fitted with manometers and at each end.

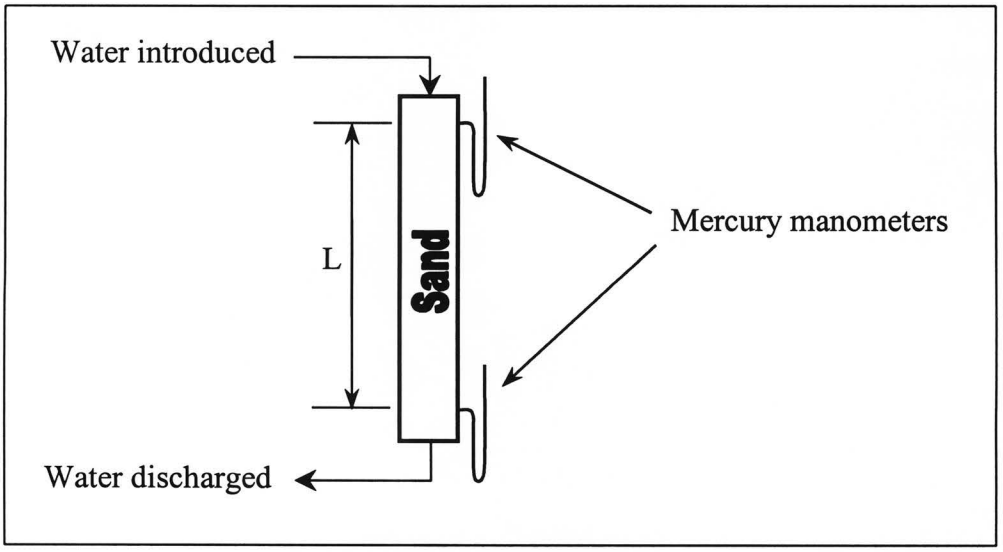


Figure 1.1: Simplified illustration of Darcy's experimental apparatus.

Fittings for the introduction and removal of water were also installed at the ends of the cylinder. A simplified drawing of the apparatus is shown in Figure 1.1. Water was introduced at the top of the column, while discharge drains from bottom. During his experiments, Darcy and his colleagues monitored the head at the ends of the cylinder and the rate of discharge from the bottom of his apparatus.

Using the measurements from these experiments, Darcy formulated the flow relationship known today as Darcy's law. Stated mathematically, Darcy's law is:

$$Q = -K \frac{dh}{dL} A \qquad (1\text{-}1)$$

Darcy's law

Where dh/dL is the hydraulic gradient over some distance L. [-]
Q is the volumetric rate of flow. [L^3/t]
K is the hydraulic conductivity. [L/t]
A is the cross-sectional area of flow. [L^2]

Since it's introduction, Darcy's law has been implemented as a mathematical tool that allows simple calculation and prediction of low velocity subsurface flows. Simply stated, Darcy's law is the assumption of a linear relationship between specific discharge (Q/A) and hydraulic gradient (dh/dL). In equation 1-1, the constant of proportionality (K) is known as

hydraulic conductivity, and was found to vary with the type of porous media being studied.

Darcy's law is acknowledged to be valid for most subsurface flows under natural conditions. However, there are numerous circumstances where Darcy's law does not accurately describe the head and velocity distributions within a given porous matrix. Turbulent, non-isothermal, compressible flows that do not behave linearly, or flows composed of non-Newtonian or chemically reactive fluids may not obey Darcy's law (Lage, 1998).

In this study, flows that do not conform to the linear relationship expressed in equation 1-1 will be termed non-Darcy or nonlinear flows.

Non-Darcy flow behavior generally lowers the volume of water that can be transmitted through a given matrix for a given hydraulic gradient. In most cases, this effect is small, although nonlinear flow behavior can be significantly increased in karstic, fractured or high permeability clastic aquifers under pumped or natural conditions. The non-Darcy effects are, for the most part, ignored by hydrogeologists, with the exception that, in the field, the reduced discharge (relative to values computed with Darcy's law) in most wells with high pumping rates is noted and recorded. Assuming that the reduction in discharge is probably due to turbulence in and around the well-bore, most hydrogeologists employ the general term "well loss", ignoring the

possibility that information on deviations from Darcy's law can provide valuable insights into the true nature of subsurface flow.

At this point, the term "well loss" must be defined and distinguished from the error in head calculation investigated in this work. Prior to defining well loss, total drawdown at the well must be specified in the traditional method. Mathematically this is given by equation 1-2.

$$s_{well} = BQ + CQ^n \tag{1-2}$$

Where s_{well} is the drawdown at the well screen [L]
Q is the well discharge [L^3/t]
B is the coefficient of drawdown due to aquifer effects [t/L^2]
C is the coefficient of drawdown caused by flow through the well screen and flow inside the well bore. [units variable depending on the value of the exponent "n"]

The term CQ^n is labeled "well loss", and is the amount of drawdown observed during well pumping that is not attributable to that predicted by Darcy's law (Todd, 1959). The exponent "n" has been assigned various values by different authors. In 1947, Jacob found that the assumption that n=2 was reasonable, while others (Rorabaugh, 1953) argued that the value of n can vary significantly.

This work does not attempt to quantify well loss as described above. No attempt was made to examine the nonlinear flow effects in the well bore

and screen. Instead, this study attempts to quantify the error in calculated drawdown when Darcy's law is assumed valid for the porous matrix itself, regardless of well bore mechanics.

It is important to have an understanding of the potential error contracted when applying a mathematical model to a particular physical situation. However, most hydrogeologists use numerical models that employ Darcy's law with little thought as to their accuracy in a given hydrogeologic setting.

This study attempts to quantify this error through simultaneous numerical modeling of the mass continuity equation using Darcy's law as well as a quadratic flow equation popularly, but erroneously labeled the Forchheimer relation. The true author of equation 1-2b is Arsene Dupuit, who modified Prony's equation describing open channel flow (Lage, 1998). Forchheimer's contribution was to publish a defense of the validity of this quadratic function to describe the pressure drop for flows in porous media (Forchheimer, 1901). However, the relationship shown in equation 1-2b was mistakenly attributed to Forchheimer (Lage, 1998), possibly because the distribution of scientific information in Europe at the end of the 19^{th} century was not very efficient. This having been said, equation 1-2b is referred to as Forchheimer's relation in this work, for the reason that that label is common throughout scientific literature and is less likely to cause confusion.

$$-dh/dL = aq \qquad (1\text{-}2a)$$

Darcy's law

$$-dh/dL = bq^2 + aq \qquad (1\text{-}2b)$$

Forchheimer's Relation

Where dh/dL is the hydraulic gradient over some distance L. [-]
q is the specific discharge or Darcian velocity (Q/A). [L/t]
a is the hydraulic resistivity (Irmay 1958) [t/L],and is equal to 1/K (always positive), where K is the hydraulic conductivity. [L/t]
b is the nonlinear resistivity coefficient [t^2L^{-2}] (always positive)

For this work, the "a" coefficient in equations 1-2a and 1-2b is defined as the inverse of hydraulic conductivity (K), and is labeled the hydraulic resistivity of the matrix (Irmay, 1958).

Upon inspection, it is obvious that the Forchheimer relationship between specific discharge and hydraulic gradient is Darcy's law with the introduction of a constant multiplied by the square of the specific discharge. This term accounts for the flow behavior resulting from the deviation from Darcian linearity. Until fairly recently, there was no analytical explanation of why the squared term could accurately predict non-Darcy pressure drop. Proof was supplied in the late 1950's when Irmay showed that the Forchheimer relation can be derived directly from the Navier-Stokes (momentum/continuity) equations. For the details of this derivation see: (Irmay, 1958).

In the past, a number of workers have studied the inadequacies of Darcy's equation, and attempted to find methods of correction. Like this work, many have focused on the use of Forchheimer's equation as a substitute for Darcy's law in analytical and numerical modeling. However, there has been a sizable amount of research into the applicability of alternate mathematical flow models. A thorough review of the earlier work, which concentrated on the different aspects of flow through porous media, is provided by Subramanya and Madhav (1978).

Since the time of Subramanya and Madhav's report several other hydrogeologists and civil engineers have researched and implemented techniques that have advanced the study of non-Darcy flow. Analytical solutions to non-Darcy flow to a well in a fractured and unfractured matrix were presented by Sen (1986, 1987, and 1988). A finite-element model that predicts the distribution of head seen in unconfined, non-Darcy flow to a well was introduced by Rao and Das (1978), while a study of the non-Darcy effects on the computation of seawater intrusion lengths was presented by Basak and Rajagopalan (1982). One of the most recent contributions was provided by Venkataraman and Rao (1998). In this paper, the authors reanalyze previous works pertaining to the calculation of non-linear resistivity coefficients and propose some empirical equations relating Reynolds numbers and porous media parameters.

It is instructional to note that petroleum engineers have also contributed a substantial body of research on non-Darcy flow, especially pertaining to the flow of natural gas through porous media. Because of the compressibility of natural gas, it is often inappropriate to apply Darcy's law to flows that incorporate a large gaseous component. Accurate prediction of natural gas well yields is of substantial economic concern, so petroleum engineers were and are motivated to find an alternate to (or correction for) Darcy's law. One of the most interesting contributions was presented by Lee, Logan and Tek (1987). Here, a one-dimensional (radial) analytical solution and numerical model describing transient flow of natural gas through porous media was introduced.

The remainder of this study describes the application of Darcy's law and Forchheimer's relation to two-dimensional, finite-difference numerical models.

In Chapter 2, governing flow equations are derived using the basic principle of continuity. Equations 1-2a and 1-2b are modified and incorporated into the flow equations, and some preliminary observations are discussed.

Chapter 3 describes the methods used to convert the continuous flow equations described in Chapter 2 into finite-difference form. Difference analogs for the needed derivatives are derived; explanations of their application to the numerical models are given; and a discussion of the techniques used to solve the matrix of equations is included.

Chapter 4 describes the model calibration to analytically derived solution surfaces. Difficulties encountered in the accurate calculation of the hydraulic conductivity at the well node, as well as the author's method of correction is discussed. The results and conclusions of the calibration simulations are also presented in Chapter 4.

The results of steady-state and transient two-dimensional, heterogeneous models are reported in Chapter 5. Six example model configurations incorporating varying distributions of conductivity heterogeneities are investigated. Head difference, velocity and Reynolds number distributions for each example are plotted and discussed.

Finally, in Chapter 6 a discussion of the results of all simulations and the conclusions derived from them is presented. Offered in this chapter are descriptions of the perceived shortcomings of the method of application of the Forchheimer-based equations, as well as recommendations for further study.

Chapter 2. Fundamental equations

In this chapter, mass-balance (continuity) principles are used to derive the fundamental flow equations that are applied to the model.

2.1 Continuity Analysis

Consider a representative elemental volume of porous matrix with dimensions $\Delta x, \Delta y, \Delta z$ shown below in Figure 2.1.

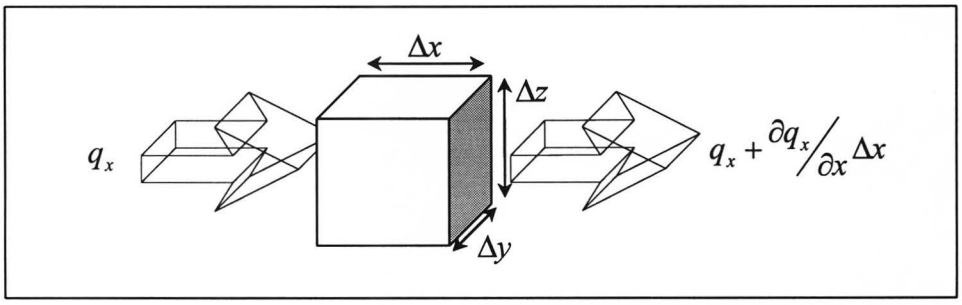


Figure 2.1: Representative volume of porous matrix

The volume of flow into the cell through the left face is the average velocity of the flow in the x direction multiplied by the area of the face perpendicular to that flow $[q_x(\Delta y \Delta z)]$, and the volume of flow out of the cell through the right face is the area of the exit face multiplied by average velocity of the flow in the x direction plus the change in that flow multiplied by the length of the cell: $\left[\left(q_x + {}^{\partial q_x}/_{\partial x}\Delta x\right)(\Delta y \Delta z)\right]$. The difference between the volume flux into

and out of the cell is the volume of fluid gained or released from storage within the cell:

$$q_x(\Delta y\Delta z)-\left[q_x(\Delta y\Delta z)+{}^{\partial q_x}\!/_{\partial x}\,\Delta x(\Delta y\Delta z)\right]=-{}^{\partial q_x}\!/_{\partial x}\,\Delta x(\Delta y\Delta z)=\Delta_x Storage \tag{2-1a}$$

Where $\Delta_x Storage$ is the volume of fluid gained or released from the cell due to flow along the x-axis. The same process can be repeated to find the volume of fluid gained or released from storage within the cell due to flow along the y-axis and z-axis:

$$q_y(\Delta x\Delta z)-\left[q_y(\Delta x\Delta z)+{}^{\partial q_y}\!/_{\partial y}\,\Delta y(\Delta x\Delta z)\right]=-{}^{\partial q_y}\!/_{\partial y}\,\Delta y(\Delta x\Delta z)=\Delta_y Storage \tag{2-1b}$$

$$q_z(\Delta x\Delta y)-\left[q_z(\Delta x\Delta y)+{}^{\partial q_z}\!/_{\partial z}\,\Delta y(\Delta x\Delta y)\right]=-{}^{\partial q_z}\!/_{\partial z}\,\Delta y(\Delta x\Delta y)=\Delta_z Storage \tag{2-1c}$$

Combining equations 2.1a-c results in equation 2-2a:

$$-{}^{\partial q_x}\!/_{\partial x}\,\Delta x(\Delta y\Delta z)-{}^{\partial q_y}\!/_{\partial y}\,\Delta y(\Delta x\Delta z)-{}^{\partial q_z}\!/_{\partial z}\,\Delta z(\Delta x\Delta y)=\Delta x\Delta y\Delta z S_s\,{}^{\partial h}\!/_{\partial t}$$
$$=\Delta Storage \tag{2-2a}$$

Equation 2-2a reduces to equation 2-2b.

$$-\frac{\partial q_x}{\partial x}-\frac{\partial q_y}{\partial y}-\frac{\partial q_z}{\partial z}=S_s\frac{\partial h}{\partial t} \tag{2-2b}$$

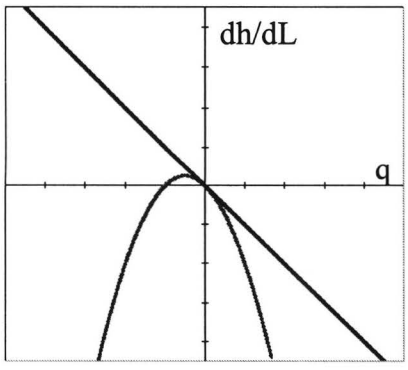
where S_s is the specific storage $(\rho g(\alpha+\beta\phi))$ [L^{-1}]

ρ is the density of the fluid [ML^{-3}]

g is the acceleration due to gravity [Lt^{-2}]
α is the compressibility of the matrix [ML^3t^{-2}]
β is the compressibility of the fluid [ML^3t^{-2}]
ϕ is the porosity [-]
q is the specific discharge [Lt^{-1}]
h is the hydraulic head (elevation + pressure/ ρg) [L]
t is time [t]

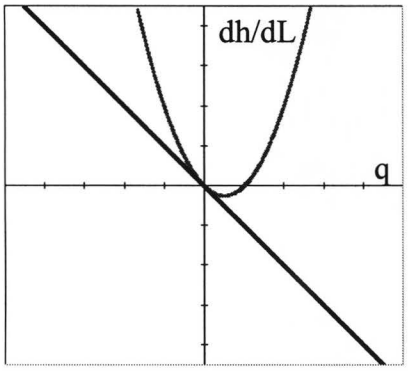
2.2 Analysis of Forchheimer's Relation

Modeling of non-Darcy flow characteristics requires that the standard form of Forchheimer's relation be transformed into a form that can be substituted into the flow equation (equation 2-2b). In order to do this, Forchheimer's relation must be solved for specific discharge (q). Prior to this, we note that the original form of the equation can be a little misleading. On close inspection, it can be seen that for the relation to make sense in Cartesian space it is composed of two equations. This is illustrated in Figures 2.2a and 2.2b:



$$-\frac{dh}{dL} = bq^2 + aq \qquad (2\text{-}3a)$$

Figure 2.2a: Forchheimer equation for positive discharge values. Parabolic curve is a plot of equation 2-3a, diagonal line is a plot of Darcy's law.



$$-\frac{dh}{dL} = -bq^2 + aq \qquad \text{(2-3b)}$$

Figure 2.2b: Forchheimer equation for negative discharge values. Parabolic curve is a plot of equation 2-3b, diagonal line is a plot of Darcy's law.

In Figures 2.2a and 2.2b, the parabolic curves represent the separate Forchheimer relations while the diagonal lines represent values calculated using Darcy's law. In both cases the "a" and "b" coefficients are set to unity, and the reader is reminded that, as defined above, these coefficients are always positive. After inspection, it is apparent that the standard form of the Forchheimer relation (eqn. 2-3a) is valid only for specific discharges (q) greater than zero. This is because specific discharge and hydraulic gradient must always be of opposite signs, and it is clear that this condition is not satisfied for negative values of specific discharge in Figure 2.2a. This may seem strange, but recall that Forchheimer's relation was formulated specifically for calculation of a *head drop* with increasing, *positive* fluid velocity. In other words, the equation was written for one-dimensional fluid flow, and never intended for use in the calculation of a *head gain* over a distance. In order for this formula to be useful in numerical modeling, a

second form of Forchheimer's relation is needed to account for negative (Cartesian) fluid velocities, which result in positive (Cartesian) hydraulic gradients. This is accomplished by assuming a positive hydraulic gradient and changing the sign of the squared specific discharge term on the right side of the equation. This results in a form of Forchheimer's relation that will only give positive hydraulic gradients for negative input velocities (specific discharge). This new form (eqn. 2-3b) is illustrated in Figure 2.2b.

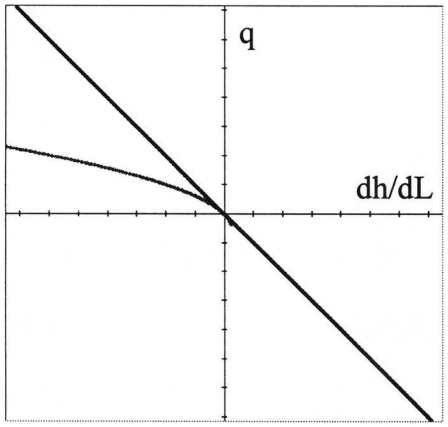
With details of Forchheimer's relation implied, the next step in utilizing equation 2-2b in numerical groundwater models is to solve for specific discharge (q) for both of the equations 2-3a and 2-3b. First, equations 2-3a and 2-3b are rearranged to obtain the standard quadratic form:

$$0 = -bq^2 - aq - \frac{dh}{dL} \qquad (2\text{-}4) \qquad\qquad 0 = bq^2 - aq - \frac{dh}{dL} \qquad (2\text{-}5)$$

Then, the quadratic formula is applied, noting that dh/dL is assumed to be negative in equation 2-4 and positive in equation 2-5. This leaves four possible solutions for specific discharge:

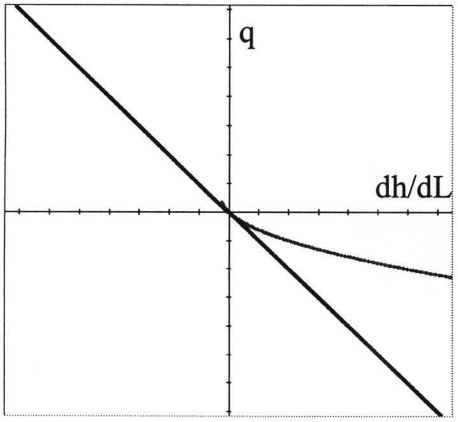
$$q = \frac{a \pm \sqrt{a^2 - 4b\,dh/dL}}{-2b} \qquad (2\text{-}6) \qquad\qquad q = \frac{a \pm \sqrt{a^2 + 4b\,dh/dL}}{2b} \qquad (2\text{-}7)$$

To account for positive and negative input hydraulic gradients inherent in multi-dimensional modeling, two fundamental equations for specific discharge are needed.



$$q = \frac{-a + \sqrt{a^2 - 4b\,dh/dL}}{2b} \qquad \text{(2-8a)}$$

Figure 2.3a: Forchheimer equation for positive discharge values. Parabolic curve is a plot of equation 2-8a, diagonal line is a plot of Darcy's law.



$$q = \frac{a - \sqrt{a^2 + 4b\,dh/dL}}{2b} \qquad \text{(2-8b)}$$

Figure 2.3b: Forchheimer equation for negative discharge values. Parabolic curve is a plot of equation 2-3b, diagonal line is a plot of Darcy's law.

Reduction of the four possible equations to the two required is accomplished by recognizing that specific discharge must equal zero when the hydraulic gradient is zero; there can be no flow without a potential energy gradient. Utilizing this fact results in the two equations that are illustrated in Figures 2.3a and 2.3b.

Again, the hydraulic resistivity and the Forchheimer "b" coefficient are set at unity. The parabolic curves represent the two desired functions listed below each figure, while the diagonal lines represent values calculated using Darcy's law. The reader should be aware that the axes of Figures 2.3a and 2.3b have been inverted from Figures 2.2a and 2.2b, resultant from the change of hydraulic gradient from the dependent variable to the independent variable. In a manner consistent with the previous two figures, equation 2-8a is valid for hydraulic gradients less than zero, while the equation 2-8b is valid for positive hydraulic gradients. Again, the requirement that the hydraulic gradient and specific discharge be of opposite signs is satisfied. It is instructional to note from these graphs that, for any given hydraulic gradient (except zero), the specific discharge will always be less for turbulent flow then for laminar flow.

2.3 Specific Discharge to Head Conversion

The following substitutions are made to convert equation 2-2b, which is in terms of the macroscopic flow velocity q, to a general flow equation in terms of head. For Darcian flow:

$\left(q = -K\,\partial h/\partial L\right)$ or, equivalently, $\left(q = -\frac{1}{a}\,\partial h/\partial L\right)$ is substituted for the specific discharge terms in the continuity equation, resulting in the familiar Darcian flow equation:

$$\frac{1}{a}\left[\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} + \frac{\partial^2 h}{\partial z^2}\right] = S_s \frac{\partial h}{\partial t} \tag{2-9}$$

For nonlinear flow, the derivatives of equations 2-8a,b with respect to distance must be found to facilitate their substitution into equation 2-2b. Beginning with the equation for negative specific discharge (eqn. 2-8b):

$$\frac{dq}{dL} = \frac{d}{dL}\left[\frac{-a + \sqrt{a^2 - 4b\,dh/dL}}{2b}\right] = \frac{d}{dL}\left[\frac{-a}{2b}\right] + \frac{d}{dL}\left[\frac{\sqrt{a^2 - 4b\,dh/dL}}{2b}\right] \tag{2-10a}$$

$$\frac{d}{dL}\left[\frac{-a}{2b}\right] = 0 \tag{2-10b}$$

$$\frac{d}{dL}\left[\frac{\sqrt{a^2 - 4b\,dh/dL}}{2b}\right] = \frac{1}{2}\left(a^2 - 4b\,dh/dL\right)^{-1/2}\left(\frac{-4b\,d^2h/dL^2}{2b}\right) = \frac{-d^2h/dL^2}{\sqrt{a^2 - 4b\,dh/dL}} \tag{2-10c}$$

Now, the derivative with respect to distance for positive specific discharge (eqn. 2-8a) is found:

$$\frac{dq}{dL} = \frac{d}{dL}\left[\frac{a - \sqrt{a^2 + 4b\,dh/dL}}{2b}\right] = \frac{d}{dL}\left[\frac{a}{2b}\right] + \frac{d}{dL}\left[\frac{-\sqrt{a^2 + 4b\,dh/dL}}{2b}\right] \qquad \text{(2-11a)}$$

$$\frac{d}{dL}\left[\frac{a}{2b}\right] = 0 \qquad \text{(2-11b)}$$

$$\frac{d}{dL}\left[-\frac{\sqrt{a^2 + 4b\,dh/dL}}{2b}\right] = -\frac{1}{2}\left(a^2 + 4b\,dh/dL\right)^{-1/2}\left(\frac{4b\,d^2h/dL^2}{2b}\right) = \frac{-d^2h/dL^2}{\sqrt{a^2 + 4b\,dh/dL}} \qquad \text{(2-11c)}$$

Substituting x,y and z for the three axes of Cartesian coordinates, six equations are produced:

	Hyd. Gradient<0	Hyd. Gradient>0
$\partial q_x / \partial x =$	$\dfrac{-d^2h/dx^2}{\sqrt{a^2 - 4b\,dh/dx}}$	$\dfrac{-d^2h/dx^2}{\sqrt{a^2 + 4b\,dh/dx}}$
$\partial q_y / \partial y =$	$\dfrac{-d^2h/dy^2}{\sqrt{a^2 - 4b\,dh/dy}}$	$\dfrac{-d^2h/dy^2}{\sqrt{a^2 + 4b\,dh/dy}}$
$\partial q_z / \partial z =$	$\dfrac{-d^2h/dz^2}{\sqrt{a^2 - 4b\,dh/dz}}$	$\dfrac{-d^2h/dz^2}{\sqrt{a^2 + 4b\,dh/dz}}$

Table 2.1: Directional Forchheimer-based terms used for substitution into continuity equation 2-2b.

The six equations listed in Table 2.1 are then substituted into the mass continuity equation (2-2b) as required, producing a flow equation that has the form:

$$\left[\frac{d^2h/dx^2}{\sqrt{a^2 \pm 4b\,dh/dx}} + \frac{d^2h/dy^2}{\sqrt{a^2 \pm 4b\,dh/dy}} + \frac{d^2h/dz^2}{\sqrt{a^2 \pm 4b\,dh/dz}}\right] = S_s \frac{\partial h}{\partial t} \tag{2-12}$$

Equation (2-13) is very similar to the standard equation for transient Darcian flow :

$$\frac{\partial^2 h/\partial x^2}{a} + \frac{\partial^2 h/\partial y^2}{a} + \frac{\partial^2 h/\partial z^2}{a} = S_s \frac{\partial h}{\partial t} \quad \text{where a=1/K} \tag{2-13}$$

It is instructional to note that if the system is homogeneous and at a steady state (volume flux in equals volume flux out) and there is no external recharge/discharge from the volume then equation 2-13 reduces to the well-known Laplace equation:

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} + \frac{\partial^2 h}{\partial z^2} = 0 \tag{2-14}$$

The character of the Forchheimer-based flow equation (2-12) is significantly different. Equation 2-12 does not reduce to the Laplace equation

2-14 even for homogeneous, steady-state flow unless at a point in the model matrix where the three spatial first-derivatives are equal.

2.4 Incorporation of Point Source/Sink Terms

To account for the point source/sink term(s) in the model matrix that allow for the simulation of well discharge/recharge, modified forms of the flow equations are used.

For Darcian flow:

$$\frac{\partial^2 h / \partial x^2}{a} + \frac{\partial^2 h / \partial y^2}{a} + \frac{\partial^2 h / \partial z^2}{a} = S_s \frac{\partial h}{\partial t} - \frac{Q}{\Delta x \Delta y \Delta z} \tag{2-15}$$

For non-Darcy flow:

$$\left[\frac{d^2 h / dx^2}{\sqrt{a^2 \pm 4b\, dh/dx}} + \frac{d^2 h / dy^2}{\sqrt{a^2 \pm 4b\, dh/dy}} + \frac{d^2 h / dz^2}{\sqrt{a^2 \pm 4b\, dh/dz}} \right] = S_s \frac{\partial h}{\partial t} - \frac{Q}{\Delta x \Delta y \Delta z} \tag{2-16}$$

Where Q is the volume of fluid added or subtracted per cell volume per unit time.

Equations 2-15 and 2-16 represent the general forms of the fundamental equations used in this study to evaluate the error incurred when Darcy's law is assumed to be valid.

With these equations in this form, the relationship between the Darcian and Forchheimer based flow equations can be examined. The smallest possible value for the denominators on the left-hand side of equation 2-16 is "a". It is apparent that for a given spatial hydraulic gradient, the time rate of change of hydraulic head calculated using the Forchheimer based flow equation will always be less than the temporal hydraulic gradient calculated using the Darcy-based equation. It is also apparent that deviation from Darcian flow will be dependent upon the values given to the "a" coefficient and the "b" coefficient. If the value of "a" is much greater than the value of "b" then, with the exception of extremely large hydraulic gradients, the terms:

$$4b\,\partial h/\partial x \qquad 4b\,\partial h/\partial y \qquad 4b\,\partial h/\partial z$$

are negligible, causing the denominators on the left side of equation 2-16 to be approximately equal to "a". When this is the case, the values calculated using the Forchheimer-based equation are virtually identical to values calculated using the Darcian equation. For large values of the ratio of "b" to "a" or for large spatial hydraulic gradients, the solutions obtained with equation 2-16 can vary significantly from the Darcian solutions. The onset of departure from Darcian linearity is heavily dependent upon the average grain size of the matrix (which dictates the resistivity coefficients in this model) as well as the magnitude of stresses (well discharge, recharge rates etc.) imposed on the model. A full discussion on the effects caused by various model parameters will follow in Chapter 6.

Chapter 3. Finite-Difference Approximation

For this investigation, two-dimensional flow to a fully penetrating well in a confined aquifer was modeled. Homogeneous, steady-flow simulations were run for calibration purposes, while heterogeneous and transient effects were incorporated to study their effects on the error produced when using Darcy-based models.

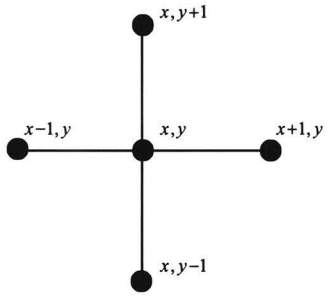


Figure 3.1: 5-Point Matrix Node Arrangement

As shown in figure 3.1, the arrangement of nodes in the model matrix are orthagonally oriented and regularly spaced. Based on this configuration, the equation for two-dimensional, Darcy-based, steady flow in an aquifer of the thickness "M" is equation 3-1:

$$\frac{\partial^2 h / \partial x^2}{a} + \frac{\partial^2 h / \partial y^2}{a} = -\frac{Q}{\Delta x^2 M} \qquad (3\text{-}1)$$

The flow equation for two-dimensional, Forchheimer-based, steady flow in an aquifer of the thickness "M" is equation 3-2:

$$\frac{d^2h/dx^2}{\sqrt{a^2 \pm 4b\, dh/dx}} + \frac{d^2h/dy^2}{\sqrt{a^2 \pm 4b\, dh/dy}} = -\frac{Q}{\Delta x^2 M} \qquad (3\text{-}2)$$

3.1 Finite-Difference Derivative Approximations

In order to convert equations 3-1 and 3-2 into finite-difference form, the following derivative approximations are employed for this model (e.g., in the x-direction):

$$\partial h/\partial x \approx \frac{h_{x+1,y} - h_{x,y}}{\Delta x} \qquad (3\text{-}3a) \qquad\qquad \partial^2 h/\partial x^2 \approx \frac{h_{x+1,y} - 2h_{x,y} + h_{x-1,y}}{\Delta x^2} \qquad (3\text{-}3b)$$

To derive a finite-difference analog for the first derivative, Taylor series approximations are employed. The Taylor series for a function *h* at the position (x+1) is

$$h_{x+1,y} = h_{x,y} + \left(\frac{\partial h}{\partial x}\right)_{x,y} \Delta x + \left(\frac{\partial^2 h}{\partial x^2}\right)_{x,y} \frac{(\Delta x)^2}{2!} + \left(\frac{\partial^3 h}{\partial x^3}\right)_{x,y} \frac{(\Delta x)^3}{3!} + \ldots + \qquad (3\text{-}4)$$

Equation 3-5 results when equation 3-4 is solved for the first derivative.

$$\left(\frac{\partial h}{\partial x}\right)_{x,y} = \frac{h_{x+1,y} - h_{x,y}}{\Delta x} - \left(\frac{\partial^2 h}{\partial x^2}\right)_{x,y} \frac{\Delta x}{2!} - \left(\frac{\partial^3 h}{\partial x^3}\right)_{x,y} \frac{(\Delta x)^2}{3!} - \left(\frac{\partial^4 h}{\partial x^4}\right)_{x,y} \frac{(\Delta x)^3}{4!} - \ldots - \qquad (3\text{-}5)$$

The first term on the right hand side of equation 3-5 is the finite-difference approximation stated above as equation 3-3a. This analog is first order correct because the first term truncated on the right hand side contained delta-x raised to the first power.

The analog for the second derivative is also obtained by using Taylor series approximations. The value of a function *h* at position (x-1) is given by equation 3-6:

$$h_{x-1,y} = h_{x,y} - \left(\frac{\partial h}{\partial x}\right)_{x,y} \Delta x + \left(\frac{\partial^2 h}{\partial x^2}\right)_{x,y} \frac{(\Delta x)^2}{2!} - \left(\frac{\partial^3 h}{\partial x^3}\right)_{x,y} \frac{(\Delta x)^3}{3!} + \ldots + \qquad (3\text{-}6)$$

Equations 3-4 and 3-6 are then added together resulting in equation 3-7.

$$h_{x-1,y} + h_{x-1,y} = 2h_{x,y} + \left(\frac{\partial^2 h}{\partial x^2}\right)_{x,y} (\Delta x)^2 + 2\left(\frac{\partial^4 h}{\partial x^4}\right)_{x,y} \frac{(\Delta x)^4}{4!} + \ldots + \qquad (3\text{-}7)$$

Equation 3-7 is then solved for the second derivative producing equation 3-8.

$$\left(\frac{\partial^2 h}{\partial x^2}\right)_{x,y} = \frac{h_{x-1,y} - 2h_{x,y} + h_{x-1,y}}{(\Delta x)^2} - \left(\frac{\partial^4 h}{\partial x^4}\right)_{x,y} \frac{(\Delta x)^2}{12} - \ldots - \qquad (3\text{-}8)$$

If the right-hand side of equation 3-8 is truncated after the first term, the finite-difference approximation for the second spatial derivative (eqn. 3-3b) is obtained. This approximation is said to be second-order correct because the first term truncated contains $(\Delta x)^2$.

3.2 CONVERSION OF STEADY-STATE FLOW EQUATIONS

To convert the spatially continuous flow equations derived in Chapter 2 into finite-difference form, equations 3-3a and 3-3b are appropriately substituted into equations 3-1 and 3-2. This is a straightforward process, however, some technical difficulties do arise.

First, during the calibration process, it was found that the method of calculation of the apparent hydraulic resistivity (D_x) had a significant influence on the solution surface.

$$\text{Where apparent hydraulic resistivity} = D_x = \sqrt{a^2 + 4b\left|\partial h / \partial x\right|} \qquad (3\text{-}9)$$

Following many trial simulations, it was determined that the most accurate results were obtained by applying an averaging method. Essentially, the apparent hydraulic resistivity at a node is the arithmetic mean of the resistivities calculated for the region between the node in question and it's adjacent nodes. In practice, the first step is the calculation of the hydraulic gradient between the node of interest and its adjacent nodes parallel to a Cartesian axis (e.g. the x-axis):

$$dh/dx_{east} \approx \frac{h_{x+1,y} - h_{x,y}}{\Delta x} \qquad (3\text{-}10a) \qquad dh/dx_{west} \approx \frac{h_{x,y} - h_{x-1,y}}{\Delta x} \qquad (3\text{-}10b)$$

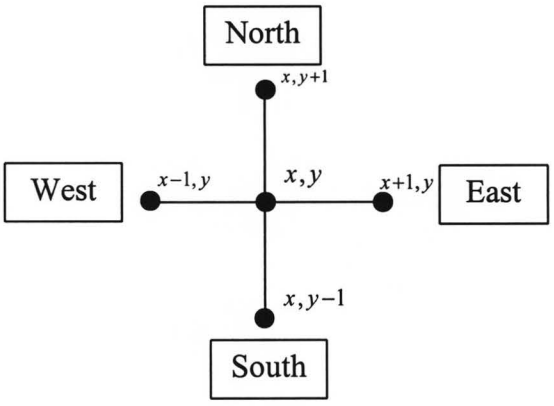


Figure 3.2 Arrangement and nomenclature of model nodes incorporated into equations in Chapter 3.

These values are then substituted into equations (3-11a,b), resulting in two values of apparent hydraulic resistivity.

$$D_{x\_east} = \sqrt{a^2 + 4b\left|\partial h / \partial x_{east}\right|} \quad \text{(3-11a)} \qquad D_{x\_west} = \sqrt{a^2 + 4b\left|\partial h / \partial x_{west}\right|} \quad \text{(3-11b)}$$

These values are then averaged to find the apparent hydraulic resistivity used in the calculation of head at that node.

$$D_x = \frac{\sqrt{a^2 + 4b\left|\partial h / \partial x_{east}\right|} + \sqrt{a^2 + 4b\left|\partial h / \partial x_{west}\right|}}{2} \quad \text{(3-12)}$$

The value of apparent hydraulic resistivity parallel to the y-axis (north-south as per Figure 3.2) is found in an analogous manner.

Following substitution of equation 3-12 (and it's analogous y-axis counterpart), equation 3-2 becomes:

$$\frac{h_{x+1,y} - 2h_{x,y} + h_{x-1,y}}{\Delta x^2 D_x} + \frac{h_{x,y+1} - 2h_{x,y} + h_{x,y-1}}{\Delta y^2 D_y} = \frac{-Q_{x,y}}{\Delta x \Delta y M} \qquad (3\text{-}13a)$$

Assuming that $\Delta x = \Delta y$, equation 2-13a is then solved for $h_{x,y}$:

$$\frac{h_{x+1,y} + h_{x-1,y}}{D_x} + \frac{h_{x,y+1} + h_{x,y-1}}{D_y} + \frac{Q_{x,y}}{M} = \frac{2h_{x,y}}{D_x} + \frac{2h_{x,y}}{D_y} \qquad (3\text{-}13b)$$

$$\left(\frac{h_{x+1,y} + h_{x-1,y}}{D_x} + \frac{h_{x,y+1} + h_{x,y-1}}{D_y} + \frac{Q_{x,y}}{M} \right) \left(\frac{D_x D_y}{2D_y + 2D_x} \right) = h_{x,y} \qquad (3\text{-}13c)$$

For Darcian flow, the approximation for the second spatial derivative (eqns. 3-3b) is substituted into equation 3-1, and then solved for $h_{x,y}$:

$$\left(\frac{h_{x+1,y} + h_{x-1,y}}{a} + \frac{h_{x,y+1} + h_{x,y-1}}{a} + \frac{Q_{x,y}}{\Delta x \Delta y M} \right) \left(\frac{a^2}{2a + 2a} \right) = h_{x,y} \qquad (3\text{-}14)$$

Equations 3-13c and 3-14 can be applied to all the nodes in the model matrix, which is then "solved" iteratively. Written in this form both equations automatically incorporate the iterative procedure known as the Gauss-Seidel method (Wang and Anderson, 1982). Using this procedure, the matrix of nodes is systematically swept through during each iteration, and the heads calculated during the current iteration sweep are immediately incorporated into the solution set. In other words, heads calculated during the current

iteration are available for the calculation of heads in neighboring cells during the same iteration.

Other technical problems arise during the implementation of derivative analogs. It is obvious that the derivative approximations (eqns. 3-3a,b) become undefined at nodes on the border of the model domain. In order to compensate for this, a perimeter of "fictitious nodes" (von Rosenberg, 1969) is included in the numerical matrix. Fictitious nodes are nodes that are not technically part of the model matrix, but are assigned values during the iteration procedure that can be drawn upon for the calculation of derivatives.

3.3 Finite-Difference Conversion of Transient Flow Equations

The equation for two-dimensional, transient Darcian flow in an aquifer of the thickness "M" is equation (3 -15):

$$\frac{\partial^2 h / \partial x^2}{a} + \frac{\partial^2 h / \partial y^2}{a} = S_s \frac{\partial h}{\partial t} - \frac{Q}{\Delta x^2 M} \qquad (3\text{-}15)$$

The flow equation for two-dimensional, transient non-Darcian flow in an aquifer of the thickness "M" is equation 3-16:

$$\frac{d^2h/dx^2}{\sqrt{a^2+4b\left|dh/dx\right|}}+\frac{d^2h/dy^2}{\sqrt{a^2+4b\left|dh/dy\right|}}=S_s\frac{\partial h}{\partial t}-\frac{Q}{\Delta x^2 M} \qquad (3\text{-}16)$$

Because of its inherent stability, an implicit iteration procedure was used (von Rosenberg, 1969). Using the implicit method, heads are calculated using values at the future (t+1) time-step. This insures that the solution matrix will converge irrespective of the choice of nodal spacing or time-step length (Quinney, 1985). Equations 3-17 and 3-18 show the implicit forms of the second spatial and first temporal derivatives respectively.

$$\partial^2 h/\partial x^2 \approx \frac{h_{x+1,y}^{t+1}-2h_{x,y}^{t+1}+h_{x-1,y}^{t+1}}{\Delta x^2} \qquad (3\text{-}17)$$

$$\partial h/\partial t \approx \frac{h_{x,y}^{t+1}-h_{x,y}^{t}}{\Delta x} \qquad (3\text{-}18)$$

In order to linearize the nonlinear denominators on the left-hand side of equation 3-16, the first spatial derivative was evaluated at an intermediate (t+1/2) time level. This is accomplished by averaging gradient values obtained using future head values and gradient values evaluated at the prior time step (von Rosenberg, 1969). Equations 3-19 to 3-23 illustrate the procedure used in this process.

$$\left(dh/dx\right)_{east}^{t+1} \approx \frac{h_{x+1,y}^{t+1}-h_{x,y}^{t+1}}{\Delta x} \qquad (3\text{-}19a)$$

$$\left(dh/dx\right)_{west}^{t+1} \approx \frac{h_{x,y}^{t+1}-h_{x-1,y}^{t+1}}{\Delta x} \qquad (3\text{-}19b)$$

$$\left(dh/dy\right)_{north}^{t+1} \approx \frac{h_{x,y+1}^{t+1} - h_{x,y}^{t+1}}{\Delta y} \quad \text{(3-20a)} \qquad \left(dh/dy\right)_{south}^{t+1} \approx \frac{h_{x,y}^{t+1} - h_{x,y-1}^{t+1}}{\Delta y} \quad \text{(3-20b)}$$

$$\left(dh/dx\right)_{east}^{t} \approx \frac{h_{x+1,y}^{t} - h_{x,y}^{t}}{\Delta x} \quad \text{(3-21a)} \qquad \left(dh/dx\right)_{west}^{t} \approx \frac{h_{x,y}^{t} - h_{x-1,y}^{t}}{\Delta x} \quad \text{(3-21b)}$$

$$\left(dh/dy\right)_{north}^{t} \approx \frac{h_{x,y+1}^{t} - h_{x,y}^{t}}{\Delta y} \quad \text{(3-22a)} \qquad \left(dh/dy\right)_{south}^{t} \approx \frac{h_{x,y}^{t} - h_{x,y-1}^{t}}{\Delta y} \quad \text{(3-22b)}$$

$$\left(dh/dx\right)^{t+1/2} \approx \frac{\left(\frac{\left(\partial h/\partial x\right)_{east}^{t+1} + \left(\partial h/\partial x\right)_{west}^{t+1}}{2}\right) + \left(\frac{\left(\partial h/\partial x\right)_{east}^{t} + \left(\partial h/\partial x\right)_{west}^{t}}{2}\right)}{2} \quad \text{(3-23a)}$$

$$\left(dh/dy\right)^{t+1/2} \approx \frac{\left(\frac{\left(\partial h/\partial y\right)_{north}^{t+1} + \left(\partial h/\partial y\right)_{south}^{t+1}}{2}\right) + \left(\frac{\left(\partial h/\partial y\right)_{north}^{t} + \left(\partial h/\partial y\right)_{south}^{t}}{2}\right)}{2} \quad \text{(3-23b)}$$

Once the derivative analogs have been calculated using equations 3-19 to 3-23, they are substituted into equation 3-16 to produce equation 3-24.

$$\frac{h_{x+1,y}^{t+1} - 2h_{x,y}^{t+1} + h_{x-1,y}^{t+1}}{\Delta x^2 \sqrt{a^2 + 4b\left|\partial h/\partial x\right|^{t+1/2}}} + \frac{h_{x,y+1}^{t+1} - 2h_{x,y}^{t+1} + h_{x,y-1}^{t+1}}{\Delta x^2 \sqrt{a^2 + 4b\left|\partial h/\partial y\right|^{t+1/2}}} = S_s \frac{h_{x,y}^{t+1} - h_{x,y}^{t}}{\Delta t} - \frac{Q}{\Delta x^2 M} \quad \text{(3-24)}$$

For the sake of readability, equations 3-25a and 3-25b are substituted to produce equation 3-26a.

$$D_x = \sqrt{a^2 + 4b\left|dh/dx\right|}^{t+1/2} \qquad \text{(3-25a)}$$

$$D_y = \sqrt{a^2 + 4b\left|dh/dy\right|}^{t+1/2} \qquad \text{(3-25b)}$$

Equation 3-26a his solved for $h_{x,y}^{t+1}$, resulting in equation 3-26c.

$$\frac{h_{x+1,y}^{t+1} - 2h_{x,y}^{t+1} + h_{x-1,y}^{t+1}}{\Delta x^2 D_x} + \frac{h_{x,y+1}^{t+1} - 2h_{x,y}^{t+1} + h_{x,y-1}^{t+1}}{\Delta x^2 D_y} = S_s \frac{h_{x,y}^{t+1} - h_{x,y}^{t}}{\Delta t} - \frac{Q_{x,y}}{\Delta x^2 M} \qquad \text{(3-26a)}$$

$$\frac{h_{x+1,y}^{t+1} + h_{x-1,y}^{t+1}}{D_x} + \frac{h_{x,y+1}^{t+1} + h_{x,y-1}^{t+1}}{D_y} + \frac{Q_{x,y}}{M} = \frac{S_s \Delta x^2 h_{x,y}^{t+1}}{\Delta t} - \frac{S_s \Delta x^2 h_{x,y}^{t}}{\Delta t} + \frac{2h_{x,y}^{t+1}}{D_x} + \frac{2h_{x,y}^{t+1}}{D_y} \qquad \text{(3-26b)}$$

$$\left(\frac{h_{x+1,y}^{t+1} + h_{x-1,y}^{t+1}}{D_x} + \frac{h_{x,y+1}^{t+1} + h_{x,y-1}^{t+1}}{D_y} + \frac{Q_{x,y}}{M} + \frac{S_s \Delta x^2 h_{x,y}^{t}}{\Delta t} \right) * \left(\frac{1}{S_s \Delta x^2 / \Delta t + 2/D_x + 2/D_y} \right) = h_{x,y}^{t+1} \qquad \text{(3-26c)}$$

Equation 3-26c is the final form used in this model used to calculate head during transient flow simulations.

3.4 INCORPORATION OF HETEROGENEITIES

For a block-centered nodal array, the Forchheimer coeficients (a,b) in equations 3-15 and 3-16 are assumed to correspond to the region of space between two adjacent nodes in the matrix. Consequently, in a heterogeneous matrix there can be up to four pairs of resistivity coefficients under consideration for each node undergoing calculation. Figure 3.1 illustrates the spatial relationship between nodes and regions of hydraulic resistivity. The flowpath volume between adjacent nodes centered in regions of differing hydraulic resistivity is subdivided into two regions. In order to accurately predict the flow between the nodes, the effective resistivity (or conductivity) of the whole flowpath volume must be found. This has traditionally been accomplished by finding the harmonic mean of the hydraulic conductivities of the adjacent cells (Domenico and Schwartz, 1990).

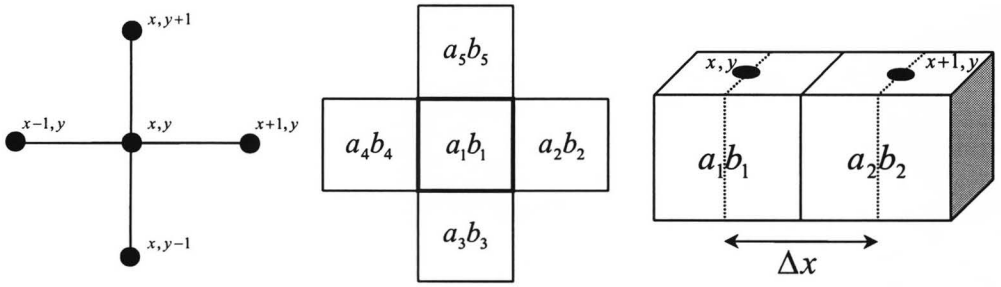


Figure 3.3: Local configuration of matrix nodes and parameter boundaries.

Equation 3-27 illustates this process for a flowpath volume divided equally between two porous mediums: the first material of hydraulic conductivity equal to K_1 and a second of hydraulic conductivity equal to K_2.

$$\overline{K} = \frac{2}{1/K_1 + 1/K_2} \tag{3-27}$$

When concerned with the calculation of the effective resistivity of a pair of porous mediums:

$$\bar{a} = 1/\bar{K} = \frac{1/K_1 + 1/K_2}{2} = \frac{a_1 + a_2}{2} \tag{3-28}$$

$$\bar{b} = \frac{b_1 + b_2}{2} \tag{3-29}$$

$$\bar{a}_{east} = \frac{a_{x,y} + a_{x+1,y}}{2} \tag{3-30a}$$

$$\bar{a}_{west} = \frac{a_{x,y} + a_{x-1,y}}{2} \tag{3-30b}$$

$$\bar{a}_{north} = \frac{a_{x,y} + a_{x,y+1}}{2} \tag{3-30c}$$

$$\bar{a}_{south} = \frac{a_{x,y} + a_{x,y-1}}{2} \tag{3-30d}$$

$$\bar{b}_{east} = \frac{b_{x,y} + b_{x+1,y}}{2} \tag{3-31a}$$

$$\bar{b}_{west} = \frac{b_{x,y} + b_{x-1,y}}{2} \tag{3-31b}$$

$$\bar{b}_{north} = \frac{b_{x,y} + b_{x,y+1}}{2} \tag{3-31c}$$

$$\bar{b}_{south} = \frac{b_{x,y} + b_{x,y-1}}{2} \tag{3-31d}$$

Once the mean directional resistivity coefficients have been calculated using equations 3-30 and 3-31, these coefficients are averaged using equations

3-32 and 3-33 to obtain the values that will be substituted into the governing equations 3-15 and 3-16.

$$a_x = \frac{\bar{a}_{east} + \bar{a}_{west}}{2} \qquad (3\text{-}32a) \qquad\qquad a_y = \frac{\bar{a}_{north} + \bar{a}_{south}}{2} \qquad (3\text{-}32b)$$

$$b_x = \frac{\bar{b}_{east} + \bar{b}_{west}}{2} \qquad (3\text{-}33a) \qquad\qquad b_y = \frac{\bar{b}_{north} + \bar{b}_{south}}{2} \qquad (3\text{-}33b)$$

Chapter 4. Model Calibration

In order to verify the accuracy of the solutions produced by the Darcy-based and Forchheimer-based equations, the output produced through various simulations was compared to analytically derived solution surfaces. The analytical solutions for radial, homogeneous flow to a well are obtained through relatively simple integrations of the Forchheimer and Darcy differential equations (Halihan 1999, see Appendix 1).

4.1 Model Construction and Parameters

As implemented, the model consists of a square numerical matrix of nodes. Figure 4.1 illustrates the distribution of node types used for model verification simulations. A central node represents a well through which discharge (Q) is occurring. A radial distribution of constant head nodes represent a recharge boundary in the aquifer where head is known.

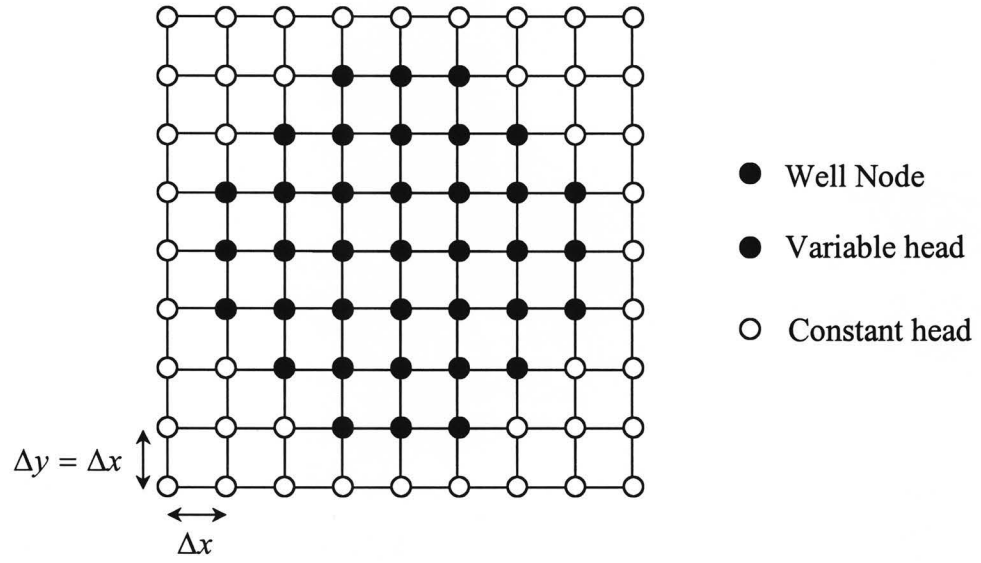


Figure 4.1: Generalized distribution of model node types.

At each of these nodes, numerical arrays were constructed that specified or recorded the hydraulic resistivity, non-linear "b" coefficient, average grain size, Darcian and non-linear head, Darcian and non-linear specific discharge (q) in the x and y directions as well as Reynolds number. For the model calibration procedure, the values of the hydraulic resistivity (a) and the Forchheimer non-linear coefficient "b" did not vary with Cartesian direction at each node, and their values were constant throughout the model matrix, resulting in a simulated homogenous, isotropic medium.

In all model verification simulations the radial distance between the well node and the constant head boundary was set at 100 meters. This

distance was subdivided by varying numbers of nodes in separate simulations in order to verify solution convergence with increasing nodal density.

A non-zero term in the discharge array simulated a fully-penetrating well at the center of the matrix in all simulations. The thickness and initial head of the modeled aquifer were always constant (specified during the initialization procedures of each run). Convergence tolerance was somewhat varied, but was always between 0.05 mm and 0.005 mm, less than well-test measurement errors.

4.2 Difficulties Encountered

It was found that when source or sink terms (well discharge/recharge) were present the matrix the numerical solution did not match the analytical solution. Unacceptable deviations (up to and in excess of 100 percent) from analytically derived surfaces were obtained in simulations where the flow was highly non-linear. This is a common difficulty encountered in the numerical representation of source and sink terms in aquifer simulation.

The problem is partially due to the nature of finite-difference discretization. Because computers are only capable of manipulating discrete data, a continuous porous matrix must be subdivided into areas that are represented by single point values (nodes). By definition, the point value of

head calculated at a node represents the average value of head within that block. This creates difficulties for blocks that contain source or sink terms that represent well recharge/discharge. The pressure calculated for a node within a well block does not accurately reflect the hydraulic gradient that is present within the grid block itself. In effect, the assumption is made that the entire volume of the grid block containing well is experiencing homogeneous flow. Unless the grid block length is assigned to be equal to the modeled well radius, this assumption is obviously false for the grid block containing a well. Because this model does not incorporate variable grid block length, it was not computationally economical to refine the grid to this degree.

The error incurred during the discretization of an aquifer can cause unwanted deviations in calculated drawdown in both Darcy-based and Forchheimer based models. However, the deviation from analytical solutions is generally much smaller for the Darcian model in high-flow, coarse-grained simulations conducted for this study. One reason for the smaller deviation in the solution surface produced by the Darcy-based model is that finite-difference gradient approximations are linear by design. This results in less error when these approximations are applied to linear partial-differential equations. For fine-grained, low-flow simulations, the difference in calculated head between the two models was less than the assigned tolerance.

Numerical errors are compounded when the Forchheimer-based model is applied. During high-flow conditions, hydraulic gradients are greater for the Forchheimer-based models than for the Darcy-based model. These larger gradients serve to amplify the error contracted by the well block/node averaging process discussed above. In addition, errors are enlarged because the apparent hydraulic resistivity of a grid block varies with hydraulic gradient when the Forchheimer-based model is applied. The over-estimation of hydraulic gradient at the well node creates a relatively large area in the model grid with significantly raised hydraulic resistivity. The result of this numerical over-estimation in resistivity is that drawdown is increased at the well block, and then transmitted to the rest of the model matrix.

The situation at the well node is further complicated by the non-differentiability of the solution curve at the well node. Because the withdrawal term in the continuity equation at the well node creates a "corner" in the potentiometric surface (Figure 4.2) rendering the solution curve non-continuous and therefore undifferentiable. Center-difference approximations traditionally employed in groundwater and reservoir models fail to accurately estimate the hydraulic gradient at this point.

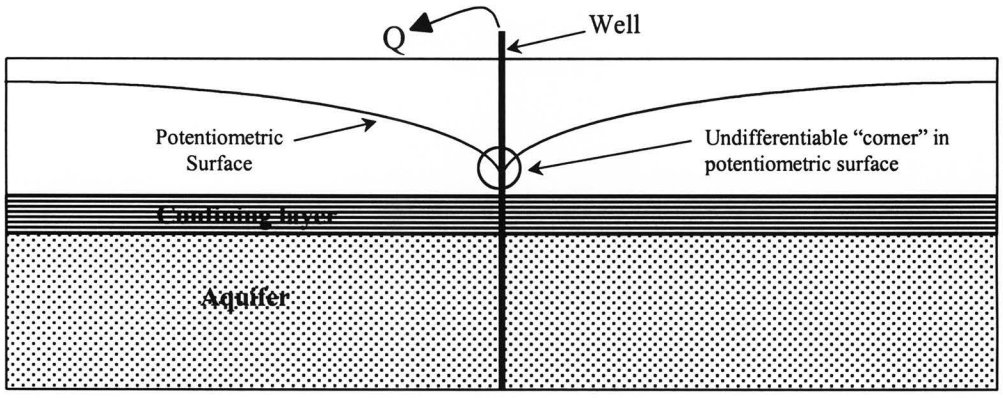


Figure 4.2: Simplified aquifer cross-section showing the location of the undifferentiable "corner" in the solution surface.

Figure 4.3 illustrates the excessive drawdown caused by these well-node effects for homogeneous, radial flow to a well centered in a square model matrix of thickness "M". The approximate average grain size for this simulation was 5 mm, resulting in a hydraulic resistivity of 10 seconds per meter. The curves plotted below represent a cross-section of the potentiometric surface calculated along a line from the well node to a constant-head boundary. As Figure 4.3 shows, the curve generated by the Forchheimer-based model is offset from the curves calculated using analytical solutions (Appendix 1), as well as the curve generated by the Darcy-based model. It is apparent that a straightforward application of the Forchheimer-based flow equation to the well node results in an unacceptable increase in the drawdown predicted by the model.

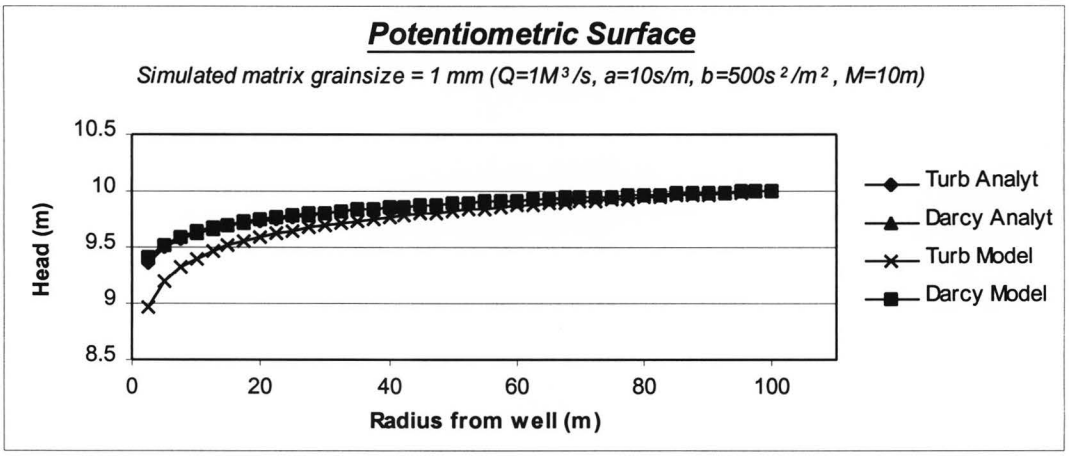


Figure 4.3: Potentiometric surface cross-section showing excessive drawdown of the Forchheimer-based model near the well node.

Many authors have developed methods for correcting for the excessive hydraulic gradient caused by these effects. Most often, this involves the application of so-called well models (Peaceman, 1978, 1990), (Willaimson and Chappelear, 1981), (Abou-Kassem and Aziz, 1985) and (Hayes, Kendall and Wheeler, 1977). The most widely accepted method of correction is D.W. Peaceman's application of a well model. Using Peaceman's method, well-block pressure values (and therefore discharge values) obtained through numerical methods are modified with values calculated using an analytical solution of flow to a well. Unfortunately, successful implementation of these established well models assumes the validity of Darcy's law, and is therefore difficult to apply to non-linear flow.

Application of a non-Darcy well-model method proved difficult. Analogous to the Peaceman technique, over-pressures were modified using the integrated form of the Forchheimer equation. However, the solution curves produced using this method also failed to match the drawdowns predicted by the analytical solution. This is probably do to the Forchheimer-based model's greater sensitivity to spatial hydraulic gradient than that of traditional Darcy-based flow models.

4.3 Upstream Apparent Hydraulic Conductivity Weighting

Because of the difficulty found in applying traditional corrective techniques, an empirical approach was taken to ascertain a new method of correction at the well node.

As shown by the equations in Figure 4.4, the difference between the Darcy-based and the Forchheimer-based continuity equations is the presence of the non-linear hydraulic resistivity coefficients (D_x and D_y) of the second spatial derivatives. Therefore, a procedure must be found to approximate the apparent hydraulic resistivity at the well node that allows the numerically calculated solution curve to mimic the behavior of the analytical solution curve. This could be accomplished by either of two methods: alteration of the head values at the well node and surrounding nodes, or direct alteration of the

values of the resistivity coefficients (D_x and D_y) at the well node prior to substitution into the continuity equation.

$$\frac{\partial^2 h/\partial x^2}{a} + \frac{\partial^2 h/\partial y^2}{a} = \frac{-Q}{\Delta x \Delta y M}$$

Darcy-based

$$\frac{d^2 h/dx^2}{D_x} + \frac{d^2 h/dy^2}{D_y} = \frac{-Q}{\Delta x \Delta y M}$$

Forchheimer-based

Figure 4.4: Darcy and Forchheimer-based flow equations.

Where $D_x = \sqrt{a^2 + 4b\left|\partial h/\partial x\right|}$ and $D_y = \sqrt{a^2 + 4b\left|\partial h/\partial y\right|}$.

Following the modification of the program code responsible for calculation of apparent hydraulic resistivity, and completion of numerous simulations, it was found that direct modification of the apparent hydraulic resistivity at the well node was simpler and produced more accurate results than modification of the hydraulic gradient affecting the well node.

Solution curves that matched analytical results reasonably well were obtained by assigning an apparent hydraulic resistivity at the well node that was a weighted average of the two upstream nodes. Figure 4.5 shows the portion of the model matrix that is evaluated when assigning the apparent hydraulic resistivity to the well block.

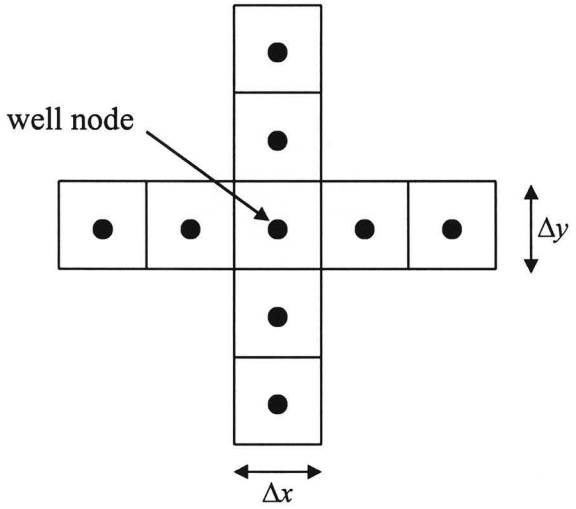


Figure 4.5: Diagram of the model matrix blocks used when assigning resistivity values to the well node.

The first step in approximating the hydraulic resistivity of the well node is the calculation of D_x and D_y for the eight nodes surrounding the well node. This is accomplished by individually calculating the apparent hydraulic resistivity for each of the eight nodes question and their four neighboring nodes (Figure 4.6) .

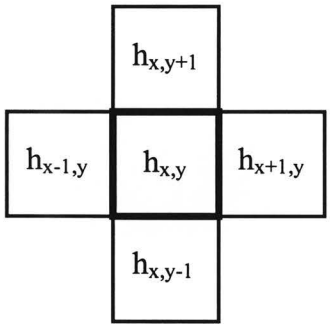


Figure 4.6: Arrangement and nomenclature used in parameter calculation.

Specifically, the two hydraulic resistivities along the x-axis are calculated and then averaged to find D_x for each node. The apparent hydraulic resistivity along the y-axis (D_y) is found in a similar manner.

$$D_x = \frac{\sqrt{a^2 + 4b\left|\frac{h_{x,y} - h_{x-1,y}}{\Delta x}\right|} + \sqrt{a^2 + 4b\left|\frac{h_{x,y} - h_{x+1,y}}{\Delta x}\right|}}{2} \qquad \text{(4-1a)}$$

$$D_y = \frac{\sqrt{a^2 + 4b\left|\frac{h_{x,y} - h_{x,y-1}}{\Delta y}\right|} + \sqrt{a^2 + 4b\left|\frac{h_{x,y} - h_{x,y+1}}{\Delta y}\right|}}{2} \qquad \text{(4-1b)}$$

Once the hydraulic resistivities for each of the eight nodes surrounding the well node have been calculated, the hydraulic resistivities of the well node are calculated using equations (4-1a,b). For the sake of clarity the eight upstream nodes are differentiated according to cardinal direction and radial distance from the well node (Figure 4.7).

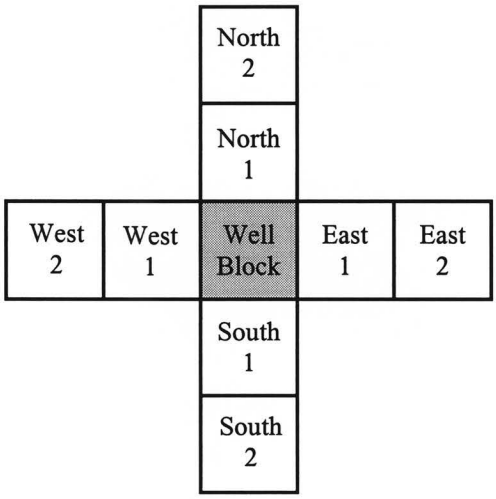


Figure 4.7: Arrangement and nomenclature of matrix blocks.

The apparent hydraulic resistivities along the x-axis for the four nodes at a distance of Δx is averaged and the same is done for the four nodes that they distance of $2\Delta x$. (eqns. 4-2a,b)

$$D_x^{radius=2} = \frac{D_x^{North\,2} + D_x^{South\,2} + D_x^{East\,2} + D_x^{West\,2}}{4} \qquad (4\text{-}2a)$$

$$D_x^{radius=1} = \frac{D_x^{North\,1} + D_x^{South\,1} + D_x^{East\,1} + D_x^{West\,1}}{4} \qquad (4\text{-}2b)$$

These values are then supplied to equation 4-3. A weighted average of the two values is calculated according to the parameter "C" which is in turn given by equations 4-4 and 4-5. The apparent hydraulic resistivity along the y-axis is calculated in an analogous manner. Equation 4-3 varies from an

equally weighted average between resistivities calculated at a radius of $2\Delta x$ and Δx for linear flows, to an average that is almost wholly weighted in favor of the hydraulic resistivities calculated at a radius of $2\Delta x$ for highly non-Darcian flows.

$$D_x^{well} = \frac{C * D_x^{radius=1} + (2 - C) * D_x^{radius=2}}{2} \qquad (4\text{-}3)$$

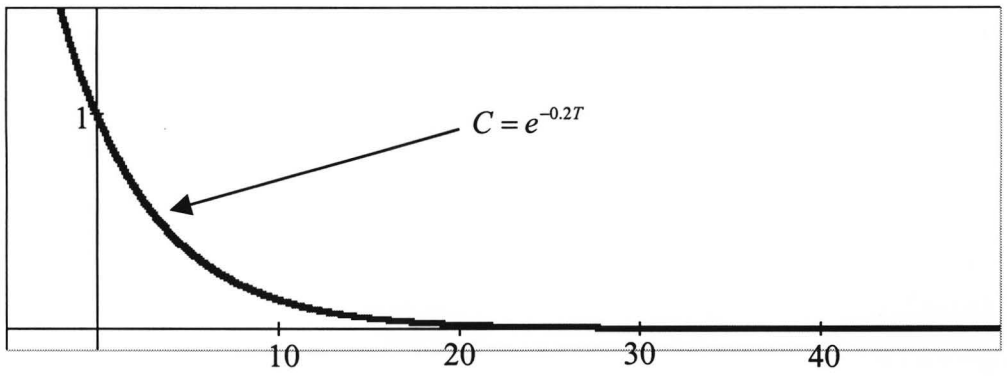
$$T = \frac{Q\sqrt{b/a}}{\Delta x M} \qquad (4\text{-}4) \qquad C = e^{-0.2T} \qquad (4\text{-}5)$$

Equations 4-2 through 4-5 were derived by simple trial and error, however, there is a rational basis behind the choice of terms. Trial simulations showed that increasing values of discharge increased non-Darcy behavior. Increased non-linearity also resulted from an increase in the value of the ratio of b to a, but to a much lesser degree than that calculated for increases in discharge. Decreases in nonlinear behavior were found to result from increases in matrix element length and formation thickness. Given these observations, it follows that these parameters should exert a significant influence on the solution curve (therefore influencing the calculation of well-node resistivity during simulations) in the vicinity of the well.

It was also noted during trial simulations that the effect of these parameters on the solution curve decreased rapidly as the value of "T" increased. Experimental simulations were run in which the parameter "C"

was calculated using linear, inverse power, inverse logarithmic and decaying exponential functions of the parameter "T".

The decaying exponential form (eqn. 4-5, Fig. 4.8) produced the most accurate results (as compared with the analytical solution) over the widest range of input parameters.



$$T = \frac{Q\sqrt{b/a}}{\Delta x M}$$

Figure 4.8: Plot of weighting coefficient "C".

The function illustrated by Figure 4.8 was also the simplest and most computationally efficient form, therefore it was chosen for application in this model.

The matrix blocks surrounding the well node are shown in Figure 4.9. As illustrated, the eight grid blocks located in the two-point upstream positions (oriented in the four cardinal directions) have a direct influence on the calculated value of apparent hydraulic resistivity applied to the well block.

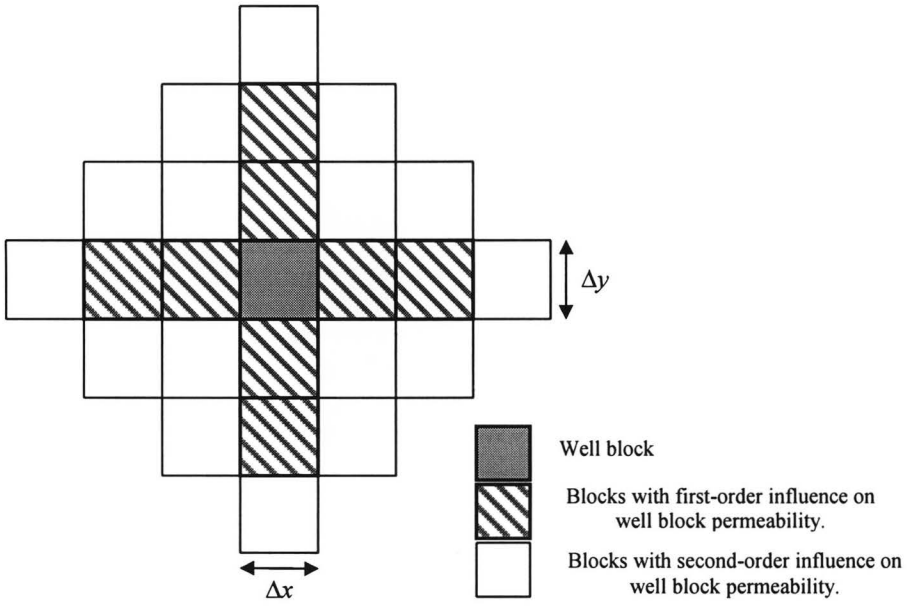


Figure 4.9: Region of model blocks with primary influence on the calculation of well node apparent hydraulic resistivity.

Sixteen other grid blocks also have influence on the calculation of resistivity during a model iteration in that the heads of these blocks are used to calculate the resistivity of the eight primary blocks.

4.4 CALIBRATION RESULTS

For calibration, an extremely simple model layout was used. Figure 4.10 illustrates the generalized arrangement of grid blocks used in order to compare the Darcy and Forchheimer-based models to their analytical solutions for radial flow to a well. In each case, the center matrix node was assigned a discharge value to simulate fluid removal from a porous matrix. A radial distance algorithm was employed to find the most circular arrangement of constant-head boundary grid blocks possible for a given matrix. The calibration simulations were run with a varying number of grid blocks ranging from a 20 x 20 matrix (corresponding to a 10 m Δx) to an 80 x 80 matrix (corresponding to a 2.5 m Δx).

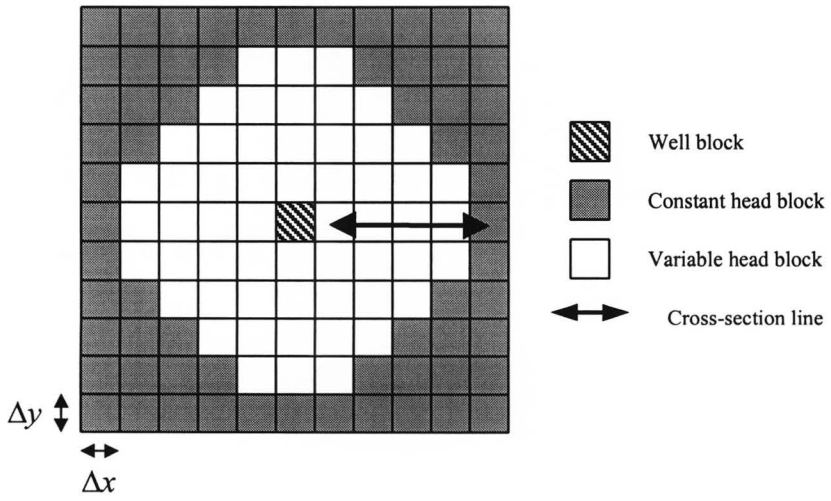


Figure 4.10: Generalized model layout for the purpose of calibration.

Figures 4.11 through 4.13 show the calibration results for three sets of simulations. These figures compare the values of head calculated for a radial cross-section through the model matrix. For each simulation, four curves were plotted: Darcy and Forchheimer-based analytical solutions and Darcy and Forchheimer-based numerical solution curves. Because both the Darcy and Forchheimer-based analytical solutions are invalid at the well node, the cross-section lengths have been restricted to the distance between the node adjacent to the well node and the constant-head boundary node.

4.4a Extremely non-linear flow calibration simulation

Figures 4.11a-c compare results obtained in a coarse matrix with an extremely high pumping rate. This simulation was intended to portray a model configuration that would represent an extreme upper limit to the non-linearity of flow encountered for this investigation. For these calibration simulations the values of the matrix linear hydraulic resistivity (a) and non-linear hydraulic resistivity (b) were chosen as 0.1 s/m and 10 s^2/m^2 respectively, simulating a matrix of 10cm cobbles. An extremely high well discharge rate of 100 m^3/s (26417 gal/s) was chosen. In all cases the aquifer thickness and initial head was set at 10 meters and the distance from the well to the constant head boundary was 100 meters.

The modeled curves shown in Figures 4.11a-c match the analytical curves reasonably well, however, there are some discrepancies. For the Forchheimer-based numerical model, the node adjacent to the well node consistently predicts less drawdown than the analytical solution. In addition, as the matrix is refined, the modeled curve shows slightly more drawdown been predicted by the analytical solution for nodes far from the well node.

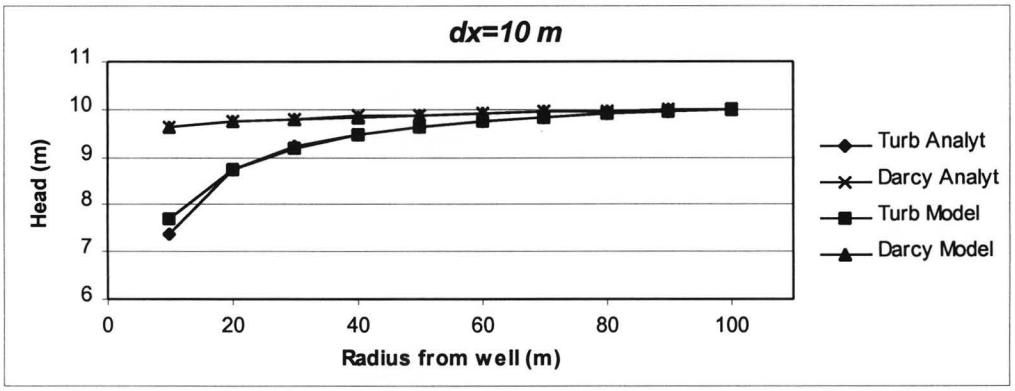


Figure 4.11a: Head distribution cross-section. Nodal spacing equals 10 m.

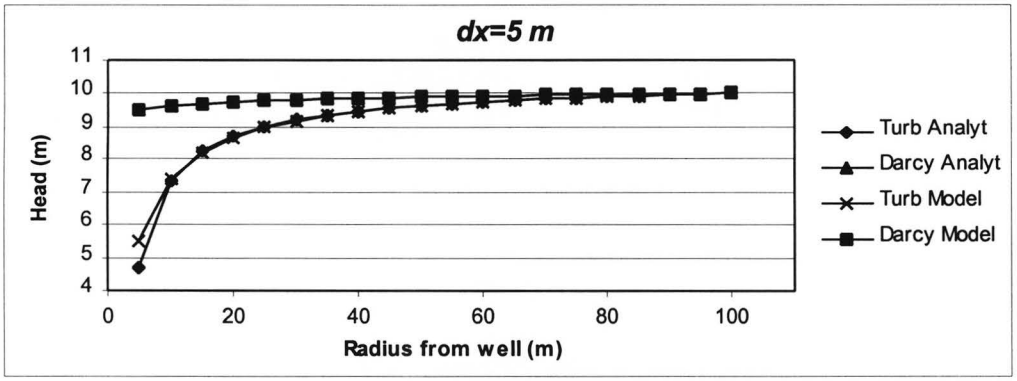


Figure 4.11b: Head distribution cross-section. Nodal spacing equals 5 m.

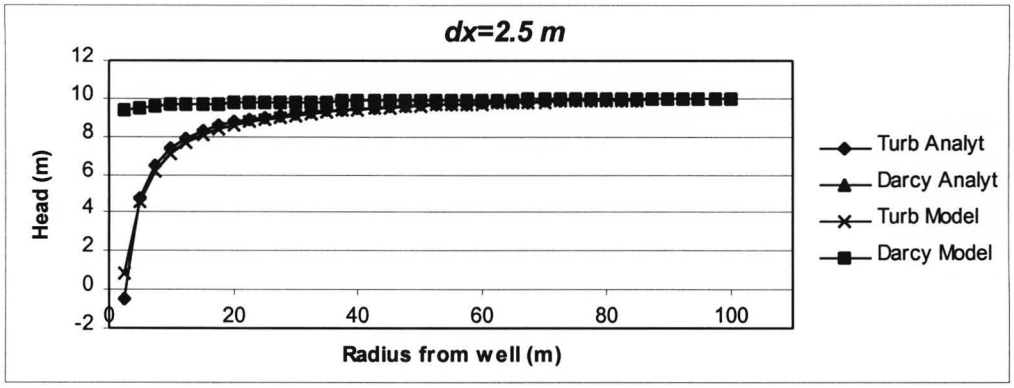


Figure 4.11c: Head distribution cross-section. Nodal spacing equals 2.5 m.

In order to quantify the error incurred using the Forchheimer-based numerical model, percent error was calculated using equation 4-6 for the sets of curves in Figures 4.11a-c. Table 4.1a shows percent error calculated for points of increasing radius from the well node at three different values of nodal spacing. In order to impose perspective upon the percent errors calculated, actual error in meters was reported for the same nodal points in Table 4.1b.

$$percent\ error = \left[\frac{\left|(analytical\ solution) - (numerical\ solution)\right|}{analytical\ drawdown}\right] * 100 \qquad (4\text{-}6)$$

	Radius from well				
	2.5m	***5m***	***10m***	***20m***	***50m***
Δx =***10m***	-	-	13.2	0.6	3.0
Δx =***5m***	-	15.1	1.0	4.3	4.0
Δx =***2.5m***	12.5	3.5	10.7	12.4	10.0

Table 4.1a: Percent error of Forchheimer-based numerical model

	Radius from well				
	2.5m	***5m***	***10m***	***20m***	***50m***
Δx =***10m***	-	-	0.35	0.008	0.011
Δx =***5m***	-	0.80	0.025	0.055	0.015
Δx =***2.5m***	1.32	0.19	0.284	0.158	0.037

Table 4.1b: Actual error (m) of Forchheimer-based numerical model

4.4b Moderately non-linear flow calibration simulation

Figures 4.12a-c compare results obtained in a moderately coarse matrix with a high pumping rate. This simulation was intended to portray a model configuration that would represent a moderately high degree of non-linear flow. For the set of calibration simulations shown below, the values of the matrix linear hydraulic resistivity (a) and non-linear hydraulic resistivity (b) were chosen as 10 s/m and 500 s^2/m^2 respectively, simulating a matrix of 5mm pebbles. A well discharge rate of 20 m^3/s (5283 gal/s) was chosen. This extremely high rate discharge was chosen to produce a good separation of curves for the purpose calibration. In all cases the aquifer thickness and initial

head was set at 10 meters and the distance from the well to the constant head boundary was 100 meters.

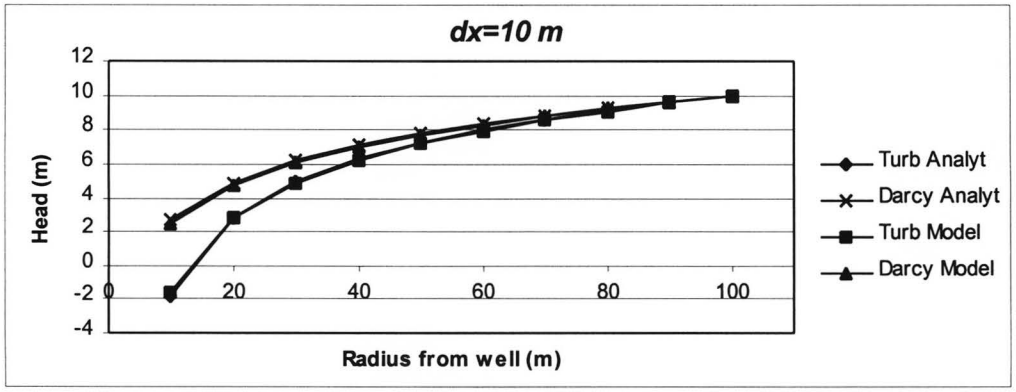


Figure 4.12a: Head distribution cross-section. Nodal spacing equals 10 m.

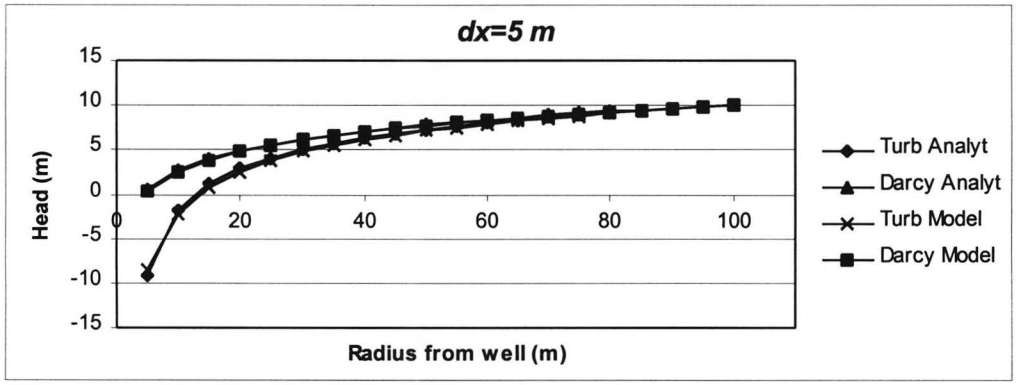


Figure 4.12b: Head distribution cross-section. Nodal spacing equals 5 m.

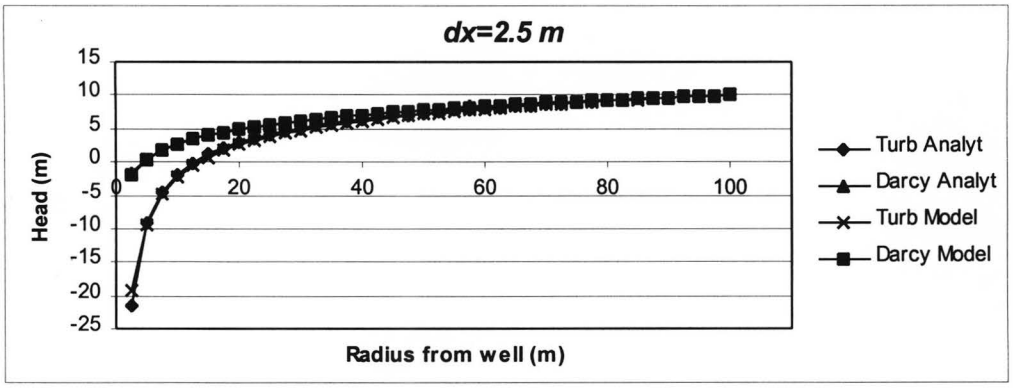


Figure 4.12c: Head distribution cross-section. Nodal spacing equals 2.5 m.

As with the simulations illustrated in Figures 4.11a-c, percent error was calculated for selected points along the cross-section line using equation 4-6 and listed in Table 4.2a. The actual error in meters is reported in Table 4.2b.

	Radius from well				
	2.5m	***5m***	***10m***	***20m***	***50m***
Δx ***=10m***	-	-	2.7	1.1	3.2
Δx ***=5m***	-	2.9	2.6	3.9	5.7
Δx ***=2.5m***	7.0	0.9	3.1	4.5	5.9

Table 4.2a: Percent error of Forchheimer-based numerical model

	Radius from well				
	2.5m	***5m***	***10m***	***20m***	***50m***
Δx ***=10m***	-	-	0.32	0.08	0.09
Δx ***=5m***	-	0.55	0.31	0.28	0.16
Δx ***=2.5m***	2.21	0.18	0.37	0.32	0.16

Table 4.2b: Actual error (m) of Forchheimer-based numerical model

The modeled curves shown in Figures 4.12a-c match the analytical curves more accurately than the modeled curves shown for the simulations represented by Figures 4.11a-c, showing less than a 10 percent error for all nodes on the solution surface. The character of deviation is the same as for the highly non-linear simulation: the Forchheimer-based numerical model predicts less drawdown for nodes adjacent to the well node, and slightly more drawdown for most other nodes.

4.4c Near-linear flow calibration simulation

Figures 4.13a-c compare results obtained in a relatively fine matrix with a moderate pumping rate. This simulation was intended to portray a model configuration that would represent a low degree of non-linear flow. For the set of calibration simulations shown below the values of the matrix linear hydraulic resistivity (a) and non-linear hydraulic resistivity (b) were chosen as 5000 s/m and 1e5 s^2/m^2 respectively, simulating a matrix with an average grainsize of 0.1 mm. A well discharge rate of 0.01 m^3/s (2.64 gal/s) was chosen to produce a small degree of separation between the Darcy and Forchheimer-based solutions. In all cases the aquifer thickness and initial head was set at 10 meters and the distance from the well to the constant head boundary was 100 meters.

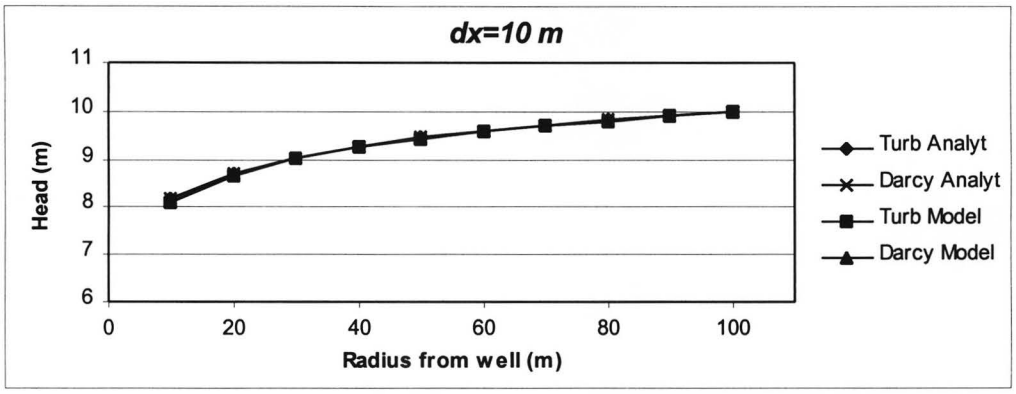


Figure 4.13a: Head distribution cross-section. Nodal spacing equals 10 m.

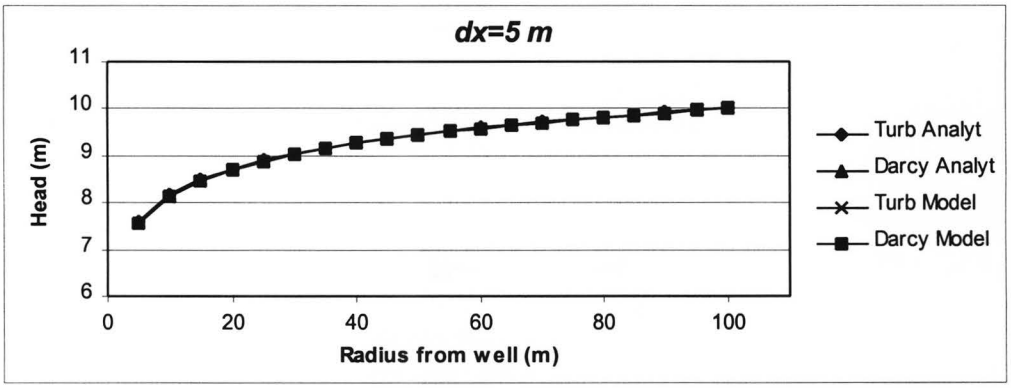


Figure 4.13b: Head distribution cross-section. Nodal spacing equals 5 m.

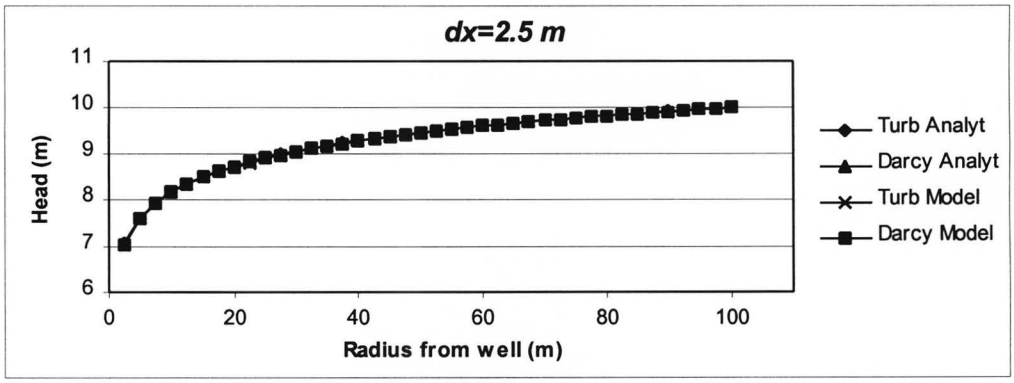


Figure 4.13c: Head distribution cross-section. Nodal spacing equals 2.5 m.

Again, percent error of the Forchheimer-based numerical solution was calculated in accordance with equation 4-6 and reported in Table 4.3a, while actual error in meters for the same points has been reported in Table 4.3b.

	Radius from well				
	2.5m	***5m***	***10m***	***20m***	***50m***
Δx ***=10m***	-	-	2.9	3.0	3.5
Δx ***=5m***	-	2.0	1.7	1.2	2.1
Δx ***=2.5m***	1.4	1.1	0.5	0.5	1.1

Table 4.3a: Percent error of Forchheimer-based numerical model

	Radius from well				
	2.5m	***5m***	***10m***	***20m***	***50m***
Δx ***=10m***	-	-	0.05	0.04	0.02
Δx ***=5m***	-	0.05	0.03	0.02	0.01
Δx ***=2.5m***	0.04	0.03	0.01	0.01	0.01

Table 4.3b: Actual error (m) of Forchheimer-based numerical model

The degree of agreement between the Forchheimer-based numerical model and the analytical solution is very good for the near-linear flow simulation described above. A percent error of less than 1.5 percent was reported for all points of the solution surface calculated using the matrix with a nodal spacing of 2.5 meters. Actual error with this degree of matrix refinement was less than 5 cm throughout the matrix. This is an identical degree of discrepancy between the Darcy-based numerical model and the

analytical solution. Tables 4.4a and 4.4b show the percent error and actual error incurred using Darcy-based numerical simulations.

	Radius from well				
	2.5m	***5m***	***10m***	***20m***	***50m***
Δx =***10m***	-	-	2.9	3.0	3.5
Δx =***5m***	-	2.0	1.7	1.2	2.1
Δx =***2.5m***	1.4	1.1	0.5	0.5	1.1

Table 4.4a: Percent error of Darcy-based numerical model

	Radius from well				
	2.5m	***5m***	***10m***	***20m***	***50m***
Δx =***10m***	-	-	0.05	0.04	0.02
Δx =***5m***	-	0.05	0.03	0.02	0.01
Δx =***2.5m***	0.04	0.03	0.01	0.01	0.01

Table 4.4b: Actual error (m) of Darcy-based numerical model

4.5 CALIBRATION CONCLUSIONS

The results of the calibration simulations show that the Darcy and Forchheimer-based numerical models predict the amount of drawdown produced with various combinations of parameters to a reasonable degree of accuracy. For non-linear flows, there will be less than a 15 percent error in accuracy for all points in the model matrix. Because of the amount of separation between the Darcy and Forchheimer-based solution surfaces at this

degree of non-linearity, a reliable estimate of the error incurred through the use of Darcian models can still be made. For moderately non-linear flow rates, the amount of separation between the Darcy-based and Forchheimer-based solutions surfaces is generally less, however, the accuracy of the models is increased, resulting in reasonably accurate predictions of Darcian error. When linear or near-linear flow regimes are modeled, the error incurred using numerical methods is reduced to less than 1.5 percent for both Darcy and Forchheimer-based models. This also allows for the accurate prediction of the error (or lack thereof) produced when the validity Darcy's law is assumed.

Chapter 5. Transient, Heterogeneous Flow Model

Heterogeneities can have a significant impact on subsurface flow patterns. Acting as conduits or boundaries, areas of greater or lesser hydraulic conductivity may dictate the direction and velocity of flow. With this two-dimensional, heterogeneous model, the impact of areal variations in aquifer conductivity on fluid drawdown caused by a single extraction well will be explored.

5.1 Model Construction and Configuration

The model was initially constructed using both the FORTRAN 77 and Turbo PASCAL programming codes. A comparison of simulation run-time showed that the FORTRAN 77 code produced simulations that ran almost 100 times faster than the corresponding PASCAL models. Consequently, all modeling was completed the FORTRAN 77 code. The full text of the computer program used in these simulations is presented in Appendices 3 and 4.

As implemented, the model consists of a square numerical matrix of varying dimensions. Most of the simulations were run using a matrix consisting of 80 nodes x 80 nodes (6400 nodes total). All the simulations listed in his work were run on an IBM PC clone with 96 Mb of memory and

an AMD K2-266Mhz central processing unit. At each of these nodes, numerical arrays were constructed that specified or recorded the hydraulic resistivity, non-linear "b" coefficient, average grain size, Darcy-based and Forchheimer-based head, Darcian and non-linear specific discharge in the x and y directions, Reynolds number, discharge as well as the apparent hydraulic resistivity in both the x and y direction. A perimeter of "dummy" nodes was also incorporated into the numerical array to facilitate the simulation of constant-head and no-flow boundary conditions when specified. The nodal spacing could be varied from simulation to simulation, but was always constant in each individual run. The generalized distribution of node types is illustrated in Figure 4.1.

To simulate the heterogeneous distribution of rock types within an aquifer the linear and nonlinear coefficients of hydraulic resistivity (a, b) were varied on a node by node basis. The model matrix nodes were assigned values for these coefficients during the model initialization procedure, and were not changed during solution iteration.

A non-zero term in the discharge array simulated a fully-penetrating well at the center of the matrix in all simulations. The thickness and initial head of the modeled aquifer were always constant (specified during the initialization procedures of each run). Convergence tolerance was somewhat

varied, but was always between 0.5 mm and 0.05 mm, generally less than well-test measurement errors.

Following the model's convergence within the specified tolerance, the specific discharge (q) was calculated (using equations 5-1 or 5-2 where appropriate) at each node from the estimated head values.

$$q_x = \frac{-a + \sqrt{a^2 - 4b\,dh/dx}}{2b} \qquad (5\text{-}1)$$ for hydraulic gradients less than zero.

$$q_x = \frac{a - \sqrt{a^2 + 4b\,dh/dx}}{2b} \qquad (5\text{-}2)$$ for hydraulic gradients greater than zero.

Where the value of dh/dx is determined for each node using the method described in Chapter 3. Using these values, the magnitude of the velocity vectors at each node was found, and from this, the Reynolds number associated with each model node was calculated using equation 5-3.

$$R_e = \frac{\rho q d}{\mu} \qquad (5\text{-}3)$$

Where R_e is the Reynolds number [-]

ρ is the density water [ML^{-3}]

q is the magnitude of the specific discharge [Lt^{-1}]

d is the average grain diameter [L]

μ is the viscosity of water [$Mt^{-1}L^{-1}$]

For all calculations, the density and viscosity of water was specified as 998 kg/m3 and 0.001 kg/m/s respectively.

The nodal Reynolds number values were calculated and recorded in order to quantify (in a general sense) the magnitude of non-linear flow behavior present in the matrix during a given simulation. Although the applicability of the Reynolds number to flows in a porous matrix is disputed (Lage, 1998), it's widespread use as an indicator of non-Darcy flow supplies a reference for the evaluation of these simulations.

5.2 Example Simulations

In order to quantify the error that might result from the application of Darcy's law in situations where it might not apply, six model configurations were explored (Tables 5.1a and 5.1b).

Example #	*Description*
Example 1	Flow to well in a matrix with a simulated average grainsize of 0.1 mm with a distribution of small, highly conductive heterogeneities.
Example 2	Flow to well in a matrix with a simulated average grainsize of 0.1 mm with a distribution of small, relatively impermeable heterogeneities.

Table 5.1a: Description of example simulations.

Example #	*Description*
Example 3	Flow to well that is completed in a band of aquifer material with a simulated average grainsize of 1 mm enclosed in a matrix with an average grainsize of 0.1 mm. The width of the band was reduced in successive simulations to ascertain the effects of increased flow channeling.
Example 4	Flow to a well in a matrix with a simulated average grainsize of 0.1 mm where the distance from the well to a large, highly permeable section of aquifer is shortened in successive simulations.
Example 5	Flow to a well in matrix with a simulated average grainsize of 0.1 mm with three highly conductive linear heterogeneities dispersed within it.
Example 6	Flow to a well matrix with a simulated average grainsize of 0.1 mm with three relatively impermeable linear heterogeneities dispersed within it.

Table 5.1b: Description of example simulations.

For all the simulations described in this chapter, a homogeneous matrix is intruded by various configurations of heterogeneities. For each model configuration, a steady-state, homogenous simulation using the resistivity parameters assigned to the matrix was first run to provide a baseline for comparison. Next, steady-state simulations were conducted that incorporated the various configurations of heterogeneities to ascertain the maximum amount of drawdown produced and the difference between the solutions obtained using the Darcy-based and Forchheimer-based models. Finally, selected transient simulations were run to determine the time dependence of the solutions. For the sake of brevity, plots of the Forchheimer-based and Darcy-based potentiometric surfaces were not included in the example illustrations below. Rather, the maximum Darcian drawdown value is included in the captions for the illustrations of the drawdown difference surfaces.

It should be noted that in all model runs the simulated well is located at the center of the matrix (41,41), and is identified by a small open cross at the center of all contour plots. Small crosses at the perimeter of the contour plots represent a radially distributed constant head boundary imposed upon the model in all simulations. All models were run using an 80 x 80 matrix of nodes utilizing 2.5 m nodal spacing. This configuration results in a 100 meter separation between the well and the radial constant-head boundary. Heterogeneities (areas of different hydraulic resistivity parameters than the

matrix) are represented by small open gray boxes in all contour plots. The parameters assigned to the matrix nodes for each model configuration are as listed in Tables 5.2a and 5.2b.

	Example 1	Example 2	Example 3
Matrix grainsize (d_m)	0.0001	0.001	0.0001
Matrix linear resistivity coefficient (a_m)	5352	137	5352
Matrix non-linear resistivity coefficient (b_m)	109269	4653	109269
Heterogeneity grainsize (d_h)	0.01	0.00001	0.001
Heterogeneity linear resistivity coefficient (a_h)	3.5	209039	137
Heterogeneity non-linear resistivity coefficient (b_h)	198	2565658	4653
Well discharge (Q)	0.004	0.50	0.1
Initial head	10.0	10.0	10.0
Model Dimensions	80 x 80	80 x 80	80 x 80
Nodal spacing (Δx)	2.5	2.5	2.5
Water density (ρ)	998.0	998.0	998.0
Gravitational acceleration (g)	9.81	9.81	9.81
Matrix porosity (n_m)	0.2	0.3	0.2
Heterogeneity porosity (n_h)	0.3	0.2	0.3
Matrix compressibility (α_m)	1.5×10^{-8}	1.5×10^{-8}	1.5×10^{-8}
Heterogeneity compressibility (α_h)	1.5×10^{-8}	1.5×10^{-8}	1.5×10^{-8}
Fluid compressibility (β)	4.8×10^{-10}	4.8×10^{-10}	4.8×10^{-10}

Table 5.2a: Set of parameters used in examples 1-3. All parameter values in SI units.

	Example 4	Example 5	Example 6
Matrix grainsize (d_m)	0.0001	0.0001	0.0001
Matrix linear resistivity coefficient (a_m)	5352	10.6	5352
Matrix non-linear resistivity coefficient (b_m)	109269	512	109269
Heterogeneity grainsize (d_h)	0.01	0.01	0.000001
Heterogeneity linear resistivity coefficient (a_h)	3.5	3.5	8164462
Heterogeneity non-linear resistivity coefficient (b_h)	198	198	60241932
Well discharge (Q)	0.004	0.004	0.004
Initial head	10.0	10.0	10.0
Model Dimensions	80 x 80	80 x 80	80 x 80
Nodal spacing (Δx)	2.5	2.5	2.5
Water density (ρ)	998.0	998.0	998.0
Gravitational acceleration (g)	9.81	9.81	9.81
Matrix porosity (n_m)	0.2	0.2	0.2
Heterogeneity porosity (n_h)	0.3	0.3	0.2
Matrix compressibility (α_m)	1.5×10^{-8}	1.5×10^{-8}	1.5×10^{-8}
Heterogeneity compressibility (α_h)	1.5×10^{-8}	1.5×10^{-8}	1.5×10^{-8}
Fluid compressibility (β)	4.8×10^{-10}	4.8×10^{-10}	4.8×10^{-10}

Table 5.2b: Set of parameters used in examples 4-6. All parameter values in SI units.

Example 1: Multiple, Small, Highly-Conductive Heterogeneities

The purpose of this simulation configuration is to determine the impact that a distribution of small, highly conductive rock types near a well produce on flow . Geologically, this type of situation might arise when pumping a karstic or anthropologically modified aquifer such as the Edwards aquifer of central Texas.

Figure 5.1 illustrates the difference between the drawdown predicted by the Forchheimer-based and Darcy-based models during a steady-state simulation in a homogeneous matrix. In this simulation, a single well extracts 0.004 m^3/s (approximately one gallon per second) from a matrix with an average grainsize of 0.0001 meters.

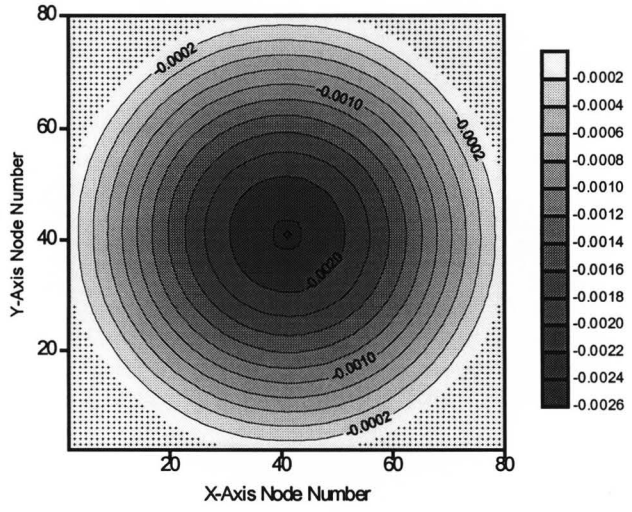


Figure 5.1: Forchheimer-based minus Darcy-based head(m) for a homogenous matrix with a grainsize of 0.0001 m. Simulated well is located at (41,41), small crosses represent the radially distributed constant head boundary. The maximum Darcy-based drawdown reported for this configuration was 1.9 meters.

As shown, there is very little difference (a maximum of approximately 2.5 mm) in the drawdown predicted by each of the models.

In Figure 5.2, the difference between the drawdown predicted by the Forchheimer-based model is significantly greater than that predicted by the Darcy-based model. The introduction of the highly conductive heterogeneities (simulated grainsize=10mm) caused in excess of a 20000 percent increase in the difference between the drawdown calculated. As expected, the greatest difference is found in the region surrounding the well where the drawdown is greatest.

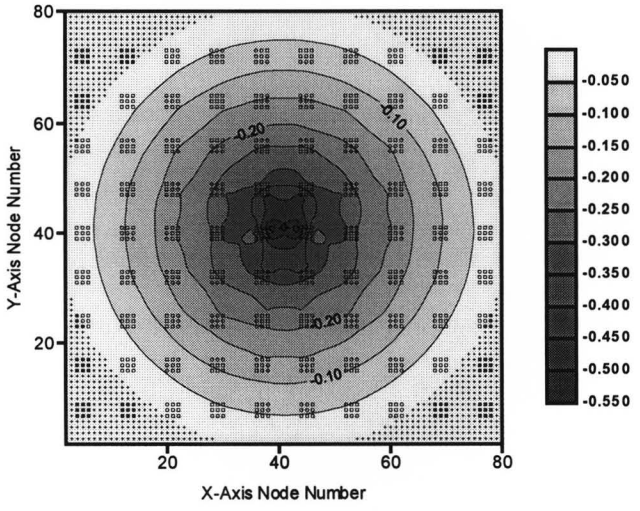


Figure 5.2: Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The maximum Darcy-based drawdown reported for this configuration was 1.9 meters.

Figures 5.3 to 5.6 display the correlation between elapsed simulation time and Forchheimer-based minus Darcy-based drawdown predictions. A rapid increase in the difference between the predicted potentiometric surfaces is seen in Figure 5.3. Within 30 seconds of the initiation of fluid extraction, the maximum difference between the two models has increased from 2.6 millimeters to 120 millimeters. However, Figures 5.4 through 5.6 show that this rate of deviation tapers off quickly. There is only a 90 mm increase in drawdown separation for the 1.5 minute interval between Figure 5.3 and Figure 5.4. The rate of increase of model separation continues to decline as a near steady-state condition is approached in Figures 5.5 and 5.6.

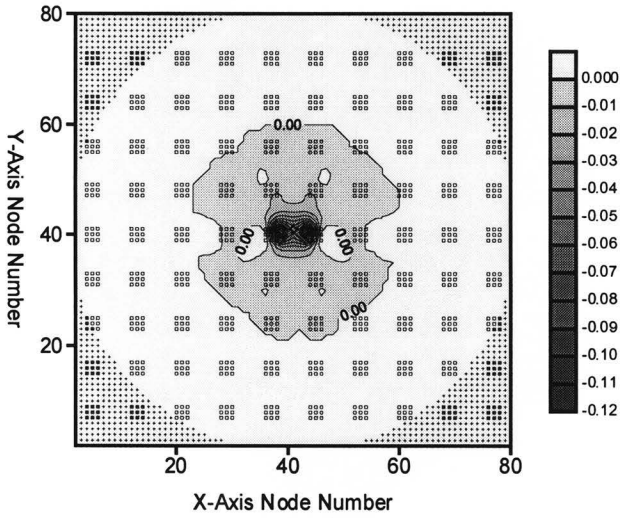


Figure 5.3: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 30 seconds. The maximum Darcy-based drawdown reported for this configuration was 0.96 meters.

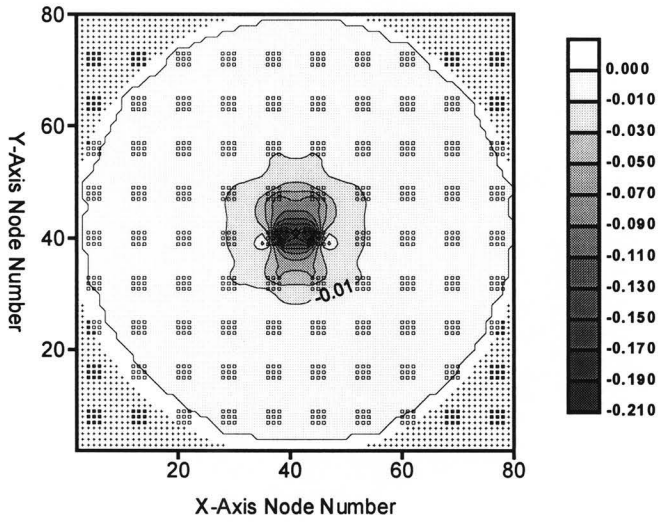


Figure 5.4: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 2 minutes. The maximum Darcy-based drawdown reported for this configuration was 1.1 meters.

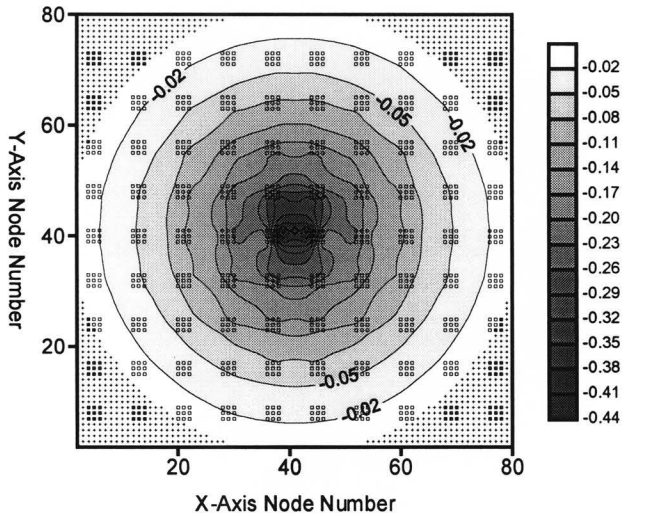


Figure 5.5: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 20 minutes. The maximum Darcy-based drawdown reported for this configuration was 1.3 meters.

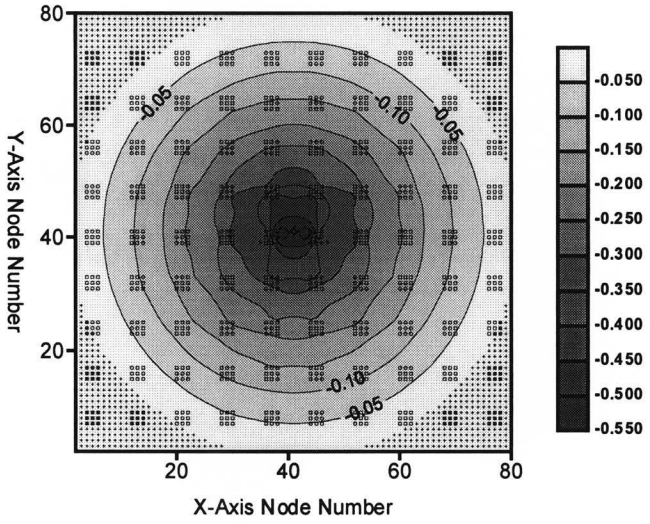


Figure 5.6: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 60 minutes. The maximum Darcy-based drawdown reported for this configuration was 1.9 meters.

The association between Reynolds number and drawdown difference is shown in Figure 5.7. Here we see that the highest Reynolds numbers are located within the highly conductive heterogeneities disbursed throughout the model matrix. It is informative to note that 100 percent of the models fine-grain matrix remained below a Reynolds number of unity throughout all the simulations conducted in example 1. Even at the well node, the Reynolds number calculated was well below the minimum threshold for turbulent flow.

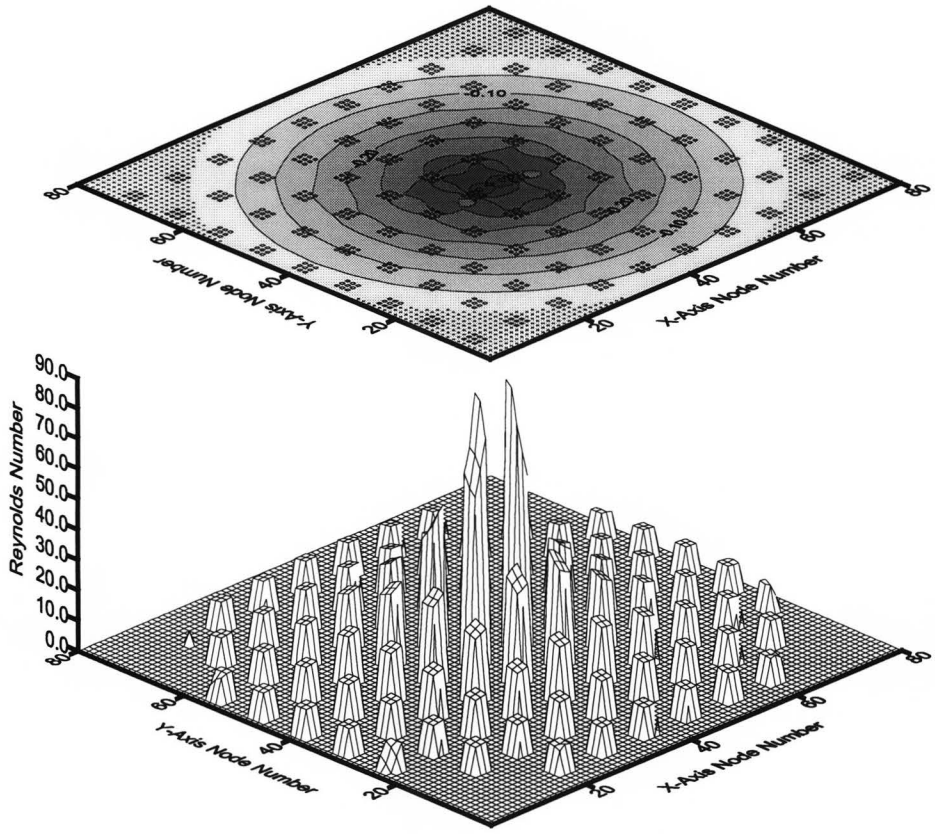


Figure 5.7: Association of Reynolds number and Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The matrix was assigned parameters that simulate a grainsize of 0.0001m. The small open boxes in the upper plot represent nodes with a grain size of 0.01 m.

Figures 5.8 through 5.11 illustrate the time dependency of the calculated Reynolds number. In these figures, Reynolds numbers show a dramatic increase in the heterogeneities adjacent to the well node in a small amount time, while outlying highly conductive zones required significantly more time to reach their peak. Since drawdown increases with time in these

simulations this effect was expected, however, the amount of time required to bring the model to a steady-state condition was not.

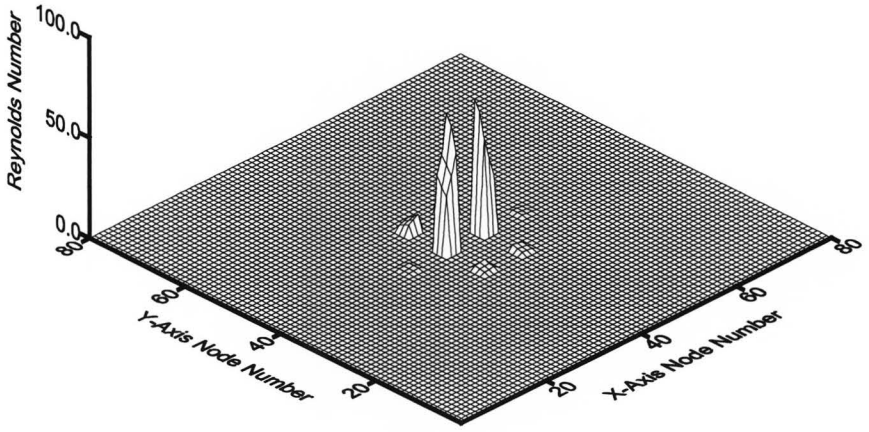


Figure 5.8: Reynolds number distribution for example 1 using transient model. Elapsed time equals 30 seconds.

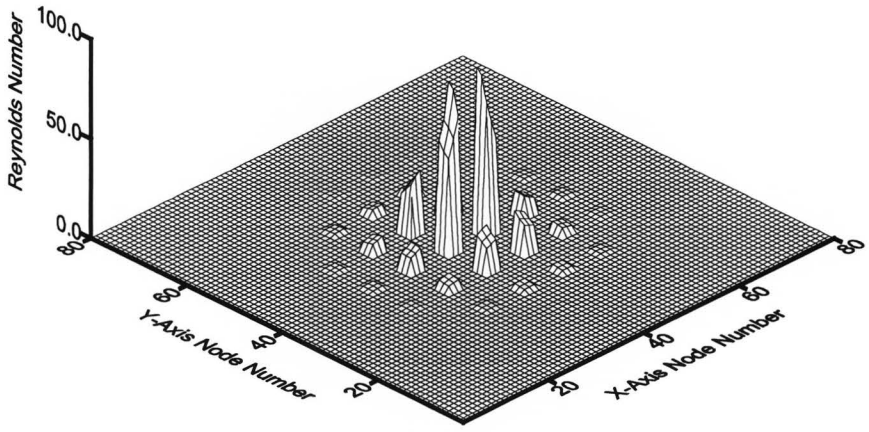


Figure 5.9: Reynolds number distribution for example 1 using transient model. Elapsed time equals 2 minutes.

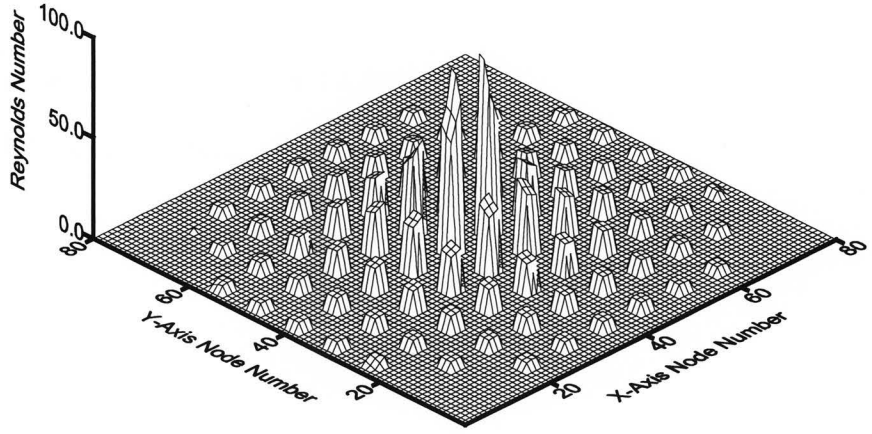


Figure 5.10: Reynolds number distribution for example 1 using transient model. Elapsed time equals 20 minutes.

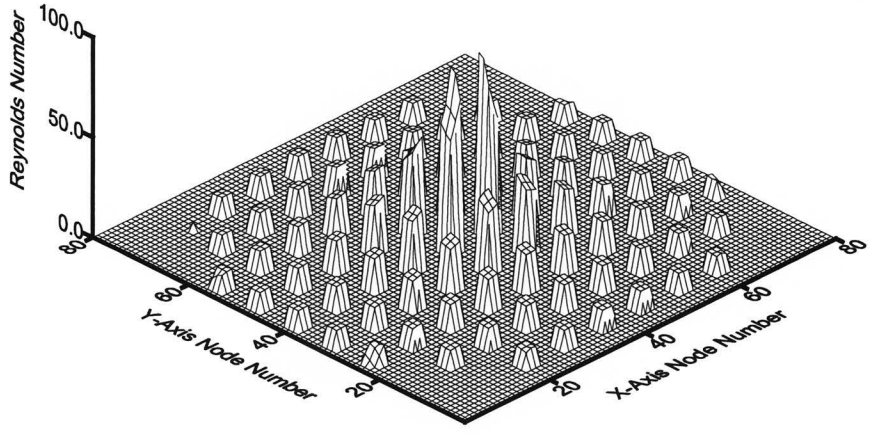


Figure 5.11: Reynolds number distribution for example 1 using transient model. Elapsed time equals 60 minutes.

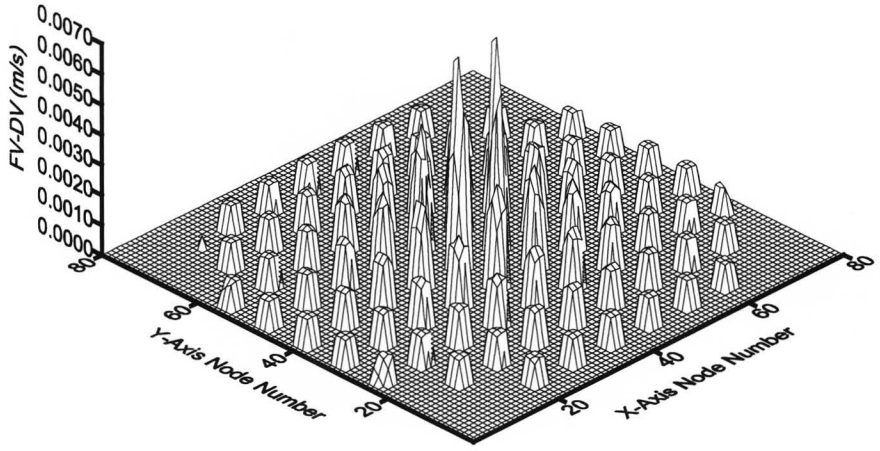


Figure 5.12: Forchheimer-based minus Darcy-based fluid velocity (m/s). Elapsed time equals 60 minutes.

Figure 5.12 shows the difference in average fluid velocity predicted by the two models. As expected, there is a substantial difference in fluid velocity in the heterogeneities surrounding the well node, while in outlying areas difference is reduced to one or two millimeters per second.

Example 2: Multiple, Small Impermeable Heterogeneities

In contrast to example 1, example 2 explores the difference in flow prediction between the Forchheimer-based and Darcy-based models in a coarse-grained matrix with low-permeability heterogeneities disbursed throughout.

As a baseline for comparison, the difference between the drawdown predicted by the Forchheimer-based and Darcy-based models during a steady-state simulation in a homogeneous matrix is reported in Figure 5.13. In this simulation, a single well extracts 0.5 m^3/s (approximately 132 gallons per second) from a matrix with an average grainsize of 0.001 meters. The significant drawdown difference of 0.38 meters reported for this simulation is due to increased non-Darcy behavior resulting from flow in a coarse-grained matrix with a relatively high pumping rate.

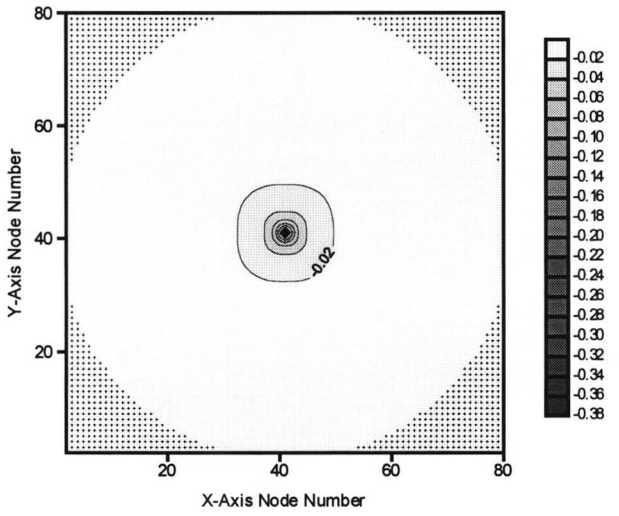


Figure 5.13: Homogenous matrix d=0.001 m Forchheimer-based minus Darcy-based head (m). The maximum Darcy-based drawdown reported for this configuration was 4.9 meters.

Figure 5.14 shows the steady-state difference in drawdown predicted when an array of small, relatively impermeable heterogeneities (average grainsize equals 0.01 mm) is introduced into the model matrix. The

maximum difference in drawdown calculated has risen from the 0.38 meter difference shown in Figure 5.13 to the 5.0 meter difference shown in Figure 5.14.

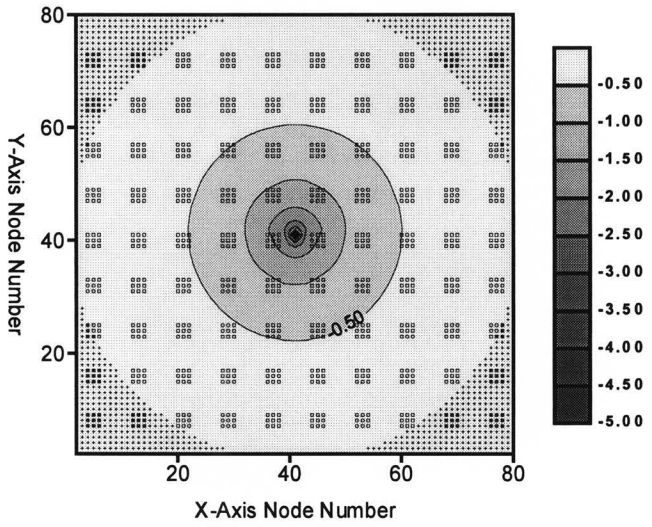


Figure 5.14: Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The matrix was assigned parameters that simulate a grainsize of 0.001m. The small open boxes represent nodes with a grain size of 0.00001 m. The maximum Darcy-based drawdown reported for this configuration was 4.9 meters.

Figures 5.15 through 5.18 display the correlation between elapsed simulation time and Forchheimer-based minus Darcy-based drawdown predictions. As these figures show, a significant portion of the total drawdown difference occurs at the well node within a few seconds, and then migrates outward radially as time elapses. The heterogeneities resist the initiation of drawdown differences between the two models for a few minutes, but eventually show the radial pattern of difference exhibited by the matrix.

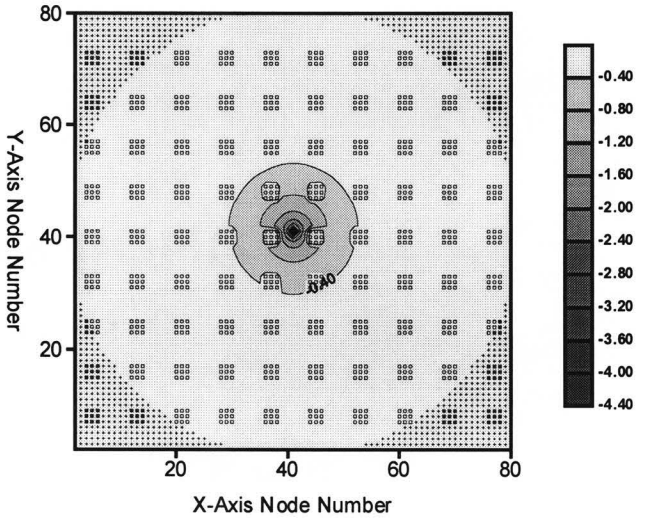


Figure 5.15: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 30 seconds. The maximum Darcy-based drawdown reported for this configuration was 4.3 meters.

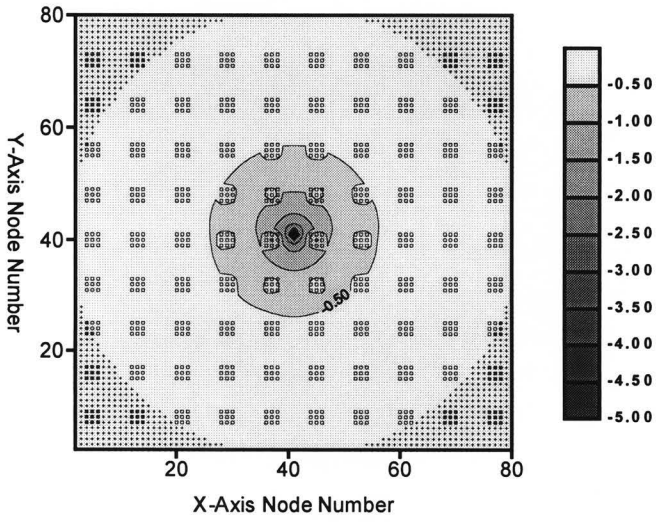


Figure 5.16: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 2 minutes. The maximum Darcy-based drawdown reported for this configuration was 4.6 meters.

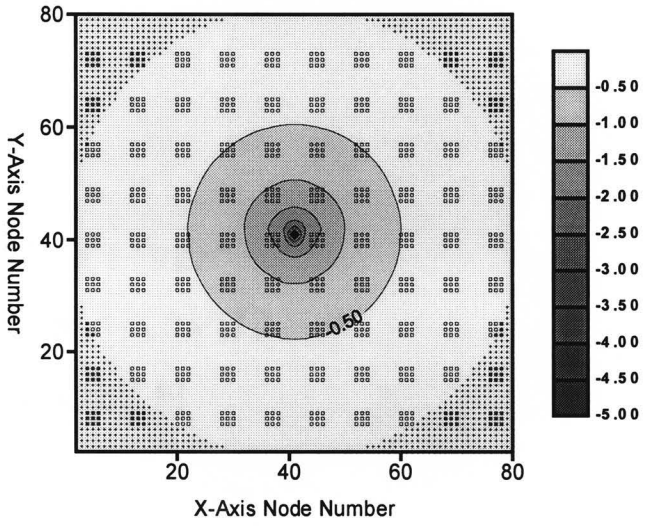


Figure 5.17: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 20 minutes. The maximum Darcy-based drawdown reported for this configuration was 4.9 meters.

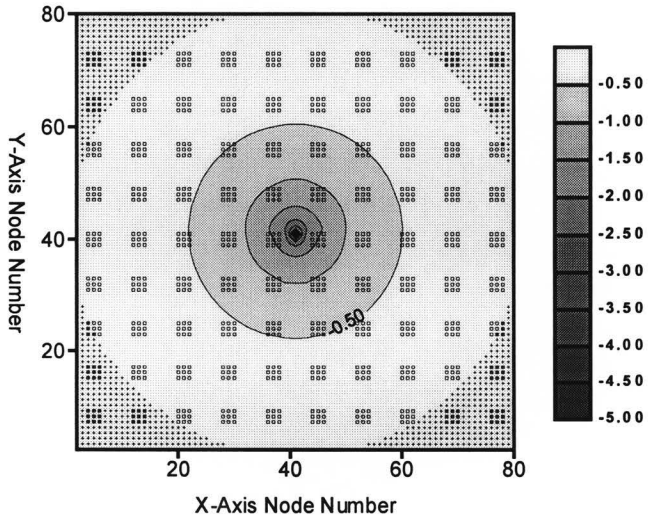


Figure 5.18: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 60 minutes. The maximum Darcy-based drawdown reported for this configuration was 4.9 meters.

Figures 5.19 through 5.22 illustrate the time dependency of the calculated Reynolds number for the model configuration described by example 2. A rapid increase in Reynolds number is associated with the area around the well. However, unlike example one, the increase is much more modest (a maximum of six compared to a maximum of 90) and is located in the matrix nodes instead of the heterogeneities. Initially, the largest increases are seen in the matrix nodes that bound the individual heterogeneities, but are transferred to the whole matrix within a matter of minutes.

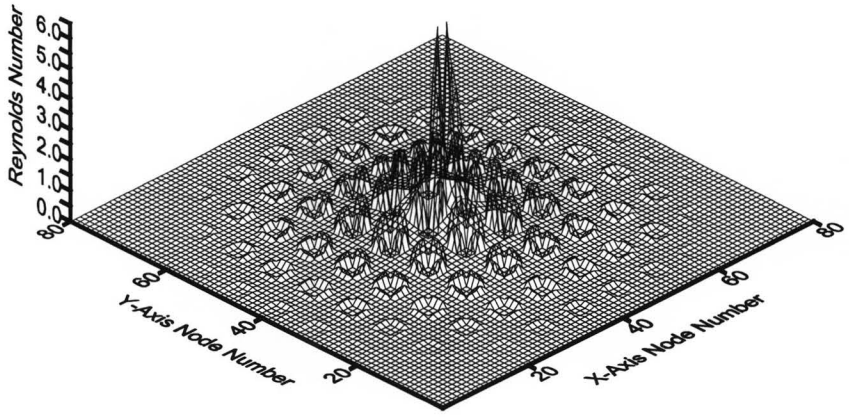


Figure 5.19: Reynolds number distribution using transient model. Elapsed time equals 30 seconds.

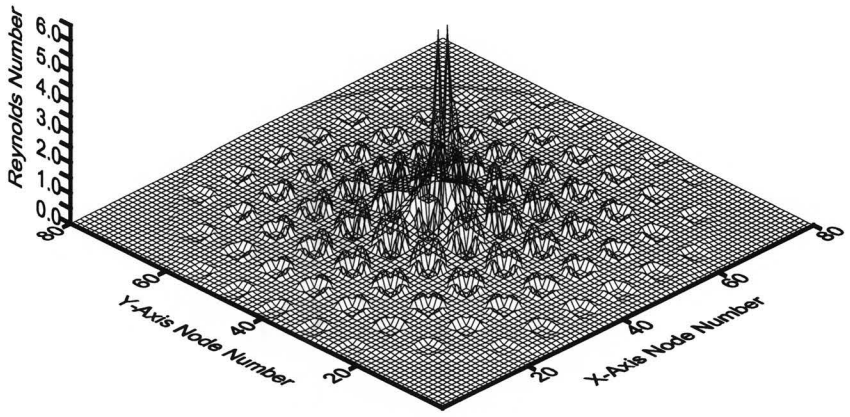


Figure 5.20: Reynolds number distribution using transient model. Elapsed time equals 2 minutes.

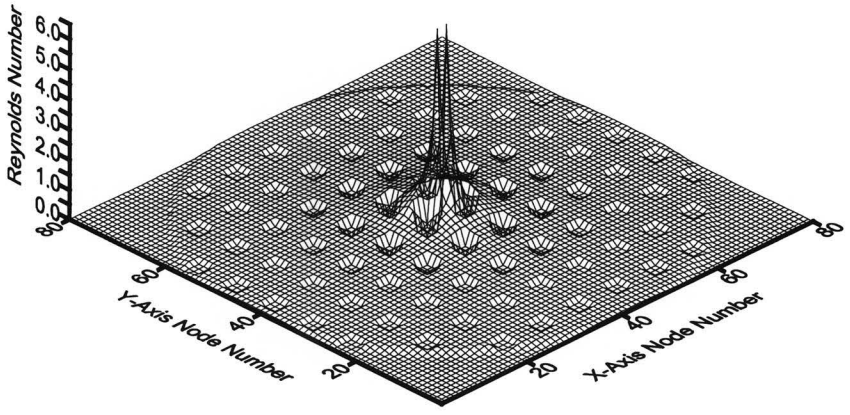


Figure 5.21: Reynolds number distribution using transient model. Elapsed time equals 20 minutes.

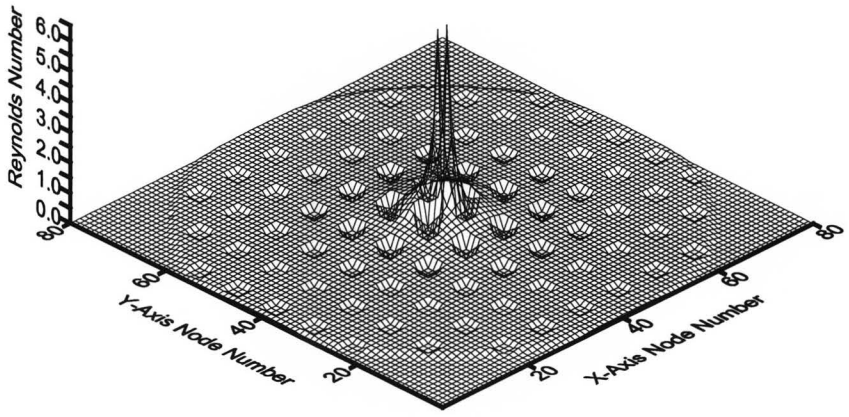


Figure 5.22: Reynolds number distribution using transient model. Elapsed time equals 60 minutes

Example 3: Single Narrowing Bar Heterogeneity

Example 3 explores the error that might be incurred using a Darcy-based numerical model in a situation where a large capacity well is completed in a narrow zone of relatively high conductivity aquifer material. In the following simulations, a discharge well is located at the center of a band of aquifer material with and average grainsize of 1 mm. This conductive section of aquifer is sandwiched between two larger sections with an average grainsize of 0.1 mm. In order to gain some understanding of how increased channeling of flow affects the difference between each model's solution surface, the highly conductive band is thinned in successive simulations.

Figure 5.23 illustrates the difference between the drawdown predicted by the Forchheimer-based and Darcy-based models during a steady-state simulation in a homogeneous matrix. In this simulation, a single well extracts 0.1 m^3/s (approximately 26 gallons per second) from a matrix with an average grainsize of 0.001 meters. As shown, there is a moderate difference (a maximum of approximately 38 cm) in the drawdown predicted by each of the models.

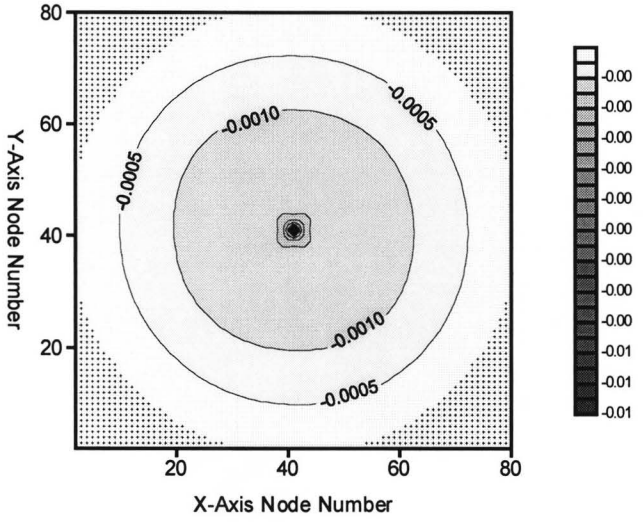


Figure 5.23: Forchheimer-based minus Darcy-based head (m) for a homogenous matrix with a grainsize of d=0.001 m. Simulated well is located at (41,41), small crosses represent the radially distributed constant head boundary. The maximum Darcy-based drawdown reported for this configuration was 1.2 meters.

In Figure 5.24, the difference between the drawdown predicted by the two models has increased significantly with the narrowing of the conductive

band to a width of 52.5 meters (21 nodes). This change produced in excess of a 40 percent increase in the difference between the drawdown calculated. Again, the greatest difference is found in the region surrounding the well where the drawdown is greatest.

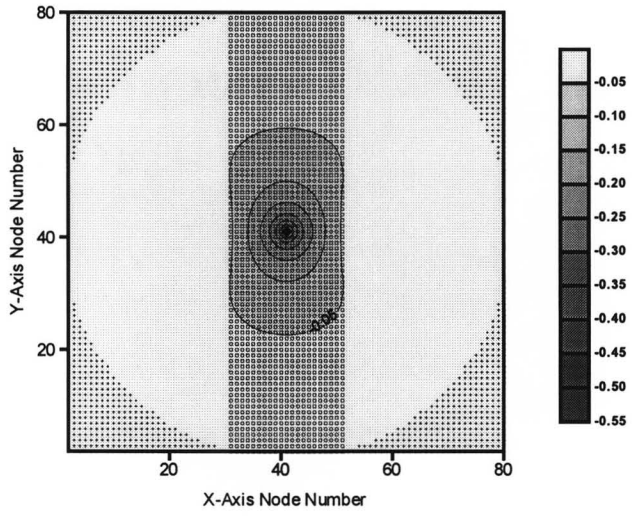


Figure 5.24: Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The maximum Darcy-based drawdown reported for this configuration was 1.2 meters.

In Figures 5.25 and 5.26, the width of the conductive zone is narrowed to 27.5 meters (11 nodes) and 17.5 meters (seven nodes) respectively. In each case, a moderate increase in the maximum difference is observed.

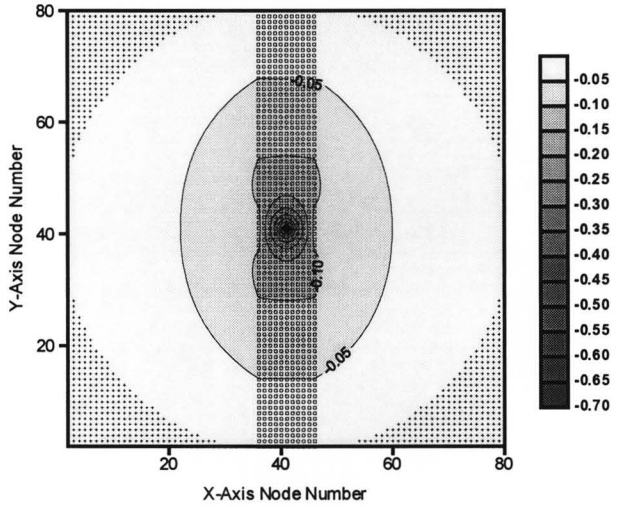


Figure 5.25: Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The maximum Darcy-based drawdown reported for this configuration was 1.2 meters.

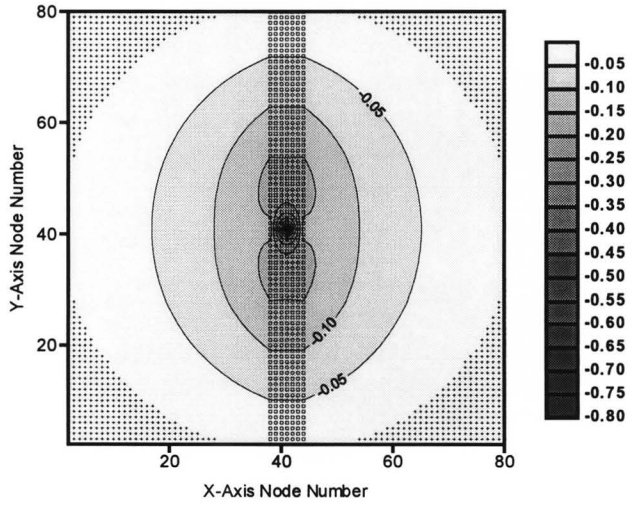


Figure 5.26: Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The maximum Darcy-based drawdown reported for this configuration was 1.2 meters.

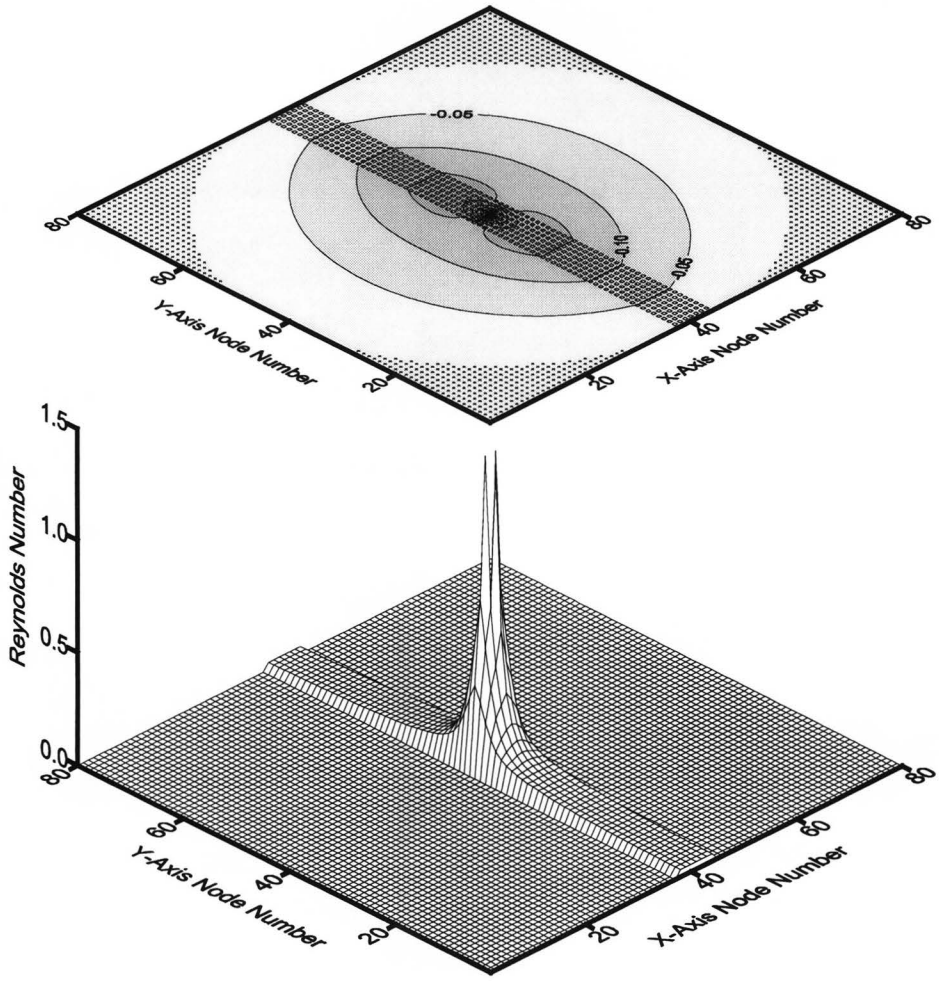


Figure 5.27: Association of Reynolds number and Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The gray bar in the upper plot represents an area of the aquifer with a grain size of 0.001 m, while remaining matrix has an average grainsize of 0.0001 m.

Figure 5.27 illustrates the relationship between the drawdown difference predicted by the two models and Reynolds number. The low

magnitude of Reynolds numbers calculated for this model configuration are consistent with the relatively small degree of surface separation shown in Figures 5.24 through 5.26.

Example 4: Single Wide Encroaching Bar Heterogeneity

In example 4, a large, highly conductive heterogeneity is placed successively closer to a pumping well in consecutive model runs. The purpose of these simulations is to study the effects that a highly conductive zone in the vicinity of a well might have on the accuracy of Darcy-based models. For comparison, the drawdown difference reported by each of the models for a homogenous matrix of 0.0001 meters and a well discharge of 0.004 meters per second is illustrated in Figure 5.1.

In Figures 5.29 through 5.31 the distance between the well located at (41,41) and a large conductive heterogeneity (average grainsize equals 1 mm) is successively shortened. As a result, separation between the Forchheimer-based and Darcy-based potentiometric surfaces increases. Figure 5.28 shows that when the conductive zone terminates 50 meters from the location of the well there is no appreciable increase in the drawdown error calculated using the Darcy-based model. However, when that distance is shortened to 25 meters (Figure 5.29) the difference between the solution surfaces increases from 2.6 mm to 19 mm. Separation between the curves increases dramatically

as the distance between the highly conductive zone and the well decreases. Figure 5.30 shows that when distance is reduced to 12.5 meters the error incurred using the Darcy-based model increases to 160 mm. When the distance is reduced to 7.5 meters in the simulation illustrated in Figure 5.31, the error is increased to 380 mm. In addition, these errors are compounded because the total drawdown predicted by the Darcy-based model decreases as the high conductivity heterogeneity encroaches upon the well location.

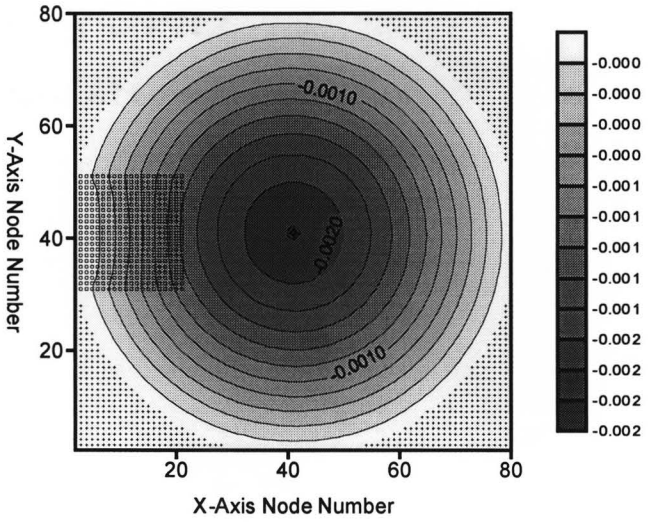


Figure 5.28: Steady-state, Forchheimer-based minus Darcy-based head (m). The gray bar represents an area of the aquifer with a grain size of 0.01 m that terminates at a distance of 50 meters (20 nodes) from the well node. The maximum Darcy-based drawdown reported for this configuration was 1.8 meters.

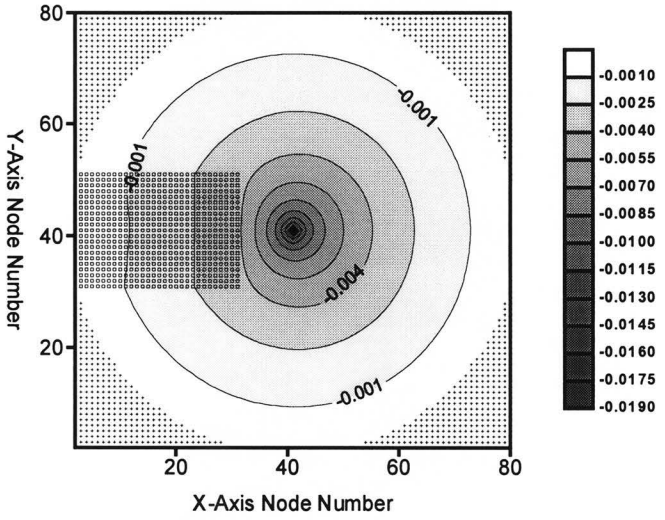


Figure 5.29: Steady-state, Forchheimer-based minus Darcy-based head (m). The gray bar represents an area of the aquifer with a grain size of 0.01 m that terminates at a distance of 25 meters (10 nodes) from the well node. The maximum Darcy-based drawdown reported for this configuration was 1.7 meters.

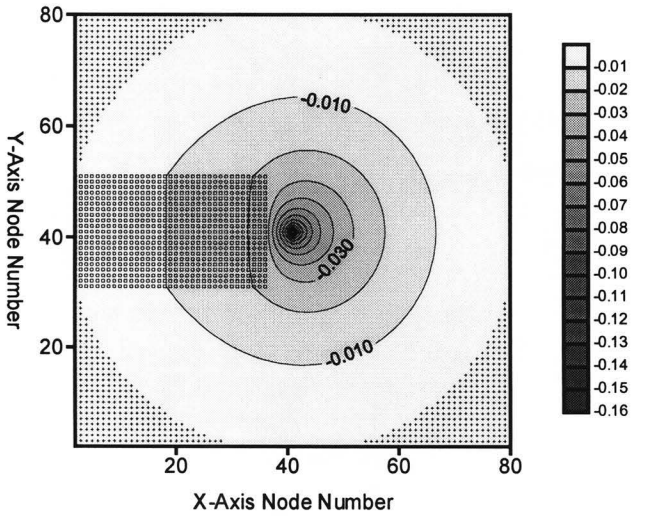


Figure 5.30: Steady-state, Forchheimer-based minus Darcy-based head (m). The gray bar represents an area of the aquifer with a grain size of 0.01 m that terminates at a distance of 12.5 meters (5 nodes) from the well node. The maximum Darcy-based drawdown reported for this configuration was 1.5 meters.

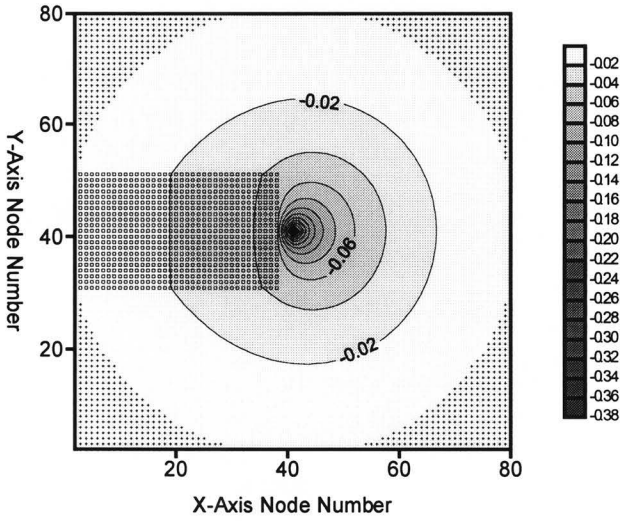


Figure 5.31: Steady-state, Forchheimer-based minus Darcy-based head (m). The gray bar represents an area of the aquifer with a grain size of 0.01 m that terminates at a distance of 7.5 meters (3 nodes) from the well node. The maximum Darcy-based drawdown for this configuration was 1.3 meters.

Figure 5.32 illustrates the relationship between Reynolds number and the deviation of the Darcy-based model's potentiometric surface and the surface calculated using the Forchheimer-based model. Here, the large spike in calculated Reynolds number clearly coincides with the portion of the

contour plot that reports the largest difference between the two types of model.

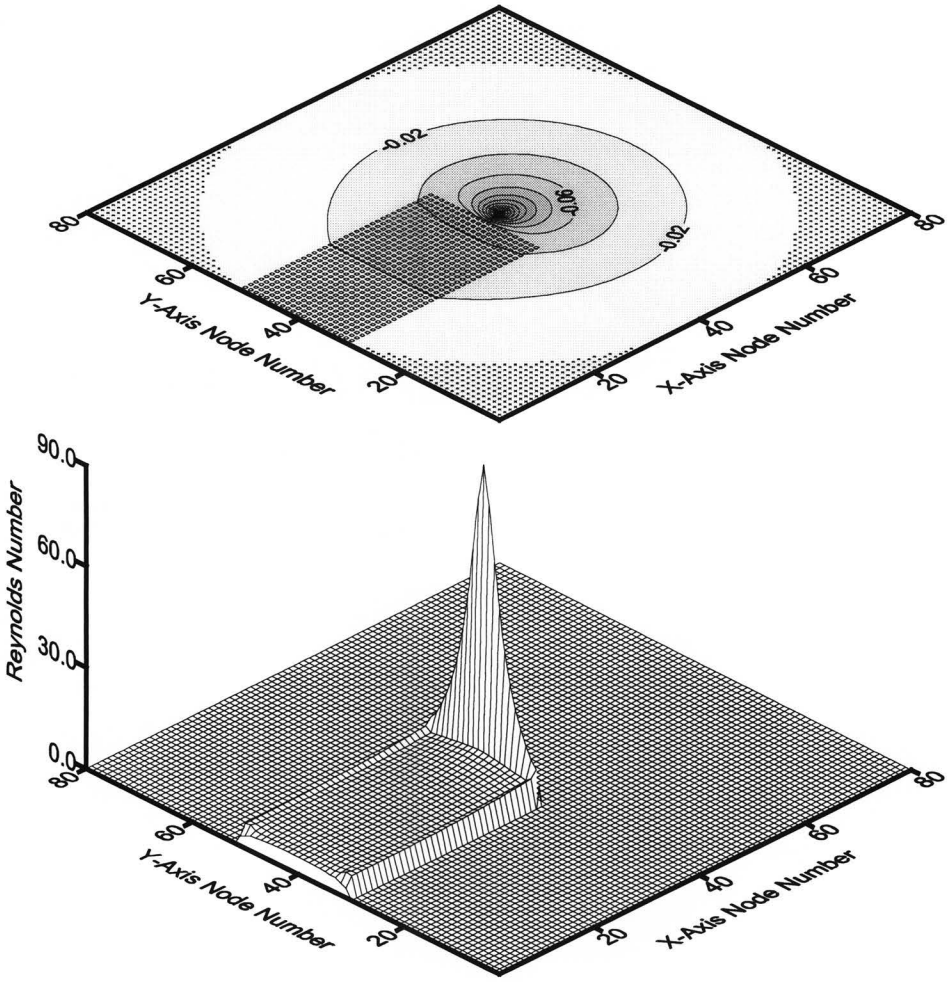


Figure 5.32: Association of Reynolds number and Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The gray bar in the upper plot represents an area of the aquifer with a grain size of 0.01 m that terminates at a distance of 7.5 meters (3 nodes) from the well position.

Example 5: Thin Channel Heterogeneities

Example 5 explores the error that might be incurred using a Darcy-based numerical model in a situation where a well located in a matrix that is divided by several highly conductive planar heterogeneities. Geologically, this situation might occur in an aquifer cut through by sets of fractures or dissolution zones. In this example, Forchheimer coefficients were selected that simulate a matrix with an average grainsize of 0.1 mm, while the planar heterogeneities are assigned and average grainsize of 10 mm. A discharge well located at the center of the matrix simulates an extraction of 0.004 m^3/s (approximately one gallon per second). For reference, the drawdown difference calculated using the two models for a homogeneous matrix of this grainsize is reported in Figure 5.1.

Figure 5.33 plots the solution difference surface that incorporates the conductive heterogeneities resulting from a steady-state simulation. When compared to the surface and maximum drawdown reported in Figure 5.1, it is seen that the drawdown difference has increased from 2.6 mm to 18 mm while the maximum Darcy-based drawdown has decreased from 1.9 meters to 1.5 meters.

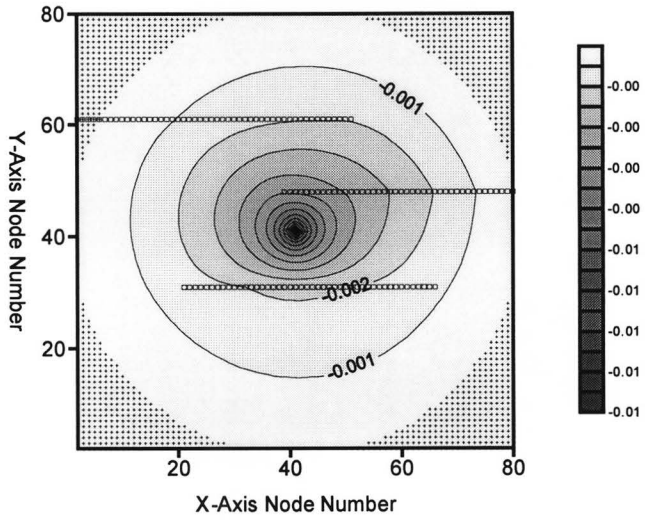


Figure 5.33: Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The matrix was assigned parameters that simulate a grainsize of 0.0001m. The small open boxes represent nodes with a grain size of 0.01 m. The maximum Darcy-based drawdown reported for this configuration was 1.5 meters.

The evolution through time of the Forchheimer-based minus Darcy-based solution surface is plotted in Figures 5.34 through 5.37. As these figures show, the geometry of the drawdown difference surface calculated his highly time dependent. In Figure 5.34, the difference surface is influenced by the proximity of the highly conductive channels, but the majority of surface separation is confined to a quasi-regular cone. For an elapsed simulation time of two minutes (Figure 5.35), the shape of the difference surface has changed significantly. Here, the surface separation around the well node is still symmetrical but areas of lesser separation have been distorted by flow through the nodes assigned lower resistivity.

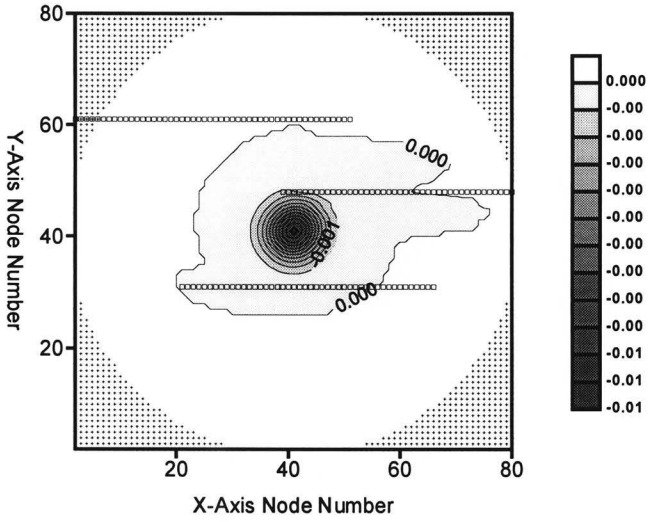


Figure 5.34: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 30 seconds. The maximum Darcy-based drawdown reported for this configuration was 1.0 meters.

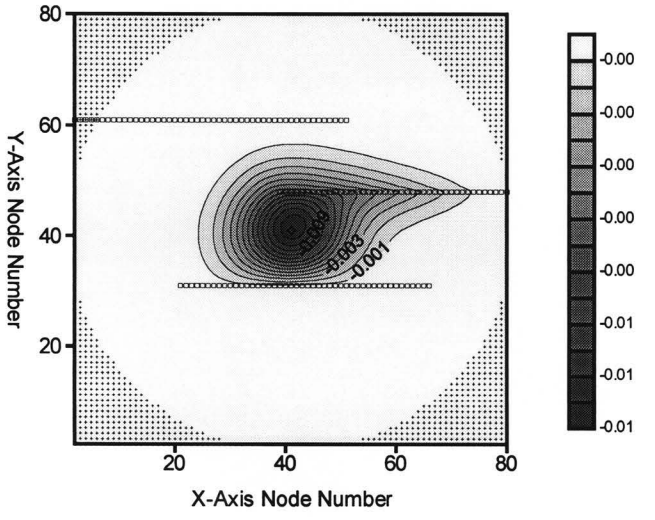


Figure 5.35: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 2 minutes. The maximum Darcy-based drawdown reported for this configuration was 1.2 meters.

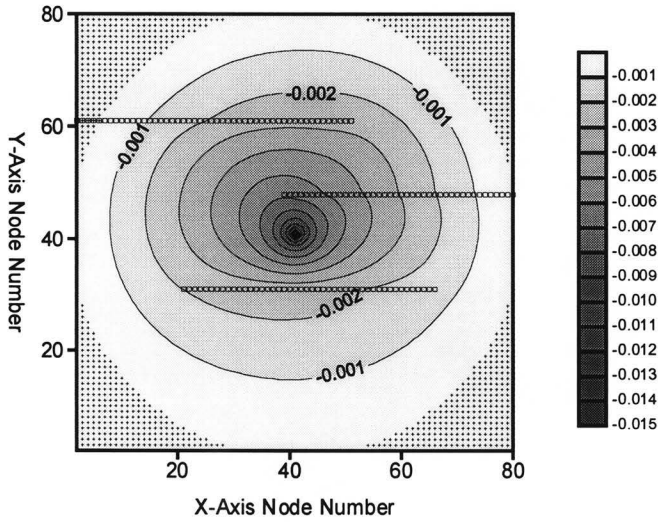


Figure 5.36: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 20 minutes. The maximum Darcy-based drawdown reported for this configuration was 1.4 meters.

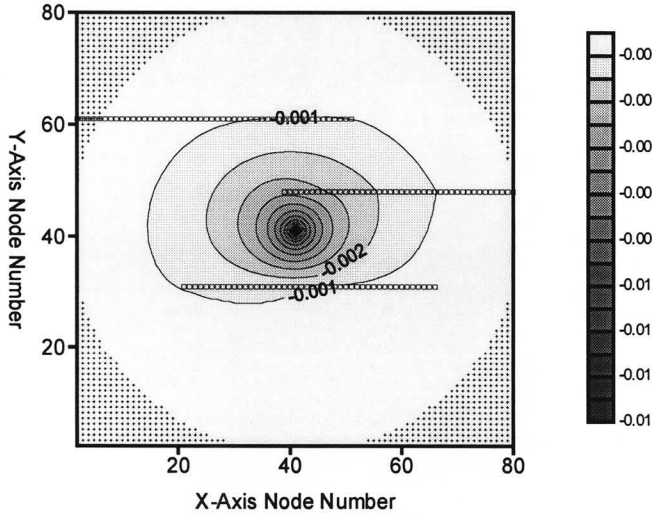


Figure 5.37: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 60 minutes. The maximum Darcy-based drawdown reported for this configuration was 1.5 meters.

The separation plots for later time periods depicted in Figures 5.36 and 5.37 show that, for the most part, radial symmetry is reestablished as steady-state conditions are approached.

Figure 5.38 illustrates the relationship between the drawdown difference predicted by the two models and Reynolds number. In this case, there is a more subtle connection between the zones reporting large Reynolds numbers and the distribution of drawdown difference. Although the shape of the difference surface is influenced by the placement of the conductive heterogeneities, the area of greatest difference does not coincide with high Reynolds number nodes.

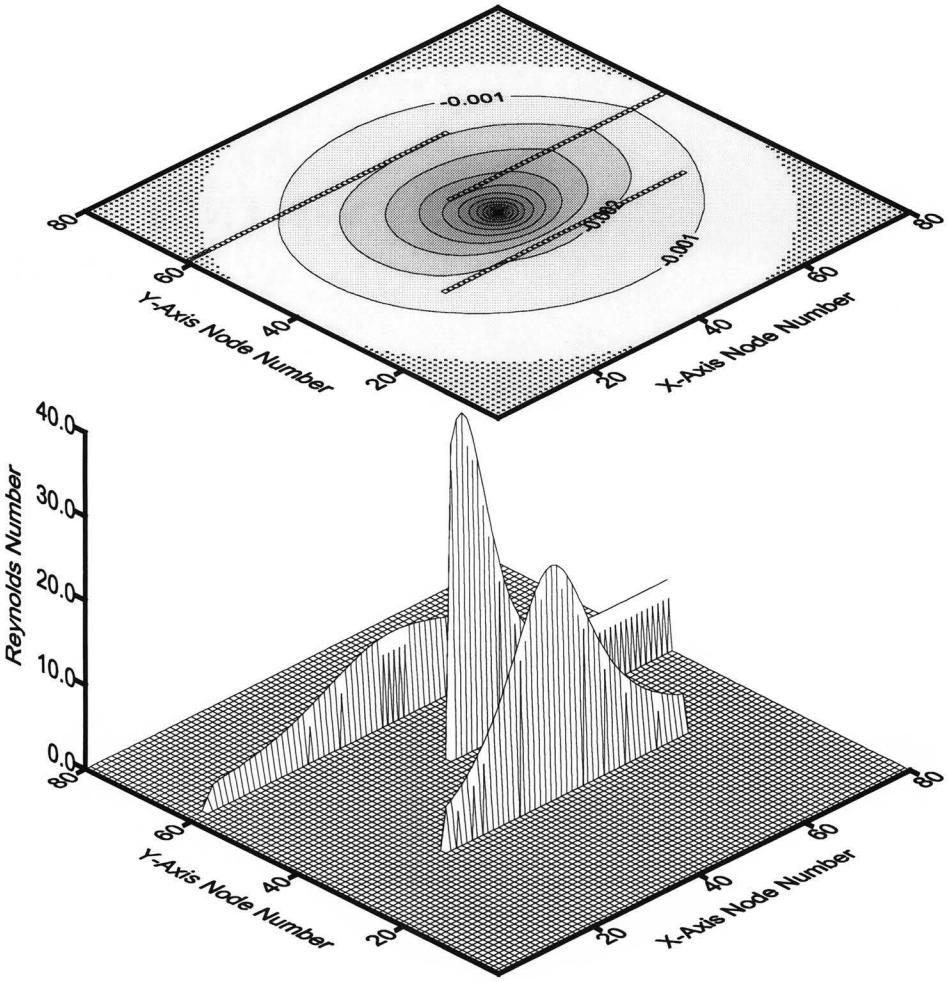


Figure 5.38: Association of Reynolds number and Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The matrix was assigned parameters that simulate a grainsize of 0.0001m. The gray lines parallel to the x-axis in the upper plot represent nodes with a grain size of 0.01 m.

Example 6: Thin Flow Boundaries

In order to examine a model configuration converse to that shown in example 5, example 6 incorporates planar, relatively impermeable heterogeneities that serve as barriers to flow. In this example, Forchheimer coefficients were selected that simulate a matrix with an average grainsize of 0.1 mm, while the planar heterogeneities are assigned and average grainsize of 0.001 mm. A discharge well located at the center of the matrix simulates an extraction of 0.004 m^3/s (approximately one gallon per second). For reference, the drawdown difference calculated using the two models for a homogeneous matrix of this grainsize is reported in Figure 5.1.

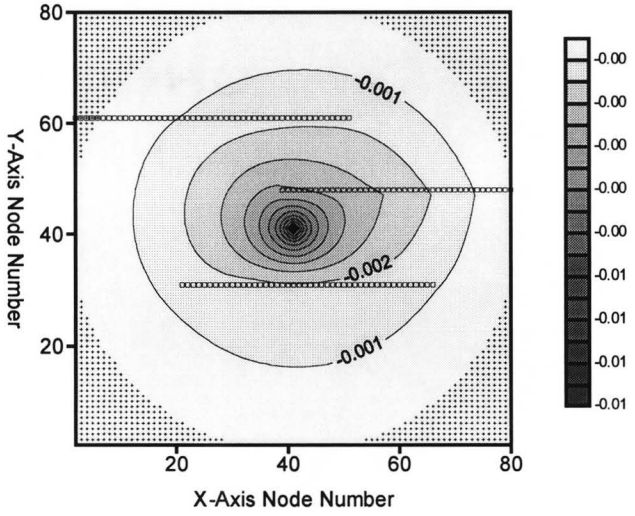


Figure 5.39: Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The maximum Darcy-based drawdown reported for this configuration was 1.5 meters.

Figure 5.39 shows the distribution of drawdown difference reported for a steady-state simulation that incorporates planar flow barriers. As illustrated, the error incurred using a Darcy-based model increased from 2.6 mm to 17 mm (550 percent) when the impermeable heterogeneities are incorporated. As with previous examples, these errors are compounded because the magnitude of maximum drawdown has been decreased.

The progression of model surface separation in time is depicted in Figures 5.40 through 5.43. As with the surfaces illustrated in the previous example, these figures show that the geometry of the difference surface is highly dependent upon the elapsed time since the initiation of pumping.

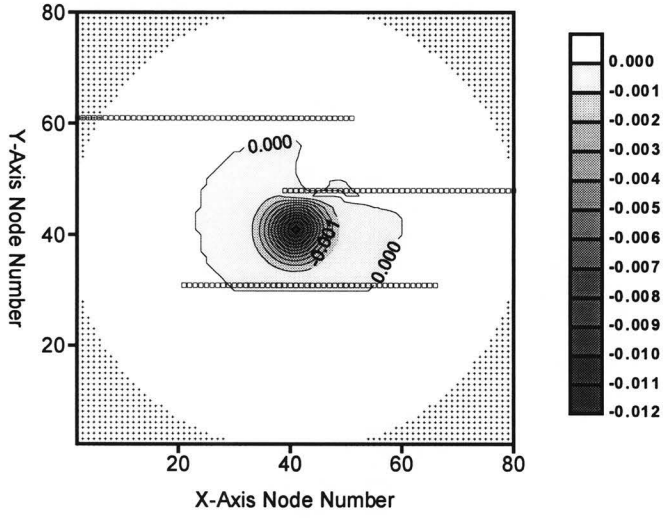


Figure 5.40: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 30 seconds. The maximum Darcy-based drawdown reported for this configuration was 1.0 meters.

Figures 5.40 through 5.43 show that quasi-radial symmetry of the difference surface is present at early times as well as when steady-state conditions are approached. At intermediate times, the difference surface is extended in the direction of orientation of the relatively impermeable heterogeneities. The magnitude of the difference between the Forchheimer-based and Darcy-based potentiometric surfaces is small and changes little over the duration of the simulation.

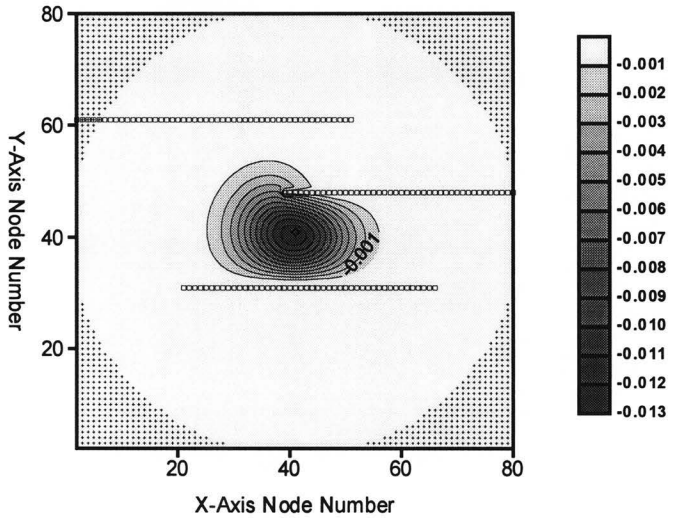


Figure 5.41: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 2 minutes. The maximum Darcy-based drawdown reported for this configuration was 1.3 meters.

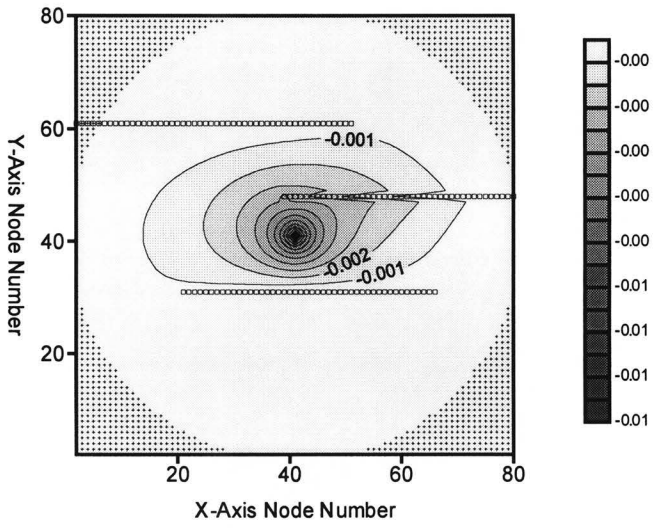


Figure 5.42: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 20 minutes. The maximum Darcy-based drawdown reported for this configuration was 1.4 meters.

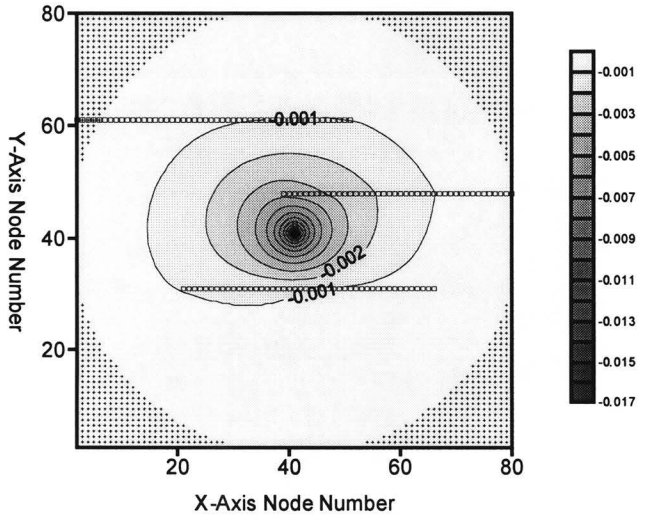


Figure 5.43: Forchheimer-based minus Darcy-based head (m) calculated using a transient, heterogeneous simulation. Elapsed time equals 60 minutes. The maximum Darcy-based drawdown reported for this configuration was 1.5 meters.

Figure 5.44 illustrates the relationship between the drawdown difference predicted by the two models and Reynolds number. In this figure it is shown that, unlike example 5, the area of greatest separation of drawdown surfaces does correspond with areas of the matrix reporting high (relative to other nodes in the matrix) Reynolds numbers. Clearly, a maximum reported Reynolds number of 0.003 is far below the accepted turbulent flow threshold of one. However, when using the equations described in this work, some finite separation of solution surfaces will be present for any simulation.

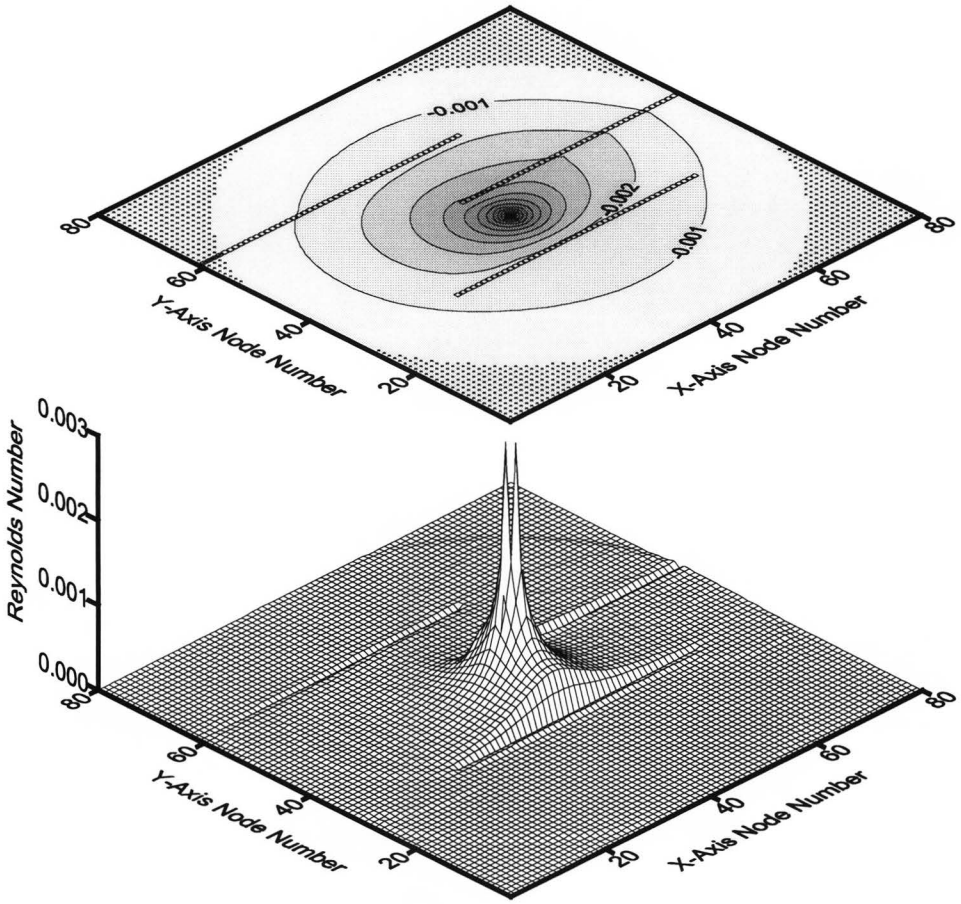


Figure 5.44: Association of Reynolds number and Forchheimer-based minus Darcy-based head (m) for a steady-state, heterogeneous simulation. The matrix was assigned parameters that simulate a grainsize of 0.0001m. The gray lines parallel to the x-axis in the upper plot represent nodes with a grain size of 0.000001 m.

Chapter 6. Conclusions

From its proposition by Darcy in 1856, the assumption of a linear relationship between hydraulic gradient and discharge (Darcy's law) has been widely applied to analytical and numerical modeling of fluid flow through porous matrix. The primary objective of this work was to quantify the error incurred when Darcy's law is applied to numerical modeling. To achieve this, four models were constructed: two-dimensional transient and steady-state Forchheimer-based models, as well as two-dimensional transient and steady-state Darcy-based models.

Following construction, these models were calibrated to analytical solution surfaces (see Appendix 1) for steady-state, radial flow to a well in a homogeneous matrix. Difficulties in calibration led to the development of an algorithm that, when applied in the calculation of apparent hydraulic resistivity at the well node, allowed for the reasonably accurate calculation of potentiometric surfaces.

In order to quantify the error that might result from the application of Darcy's law in situations where it might not apply, six model configurations were explored:

1. Flow to well in a matrix with a simulated average grain size of 0.1 mm with a distribution of small, highly conductive heterogeneities.

2. Flow to well in a matrix with a simulated average grain size of 0.1 mm with a distribution of small, relatively impermeable heterogeneities.
3. Flow to well that is completed in a band of aquifer material with a simulated average grain size of 1 mm enclosed in a matrix with an average grain size of 0.1 mm. The width of the band was reduced in successive simulations to ascertain the effects of increased flow channeling.
4. Flow to a well in a matrix with a simulated average grain size of 0.1 mm where the distance from the well to a large, highly permeable section of aquifer is shortened in successive simulations.
5. Flow to a well in matrix with a simulated average grain size of 0.1 mm with three highly conductive linear heterogeneities dispersed within it.
6. Flow to a well matrix with a simulated average grain size of 0.1 mm with three relatively impermeable linear heterogeneities dispersed within it.

In these examples, the effect of the zones of greater or lesser hydraulic conductivity (heterogeneities) is calculated for steady-state and transient conditions. The geometry and placement of the heterogeneities was arbitrary, but were selected to provide simple, representative examples of the types of conductivity variations that might be present in the vicinity of an extraction well. As described in Chapter 5, fluid velocities and Reynolds numbers are calculated from the head distributions recorded. These variables were written

to tab-delimited text files at the end of each simulation. From these text files, contour and three-dimensional surface plots were generated.

The following sections summarize the results and conclusions garnered during each example configuration.

6.1 Example 1 Results and Conclusions

Example 1 explored the effects a distribution of small, conductive heterogeneities has on the accuracy of Darcy-based drawdown predictions. Here, regularly-spaced zones with a simulated grain size of 10 mm were introduced into a matrix with a grain size of 0.1 mm and a central well extracting approximately 1 gallon per second.

The reference plot for a matrix without heterogeneities (Figure 5.1) shows very little difference between the potentiometric surfaces calculated by the two steady-state models. When the conductive zones are introduced, the separation between the two curves makes up a relatively large percentage of the total Darcian drawdown. However, the significant difference in head predicted by the two models is generally confined to the area around the well.

The plots displaying the evolution of the surface separation through time for example 1 show that a large amount of the difference in drawdown

occurs within the first few seconds. However, the ratio of drawdown difference to the total Darcian drawdown increases from approximately 0.12 percent to 0.29. This shows that although the drawdown at early times is significant, non-Darcy behavior does not fully develop until later in the simulation. In other words, the error produced by the Darcy-based model is smallest at early times, and grows greater as time progresses.

The time-step surface plots of Reynolds number give a possible explanation for this phenomenon. At early times, relatively few of the conductive heterogeneities show elevated Reynolds numbers. However, as time progresses, in increasing number of the highly conductive zones showed substantial increases in their associated Reynolds number. This evidence points to the conclusion that the separation between the two model solutions is an effect that is cumulative over the entire matrix.

6.2 Example 2 Results and Conclusions

Example 2 explores flow in a matrix with a configuration of heterogeneities inverse to that in example 1. Here, flow to a central well passes through a relatively coarse-grained matrix regularly interspersed with small, relatively impermeable zones. In this simulation, the well extracts 0.5 m^3/s (approximately 132 gallons per second) from a matrix with an average grain size of 1 mm, and heterogeneities with a grain size of 0.01 mm. As

expected, the course matrix and high pumping rate in the homogeneous, benchmark simulation (illustrated in Figure 5.13) shows a much larger separation of solution surfaces than that illustrated in Figure 5.1. When the heterogeneities are introduced, the results are dramatic. The separation between models solutions increased from 0.38 meters for the homogenous simulation to almost 5.0 meters (> 1200 percent) for the steady-state solution incorporating the impermeable zones. This figure is especially large when compared to the 4.9 meters of total Darcy-based drawdown. The 5.0 meters of difference denotes a 9.9 m Forchheimer-based drawdown at the well, however, the reader should keep in mind that the rate of pumping used in this example is extremely large.

The time-series contour plots of drawdown difference for this example show that equilibrium conditions are reached relatively quickly. After 30 seconds, almost 90 percent of the drawdown difference is reached. Almost 100 percent of the separation between solution curves is achieved by an elapsed simulation time of two minutes. It is informative to note that the Darcy-based model evolved through time in an almost identical manner; approximately 90 percent of total drawdown was reached within 30 seconds, while maximum drawdown was attained by two minutes. However, as shown in Figures 5.15 through 5.18, the impermeable heterogeneities require substantially more time to approximate steady-state conditions.

The distribution of matrix Reynolds numbers for example 2 is more difficult to evaluate. Large Reynolds numbers were expected to accompany the large separation of solution surfaces calculated by these simulations. Instead, as Figure 5.22 illustrates, the maximum Reynolds number value is approximately 6.0, much lower than the 90 calculated for example 1. Again, the high degree of solution separation might be due to a cumulative effect throughout the matrix. Although the Reynolds number values are much lower in example 2, many more nodes exhibit an elevated value than in example 1.

6.3 Example 3 Results and Conclusions

The results of successive model runs simulating a pumping well located in the center of a relatively permeable band of aquifer material is shown in example 3. To observe the effects produced when flow is increasingly constricted, the width of the permeable band is reduced from 52.5 meters to 17.5 meters over the span of three simulations. For these model runs, the permeable band is composed of aquifer material with an average grain size of 10 mm, while the blocks of matrix material enclosing it out in average grain size of 0.1 mm. The central pumping well is discharging a rate of 0.1 m^3/s (approximately 26 gallons per second).

As a baseline for comparison, Figure 5.23 reports the model drawdown difference predicted for a homogeneous matrix of grain size equal

to 1 mm. The difference of 12 mm at the well bore is small; the ratio of the drawdown difference to total Darcy-based drawdown (approximately 1.2 meters) is approximately 1.0 percent.

The separation between solution surfaces increases as flow is constrained to the permeable band by the relatively fine-grained matrix blocks on either side of it. Figure 5.24 shows that the drawdown difference has increased to 0.55 meters while the total Darcy-based drawdown predicted remained at a constant 1.2 meters. The drawdown difference increased to 0.70 meters in the simulation plotted in Figure 5.25. Finally, a 0.80 m solution separation was calculated when the conductive band was narrowed to 17.5 meters for the simulation recorded in Figure 5.26. The ratio of drawdown difference to total Darcy-based drawdown was increased to approximately 66 percent by reducing the thickness of the conductive zone.

Relatively low Reynolds numbers were calculated for the nodes in example 3. It is apparent that the maximum value of matrix Reynolds numbers alone is not a reliable indicator of the magnitude of the error resulting from the application of the Darcy-based model. These observations corroborate Lage's conclusion that the use of the Reynolds number to predict non-Darcy flow is incorrect and should be discontinued (Lage, 1998).

6.4 Example 4 Results and Conclusions

Close proximity of highly conductive aquifer material to a well can have a significant impact on any flow model. In example 4, the effect of this type of local heterogeneity on the accuracy of Darcy-based models is explored. In four separate steady-state simulations, a large heterogeneity with a simulated average grain size of 10 mm is placed successively closer to a well completed within a matrix with an average grain size of 0.1 mm. The central well was pumped at a rate of approximately one gallon per second (0.004 m^3/s) in all cases.

As with previous examples, the introduction of conductive heterogeneities increased the separation between the potentiometric surfaces calculated using Forchheimer-based in Darcy-based models. Identically with example 1, the reference drawdown difference was 2.6 mm for a homogenous matrix of a grain size of 0.1 mm. The drawdown difference failed to increase when the conductive zone terminated 50 meters from the well, showing that the conductive zone as some maximum range of influence on Darcy-based model accuracy. A moderate increase in solution separation (2.6 to 19 mm) is seen when the conductive zone intrudes to within 25 meters of the well. Increasingly substantial increases in drawdown difference occurred when the conductive heterogeneity was placed within 12.5 meters and 7.5 meters of the

well node in the next two simulations. The Darcy-based model's underestimation of drawdown was compounded in the simulations illustrated by Figures 5.28 through 5.31. Percent drawdown error increased because the total Darcy-based drawdown decreased as the distance between the conductive zone and the well was reduced.

In example 4, the relationship between drawdown difference and Reynolds number shown in Figure 5.32 was expected. In this case, large Reynolds numbers were calculated for the portion of the matrix immediately adjacent to the well node, as well as for nodes in the conductive zone.

6.5 Example 5 Results and Conclusions

Simulations run using the matrix parameter configuration outlined in example 5 shown the error contracted when a relatively fine-grained matrix is cut through with planar permeable zones. As with examples 1,3, and 4, a central well is completed within a matrix with a simulated grain size of 0.1 mm, while the conductive zones incorporate a grain size of 10 mm. A well extraction rate of approximately one gallon per second was assigned for the simulations incorporated within example 5. For reference, the drawdown difference plot for a matrix of this size is illustrated in Figure 5.1.

Figure 5.33 shows the relatively small amount of Darcy-based error associated with this model configuration. The 18 mm maximum separation of solution surfaces represents a 1.2 percent total drawdown error for steady-state conditions.

The transformation through time of the head difference surface illustrated by Figures 5.34 through 5.37 shows a distinct evolution of its geometry. For the first few seconds after the initiation of fluid extraction, the shape of the drawdown difference surface is that of a relatively regular cone. Over the next few minutes, this cone intersects the planar conductive zone that extends to the constant head boundary. The flow then follows this conduit, and, as fluid velocity increases, the drawdown difference in this zone also increases, distorting the regularity of the cone. As equilibrium is approached, the drawdown difference surface regains some radial symmetry.

Figure 5.38 compares Reynolds number distributions with the steady-state drawdown difference curve plotted for example 5. In this case, large Reynolds numbers are calculated for the highly conductive zones, but not for the region adjacent to the well. This distribution of Reynolds numbers is distinctly different from previous examples. In almost every other case, the largest Reynolds numbers were restricted to the region about the well. The relatively small number of nodes exhibiting large Reynolds numbers coupled with the small drawdown surface separation provides additional evidence that

the magnitude of Darcian error is dependent on the percentage of the matrix nodes showing elevated Reynolds values.

6.6 Examples 6 Results and Conclusions

Example 6 explores a matrix configuration converse to that shown in example 5. Here, a moderately fine-grained matrix (grain size equals 0.1 mm) is intruded upon by relatively impermeable, planar (grain size equals 0.001 mm) heterogeneities. As with other examples incorporating a matrix of this grain size, the well discharge rate is set at approximately one gallon per second (0.004 m^3/s).

The amount of difference between head calculated using the Forchheimer-based model and the Darcy-based model is very similar to the difference calculated in example 5. Again, the surface separation is approximately 17 mm, while the Darcy-based percent drawdown error is about one percent. Like example 5, the shape of the solution difference surface undergoes an evolution as time progresses from the initiation of fluid extraction to steady-state conditions. At intermediate times, there is a distortion of the radial symmetry developed during early time steps. This symmetry is restored as simulation time is extended.

Figure 5.44 illustrates the relationship between the drawdown difference predicted and Reynolds number for example 6. In this case, the nodes with the largest Reynolds numbers are associated with the region directly adjacent to the well node. However, these Reynolds numbers are of a very small magnitude, an observation that is consistent with the small solution-surface separation.

6.7 Summary of Conclusions

One of the goals of this study was to ascertain when and under what conditions it may be unacceptable to apply models incorporating Darcy's law. To this end, numerous numerical simulations incorporating varying matrix parameters were run. The results of these model runs show that, in general, Darcy-based models are reasonably accurate in situations where the aquifer is relatively fine-grained, homogeneous and not highly stressed. The determination of the applicability of a model to a particular situation rests with the individual researcher. The goals of the study are paramount when deciding the applicability of Darcy's law. If one wishes to describe subsurface fluid flow on a regional scale, then Darcy-based numerical models are sufficiently accurate. As shown by the previous examples, there can be a significant difference in the head calculated in areas where the aquifer is highly stressed (i.e. adjacent to a pumping well), however, the amount of head difference decreases rapidly as distance from the well increases. For example,

a maximum head difference of five meters was reported for the well node in example 2, but the difference decreased to less than 0.5 meters at a distance of approximately 50 meters from the well. Similar trends were observed in all other model runs. With this in mind, it is apparent that, as long as regional variations in matrix hydraulic conductivity are faithfully incorporated into the model, little precision (on a regional scale) is gained by applying the computationally expensive algorithms, such as described in this work.

Alternately, if a researcher is attempting to accurately predict the position of the potentiometric surface in an area near a source of hydrologic stress (i.e. an injection/extraction well) then it is advisable to employ a Forchheimer-based model. In this case, the magnitude of the stress as well as the proximity of any heterogeneity with significantly different hydrologic properties is also a factor. In example 4, virtually no head difference was calculated for a homogenous matrix (grain size equals 0.0001 m), or for the case where a highly conductive heterogeneity terminates 50 meters from the well node. However, the maximum head difference increased to approximately 30% of the total Darcy-based drawdown (0.38/1.3 m) as the distance between the conductive zone and the well was reduced to 7.5 meters.

As the examples in this work show, there is little difference between the potentiometric surfaces generated using the two types of models for low pumping rates in homogenous matrices. However, significant errors can be

incurred when Darcy-based models are applied to coarse-grained, highly stressed or heterogeneous aquifers.

Ease of use and computational efficiency are also major factors when deciding whether to use a Forchheimer-based or Darcy-based model. If one wishes to employ a Forchheimer-based model, the hydraulic resistivity, as well as Forchheimer's nonlinear resistivity coefficient (b) need to be defined for every node in the model matrix. Computational efficiency also degrades when employing a Forchheimer-based model because of the increased complexity of the governing equations.

Accurate calculation of the hydraulic gradient at the well node presents a significant hurdle to the successful application of a Forchheimer-based model. The calculation of the apparent hydraulic resistivity of a model cell is dependent upon the accurate approximation of the local hydraulic gradient. Coarse model matrix discretization, as well as the non-continuous nature of the potentiometric surface due to the representation of wells with source/sink terms in the governing flow equations can lead to significant errors when calculating hydraulic gradient. Although commonplace, these effects are not easily overcome. The correction algorithm employed by the model described in this report performed adequately for the purposes of this research. However, significant deviations from the analytical solution were reported, and it is clear that if greater precision is required this algorithm must

be modified or replaced. Although linear models must also incorporate correction algorithms to compensate for the inherent numerical errors, Darcy-based models enjoy a serious advantage in this respect: Darcian matrix cell resistivity is not affected by the local hydraulic gradient, therefore gradient calculations are not necessary during iteration.

The two-dimensional limitation of this model is also a handicap. The necessity of simulating heterogeneities that penetrated the entire aquifer thickness probably magnified the amount of drawdown curve separation. In order to quantify Darcy-based error more accurately, future work should incorporate flow in three dimensions.

Appendix 1. Analytical Solutions

Derivation of the Darcy-based and Forchheimer-based analytical solutions for drawdown resulting from radial flow to a fully penetrating well in a confined aquifer is described below. For both cases, the solutions are exclusively applicable to steady-state flow in a homogenous matrix.

Both the Darcy and Forchheimer-based are derived in the manner first described by G. Theim (Theim, 1906). First, the volumetric rate of flow (Q) is equated to the average velocity of flow times the area of flow (equation A1-1a). Next, the average pore-fluid velocity (q) is substituted into equation A1-1a. Equation A1-1b is then rearranged to find pore-fluid velocity (q) explicitly in terms of discharge and area.

$Q = vA$ Discharge [L^3/t] = velocity [L/t] * area of flow [L^2] (A1-1a)

$Q = qA$ Where q is average pore fluid velocity. (A1-1b)

$$q = \frac{Q}{A} \quad \text{(A1-1c)}$$

Finally, the equation for the curved surface area of a cylinder is substituted for "A" to produce equation A1-1d.

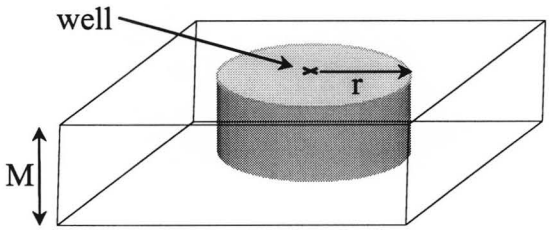


Figure A1.1: Simplified aquifer diagram.

$$\frac{Q}{A} = q = \frac{Q}{2\pi r M} \qquad \text{(A1-1d)}$$

where M is the aquifer thickness
r is the radial distance from the well

A1.1 Analytical solution to the Forchheimer relation

$$-dh/dL = bq^2 + aq$$

Forchheimer's Relation

Using the quadratic formula to solve Forchheimer (assuming positive q) results in equation A1-2:

$$q = \frac{-a + \left(a^2 - 4b\, dh/dr\right)^{1/2}}{2b} \tag{A1-2}$$

Combining Equations A1-1d and A1-2 results in equation A1-3:

$$\frac{Q}{2\pi r M} = \frac{-a + \left(a^2 - 4b\, dh/dr\right)^{1/2}}{2b} \tag{A1-3}$$

Solving for Equation A1-3 for dh/dr results in Equation A1-4:

$$dh/dr = \left(\frac{-1}{4b}\right)\left[\frac{Q^2 b^2}{\pi^2 r^2 M^2} + \frac{2aQb}{\pi r M}\right] \tag{A1-4}$$

If two observation wells exist where the head is h_1 at a distance of r_1 and h_2 at a distance of r_2, integration of Equation 1-4 using these boundary conditions results in equation 1-5b.

$$\int_{h_1}^{h_2} dh = \int_{r_1}^{r_2}\left(\frac{-1}{4b}\right)\left[\frac{Q^2 b^2}{\pi^2 r^2 M^2} + \frac{2aQb}{\pi r M}\right] dr \tag{A1-5}$$

$$(h_2 - h_1) = -\left(\frac{Q^2 b}{4\pi^2 M^2}\right)\left(\frac{1}{r_1} - \frac{1}{r_2}\right) - \left(\frac{aQ}{2\pi M}\right)\ln\left(\frac{r_2}{r_1}\right) \tag{A1-6}$$

If the head is known at some distance from the well (i.e. a constant head boundary) then equation A1-6 can be solved to find the head at any point on the potentiometric surface (eqn. A1-7).

$$h_1 = h_2 + \left(\frac{Q^2 b}{4\pi^2 M^2}\right)\left(\frac{1}{r_1} - \frac{1}{r_2}\right) + \left(\frac{aQ}{2\pi M}\right)\ln\left(\frac{r_2}{r_1}\right) \qquad \text{(A1-7)}$$

Figure A1.2 diagrams the configuration and use of the variables described above.

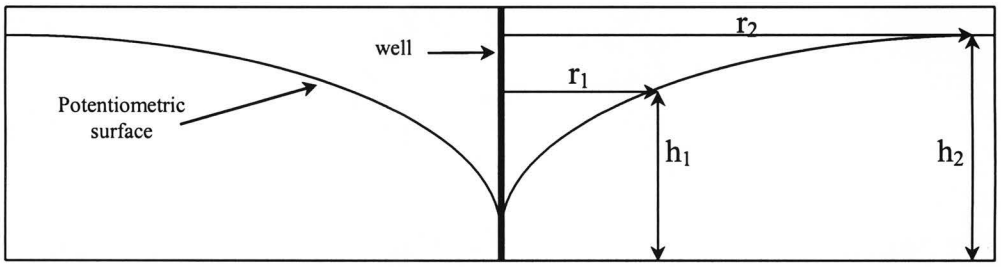


Figure A1.2: Cross-sectional aquifer diagram.

A1.2 ANALYTICAL SOLUTION TO THE DARCY EQUATION

In this section, a solution to the Darcy equation is derived using a technique analogous to the one used above to derive an analytical solution to the Forchheimer equation.

$$-dh/dL = aq$$

Darcy's law

Darcy's law is solved for specific discharge (q) and then substituted into equation A1-1d.

$$\frac{Q}{2\pi rM} = \frac{1}{a}\frac{dh}{dr} \tag{A1-8}$$

Equation (A1-8) is solved for hydraulic gradient, and integrated using the boundary conditions described in the previous section.

$$\frac{dh}{dr} = \frac{Qa}{2\pi rM} \tag{A1-9}$$

$$\int_{h_1}^{h_2} dh = \int_{r_1}^{r_2} \left(\frac{-Qa}{2\pi M}\right)\left(\frac{1}{r}\right) dr \tag{A1-10}$$

$$\left(h_2 - h_1\right) = -\left(\frac{Qa}{2\pi M}\right)\ln\left(\frac{r_2}{r_1}\right) \tag{A1-11}$$

$$h_1 = h_2 + \left(\frac{Qa}{2\pi M}\right)\ln\left(\frac{r_2}{r_1}\right) \tag{A1-12}$$

If the head is known at some distance from the well (i.e. a constant head boundary) then Equation A1-11 can be solved to find the head at any point on the potentiometric surface (eqn. A1-12).

Appendix 2. Resistivity Coefficients

Tables A2.1a, A2.1b and A2.1c contain the values of the Forchheimer coefficients of linear hydraulic resistivity (a) and nonlinear hydraulic resistivity (b) reported by various authors. These values were used to formulate the equations shown in Figures A2.1 and A2.2. Extrapolation of Forchheimer coefficients (a,b) for grainsizes not explicitly listed in the tables was accomplished using these equations.

For review and listings of the Forchheimer coefficients reported by various authors see Basak (1976) and Venkataraman and Rao (1998).

Grain Diam. (m)	Grain Diam. (mm)	a (s/m)	b (s^2/m^2)	Source
0.00054	0.54	739	7450	Ahmed (1967)
0.000764	0.764	380	4540	
0.00107	1.07	230	3080	
0.0014	1.4	149	2400	
0.00199	1.99	93.8	1790	
0.00095	0.95	78.9	2230	Dudgeon (1966)
0.00101	1.01	99	2630	Subba (1969)
0.00101	1.01	115	3450	
0.0017	1.7	32.5	1100	
0.0017	1.7	47.5	1990	
0.0017	1.7	40	1640	

Table A2.1a Reported values of "a" and "b" Forchheimer coefficients. Modified from Basak (1976)

Grain Diam. (m)	Grain Diam. (mm)	a (s/m)	b (s^2/m^2)	Source
0.0017	1.7	51.5	3300	Rao and Suresh (1970)
0.0017	1.7	170	16000	Rao and Suresh (1972)
0.00286	2.86	13.5	720	
0.00286	2.86	22.5	880	
0.00286	2.86	34	4000	
0.00404	4.04	7.5	530	
0.00404	4.04	10.5	780	Rao and Suresh (1970)
0.0055	5.5	4.3	430	
0.0055	5.5	7.5	550	
0.0055	5.5	10.5	780	
0.0055	5.5	23	3800	
0.0044	4.4	72	4800	Rao and Suresh (1972)
0.00286	2.86	52	6400	
0.002	2	19.04	2174	
0.011	11	1.15	162	Dudgeon (1966)
0.012	12	1.89	262	
0.019	19	0.82	145	
0.04	40	0.24	51	
0.084	84	0.064	15	
0.019	19	1.04	127	
0.0048	4.8	15.14	825	Volker (1969)
0.00318	3.18	28.8	930	Volker (1975)
0.00636	6.36	6	420	Niranjan (1973)
0.00636	6.36	8	480	
0.00636	6.36	10	670	
0.01115	11.15	1.6	260	
0.01115	11.15	7.6	410	
0.01115	11.15	16	540	
0.0175	17.5	1	102	
0.0175	17.5	2	105	
0.0175	17.5	3.5	173	
0.0238	23.8	0.5	40	

Table A2.1b Reported values of "a" and "b" Forchheimer coefficients. Modified from Basak (1976)

Grain Diam. (m)	Grain Diam. (mm)	a (s/m)	b (s^2/m^2)	Source
0.0238	23.8	0.5	82.4	Niranjan (1973)
0.0238	23.8	0.7	148	
0.0333	33.3	0.8	21	
0.0333	33.3	1	29	
0.0333	33.3	5.5	40	
0.0466	46.6	0.2	10	
0.0466	46.6	0.4	19	
0.0466	46.6	2.8	37.2	
0.00258	2.58	69.4	1650	
0.0055	5.5	0.232	5.5	Ahmed (1967)
0.00815	8.15	0.45	5.76	Sastry (1976)
0.0147	14.7	0.5	18.8	
0.021	21	0.393	8.25	

Table A2.1c Reported values of "a" and "b" Forchheimer coefficients. Modified from Basak (1976)

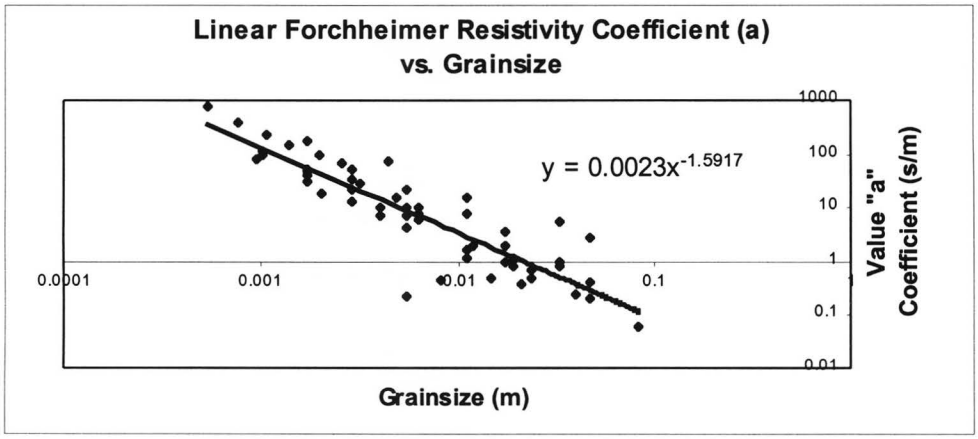


Figure A2.1: Plot of linear hydraulic resistivity (a) vs. grainsize. The equation defining the least-squares regression curve was used for parameter extrapolation in this work.

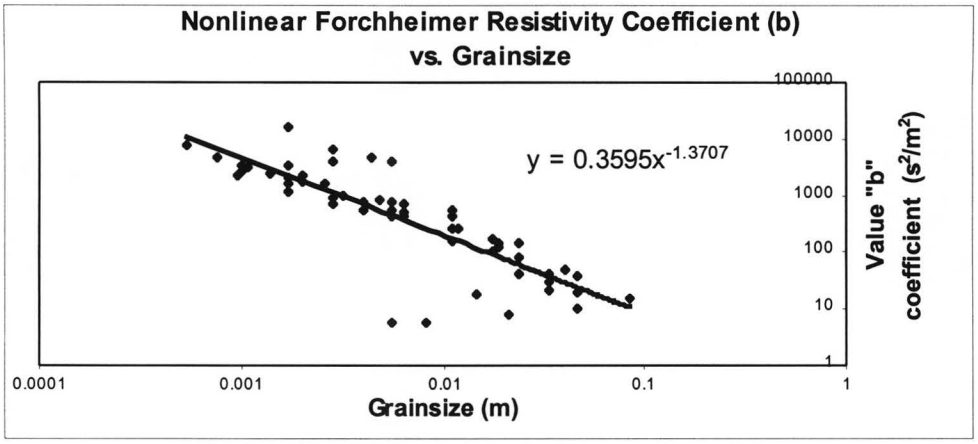


Figure A2.2: Plot of nonlinear hydraulic resistivity (b) vs. grainsize. The equation defining the least-squares regression curve was used for parameter extrapolation in this work.

Appendix 3. Steady-State Model Code

A3.1 Steady-State Main Block

```
C     ******************************************************
c     2D, Steady-State Non-Darcy Flow  Through a Porous Matrix
c     James Bene'  Spring 1999
c     ******************************************************

      Program Turb2D_steady
      implicit none

c     _____Declare all arrays, parameters and variables_____

      integer length,W1X,W1Y
      integer UnitNum,SeriesUnitNum,InfoUnitNum
      integer Itcounter,x,y,test,Radial
      integer XStart,XEnd,YStart,YEnd
      integer KBeg(length)
      integer KEnd(length)
      character *6 MKstr,HetKstr,BType
      character *3 TimeUnit
      real(8) a,b,discrimx,discrimy,MK,HetK
      real(8) Heta,Hetb,Hetd
      real Het1XStart,Het1XEnd,Het2XStart,Het2XEnd,Het3XStart,Het3XEnd
      real Het1YStart,Het1YEnd,Het2YStart,Het2YEnd,Het3YStart,Het3YEnd
      real(8) AbError1,AbError2,DAbError1,DabError2
      real(8) W1Q,TQMult,TQMultSlope,Qsign,MagV
      real(8) dhdx,dhdy,dhdx1,dhdx2,dhdy1,dhdy2
      real(8) North,South,East,West
      real(8) Err,DErr,MaxError,DMaxError
      real(8) Xdist,Ydist,Totdist,Tempdist,Radius
      real(8) Tol,Rho,Mu,omega,hOld,DhOld
      real(8) Ma,Mb,Md,No_Flow,Spots
      real(8) InitialHead,dx,dy,C,T,M
      parameter(UnitNum=10,InfoUnitNum=11,SeriesUnitNum=12)
      parameter(length=82)
      real(8) h(length,length)
      real(8) aa(length,length)
      real(8) bb(length,length)
```

```
real(8) d(length,length)
real(8) Reynolds(length,length)
real(8) Vx(length,length)
real(8) Vy(length,length)
real(8) DVx(length,length)
real(8) DVy(length,length)
real(8) Dh(length,length)
real(8) Q(length,length)
real(8) discX(length,length)
real(8) discY(length,length)

parameter(TimeUnit='sec')
parameter(No_Flow=0)
parameter(Radial=1)
parameter(Spots=1)
parameter(Tol=0.00005)
parameter(W1X=length/2, W1Y=length/2)
parameter(InitialHead=10)
parameter(Rho=998.0, Mu=0.001)
parameter(omega=1.7)
parameter(dx=2.5,dy=dx)
parameter(Md=0.0001)
parameter(Hetd=0.000001)
parameter(W1Q=.004)
parameter(M=10)

parameter(Het1XStart=1)
parameter(Het1XEnd=1)
parameter(Het1YStart=1)
parameter(Het1YEnd=1)
parameter(Het2XStart=1)
parameter(Het2XEnd=1)
parameter(Het2YStart=1)
parameter(Het2YEnd=1)
parameter(Het3XStart=1)
parameter(Het3XEnd=1)
parameter(Het3YStart=1)
parameter(Het3YEnd=1)

open (unit=UnitNum,file='Turb2D_steady.dat')
open (unit=InfoUnitNum,file='Turb2Dinfo_steady.dat')
open (unit=SeriesUnitNum,file='Turb2Series_steady.dat')

Call CalcCoefs (Md,Ma,Mb,TimeUnit)
```

```
      Call CalcCoefs (Hetd,Heta,Hetb,TimeUnit)
      MK=1/Ma

c     _____Set Initial Matrix Values_____

      write(*,*)'Initializing Matrix:',length,length
      h=InitialHead
      Dh=InitialHead
      aa=Ma
      bb=Mb
      d=Md
      Vx=0.0
      Vy=0.0
      DVx=0.0
      DVy=0.0
      Reynolds=0.0
      discX=0.0
      discY=0.0
      Q=0.0

c     _____Determine Radial Constant-Head Boundary Location_____

      if (Radial.eq.1) then
            Radius=sqrt((real(length)/2-1)**2)
            KBeg=0
            KEnd=0
            do x=1,length
                  tempdist=0
                  do y=1,length
                        Totdist=sqrt((real(w1x)-x)**2+(real(w1y)-y)**2)
                        if ((Totdist.lt.Radius).and.(KBeg(x).eq.0)) then
                              KBeg(x)=y
                        end if
                        if (Totdist.lt.Radius) then
                              KEnd(x)=y
                        end if
                  end do
            end do
      end if

c     _____Insert Heterogeneity Parameters into Matrix_____

      XStart=int(Het1XStart)
```

```
XStart=int(Het1XStart)
XEnd=int(Het1XEnd)
YStart=int(Het1YStart)
YEnd=int(Het1YEnd)
call Init(aa,length,XStart,XEnd,YStart,YEnd,Heta)
call Init(bb,length,XStart,XEnd,YStart,YEnd,Hetb)
call Init(d,length,XStart,XEnd,YStart,YEnd,Hetd)
XStart=int(Het2XStart)
XEnd=int(Het2XEnd)
YStart=int(Het2YStart)
YEnd=int(Het2YEnd)
call Init(aa,length,XStart,XEnd,YStart,YEnd,Heta)
call Init(bb,length,XStart,XEnd,YStart,YEnd,Hetb)
call Init(d,length,XStart,XEnd,YStart,YEnd,Hetd)
XStart=int(Het3XStart)
XEnd=int(Het3XEnd)
YStart=int(Het3YStart)
YEnd=int(Het3YEnd)
call Init(aa,length,XStart,XEnd,YStart,YEnd,Heta)
call Init(bb,length,XStart,XEnd,YStart,YEnd,Hetb)
call Init(d,length,XStart,XEnd,YStart,YEnd,Hetd)

c       _______Place Distributed Heterogeneities (Ex. 1 and 2 only)_______

if (Spots.eq.1) then
        y=7
        do while(y.lt.length-4)
                do x=4,length-4,8
                        XStart=x
                        XEnd=x+2
                        YStart=y
                        YEnd=y+2
                        call Init(aa,length,XStart,XEnd,YStart,YEnd,Heta)
                        call Init(bb,length,XStart,XEnd,YStart,YEnd,Hetb)
                        call Init(d,length,XStart,XEnd,YStart,YEnd,Hetd)
                end do
                y=y+8
        end do
end if

c       _____Set Well Node Parameters_____

Q(w1x,w1y)=-W1Q
```

```
      T=(W1Q/dx/M)*(Mb/Ma)**0.5
      C=exp(-0.2*T)

c     _____Display model parameter values_____

      if (No_Flow.eq.1) then
            BType='NoFlow'
      else
            BType=' Const'
      end if
      write(*,*)'_____'
     write(*,8)'Matrix a:',Ma,'Matrix b:',Mb,'Matrix d:',Md
      write(*,25)'MK:',Mk,Mkstr
      write(*,26)'MT:',MK*M,'Tol:',Tol,
     &                  'Matrix Head:',InitialHead
     write(*,*)'_____'
     write(*,*)'Beginning Iterations...'


c     _____
c       _____Begin Iteration Loop_____


c           ...Begin Forward Sweep of Darcy-Based Matrix...

      DMaxError=Tol*100
      ItCounter=0

      do while (DMaxError.gt.Tol)
            DMaxError=0
            Itcounter=Itcounter+1
            do x=2,length-1
                  do y=KBeg(x),KEnd(x)
                        DhOld=Dh(x,y)
                        Call DarcyHead (x,y,Dh,aa, Q,dx,dy,M,
     &                                    length,DhOld,omega)

c                 ....Calculate max change from previous iteration....

                                    DErr=abs(Dh(x,y)-DhOld)
                                    DAbError1=Dh(x,y)-DhOld
                                    if (DErr .gt. DMaxError) then
                                          DMaxError=DErr
                                          DAbError2=DAbError1
```

```
                                        end if
                    end do
                end do

                if (No_Flow.eq.1) then
                    Call NoFlow2d (Dh,length)
                end if

c                   ....Display iteration statistics and well heads....

                if ((mod(Itcounter,50) .eq.0).or.(ItCounter.eq.1)
     &              .or.(DMaxError.lt.Tol)) then
                    Call DisplayIterStats2d (h,Dh,length,ItCounter,
     &                      AbError2,DAbError2,w1x,w1y)
                end if

c               ...Begin Reverse Sweep of Darcy-Based Matrix...

                DMaxError=0
                Itcounter=Itcounter+1
                do x=length-1,2,-1
                    do y=KEnd(x),KBeg(x),-1
                        DhOld=Dh(x,y)
                        Call DarcyHead (x,y,Dh,aa,Q,dx,dy,M,
     &                                  length,DhOld,omega)

c                   ....Calculate max change from previous iteration....

                        DErr=abs(Dh(x,y)-DhOld)
                        DAbError1=Dh(x,y)-DhOld
                        if (DErr .gt. DMaxError) then
                            DMaxError=DErr
                            DAbError2=DAbError1
                        end if
                    end do
                end do

                if (No_Flow.eq.1) then
                    Call NoFlow2d (Dh,length)
                end if


c                   ....Display iteration statistics and well heads....
```

```
            if ((mod(Itcounter,50) .eq.0).or.(ItCounter.eq.1)
     &          .or.(DMaxError.lt.Tol)) then
                    Call DisplayIterStats2d (h,Dh,length,ItCounter,
     &                                  AbError2,DAbError2,w1x,w1y)
            end if

        end do

c       _____Begin Forchheimer-based Calculations_____

c           ...Begin Forward Sweep of Forchheimer-Based Matrix...

        h=Dh
        MaxError=Tol*100
        ItCounter=0
        do while (MaxError.gt.Tol)
            MaxError=0
            Itcounter=Itcounter+1
            do x=2,length-1
                do y=KBeg(x),KEnd(x)
                    hOld=h(x,y)
                    Call CalcHead2d_st_hom (x,y,h,aa,bb,Q,dx,dy,M,
     &                                  length,hOld,omega,C,w1x,w1y)

c                   ....Calculate max change from previous iteration....

                                        Err=abs(h(x,y)-hOld)
                                        AbError1=h(x,y)-hOld
                                        if (Err .gt. MaxError) then
                                            MaxError=Err
                                            AbError2=AbError1
                                        end if
                end do
            end do

            if (No_Flow.eq.1) then
                Call NoFlow2d (h,length)
            end if

c             ....Display iteration statistics and well heads....

            if ((mod(Itcounter,50) .eq.0).or.(ItCounter.eq.1)
     &          .or.(MaxError.lt.Tol)) then
                    Call DisplayIterStats2d (h,Dh,length,ItCounter,
```

```
     &                                    AbError2,DAbError2,w1x,w1y)
                end if

c               ...Begin Reverse Sweep of Forchheimer-Based Matrix...

                MaxError=0
                Itcounter=Itcounter+1
                do x=length-1,2,-1
                        do y=KEnd(x),KBeg(x),-1
                                hOld=h(x,y)
                                Call CalcHead2d_st_hom (x,y,h,aa,bb,Q,dx,dy,M,
     &                                  length,hOld,omega,C,w1x,w1y)

c                       ....Calculate max change from previous iteration....

                                Err=abs(h(x,y)-hOld)
                                AbError1=h(x,y)-hOld
                                if (Err .gt. MaxError) then
                                        MaxError=Err
                                        AbError2=AbError1
                                end if

                        end do
                end do

                 if (No_Flow.eq.1) then
                        Call NoFlow2d (h,length)
                 end if

c                 ....Display iteration statistics and well heads....

                if ((mod(Itcounter,50) .eq.0).or.(ItCounter.eq.1)
     &              .or.(MaxError.lt.Tol)) then
                        Call DisplayIterStats2d (h,Dh,length,ItCounter,
     &                          AbError2,DAbError2,w1x,w1y)
                end if
        end do

c               _____End Iteration Loop_____
c       ____________________________________________________________

c       _____Calculate Velocities and Reynolds Numbers_____
```

```
      write(*,*)
      write(*,*)'Calculating Velocities and Reynolds Numbers...'
      do x=2,length-1
            do y=2,length-1
                  Call Velocity2d (x,y,h,Dh,aa,bb,Vx,Vy,
     &                              DVx,DVy,MagV,dx,dy,length)
                  Reynolds(x,y)=abs(Rho*sqrt(Vx(x,y)**2
     &                                 +Vy(x,y)**2)*d(x,y)/Mu)

                  dhdx1=abs(h(x+1,y)-h(x,y))/dx
                  dhdx2=abs(h(x,y)-h(x-1,y))/dx
                  dhdy1=abs(h(x,y+1)-h(x,y))/dy
                  dhdy2=abs(h(x,y)-h(x,y-1))/dy
                  dhdx=(dhdx1+dhdx2)/2
                  dhdy=(dhdy1+dhdy2)/2
                  a=aa(x,y)
                  b=bb(x,y)
                  discX(x,y)=sqrt(a*a+4*b*dhdx)
                  discY(x,y)=sqrt(a*a+4*b*dhdy)
            end do
      end do

c     _____Write Results to Screen_____

      write(*,*)'_____'
      write(*,8)'Matrix a:',Ma,'Matrix b:',Mb,'Matrix d:',Md
      write(*,25)'MK:',Mk,Mkstr
      write(*,26)'MT:',MK*M,'Tol:',Tol,'Matrix Head:',InitialHead
      write(*,*)'_____'
      write(*,*)'Delta X:',dx,'M:',M
      write(*,*)'T=',T
      write(*,*)'C=',C
      write(*,*)'Total Simulation Time:  STEADY STATE'
      write(*,*)'_____'
      write(*,*)


c     _____Write cross-section head values to text file_____

      do y=length/2,length-1
            write(12,30)y,h(w1x,y),Dh(w1x,y),h(w1x,y)-Dh(w1x,y)
      end do
      close(unit=SeriesUnitNum)
```

```
c       _____Write full 2D grid to data file_____

        write(*,*)('Writing simulation to file...')
        Call WriteGrid (h,Dh,aa,bb,d,Reynolds,
     &                  Vx,Vy,DVx,DVy,discX,discY,
     &                  dx,dy,length,UnitNum)
        close(unit=UnitNum)

c       _____Write simulation info to data file_____

        Call SimInfo (Md,Ma,Mb,Hetd,Heta,Hetb,
     &                  InitialHead,W1Q,C,dx,Tol,M,
     &                  h,Dh,BType,w1x,w1y,length,InfoUnitNum)
        close(unit=InfoUnitNum)

c       _____Output Format List_____

8       format(A10,G11.4,A11,G11.4,A11,F11.8)
9       format(A14,F12.5,A14,F12.3,A8)
25      format(A10,G13.2,G13.2,A6)
26      format(A10,F10.4,A10,F10.7,A15,F8.2)
30      format(I15,F15.5,F15.5,F15.5)
        Call beepQQ(1000,230)
        Call beepQQ(500,130)
        stop
        end
```

A3.2 FORCHHEIMER COEFFICIENT CALCULATION

```
c       ************************************************************
c       Calculates Forchheimer resistivity coefficients
c       from grain size.  (see Appendix 2)
c       ************************************************************

    Subroutine CalcCoefs (d,a,b,TimeUnit)

        implicit none
        integer test
        real(8) d,a,b
        character *3 TimeUnit
```

```
      if (TimeUnit.eq.'day') then
            a=(0.0023*d**(-1.5917))/86400
            b=(0.3595*d**(-1.3707))/86400/86400
      end if

      if (TimeUnit.eq.'sec') then
            a=(0.0023*d**(-1.5917))
            b=(0.3595*d**(-1.3707))
      end if

      return
      end
```

A3.3 FORCHHEIMER-BASED HEAD CALCULATION

```
c     ************************************************************
c       Calculates steady-state node head values based on
c       the Forchheimer relation
c     ************************************************************

      Subroutine CalcHead2d_st_hom (x,y,h,aa,bb,Q,dx,dy,M,
     &                              length,hOld,omega,C,w1x,w1y)

      implicit none

      integer x,y,x0,y0,length,w1x,w1y,test
      real(8) h(length,length)
      real(8) aa(length,length)
      real(8) bb(length,length)
      real(8) Q(length,length)
      real(8) dx,dy,M,C,Sx1,Sx2,Sy1,Sy2,W1,W2
      real(8) dhdx,dhdy,dhdx1,dhdx2,dhdy1,dhdy2,omega
      real(8) discrimx1,discrimx2,discrimy1,discrimy2
      real(8) a,b,ax1,ax2,bx1,bx2,ay1,ay2,by1,by2
      real(8) T1,T2,T3,T4,hOld,hLast,discrimx,discrimy
      real(8) DSum,PerDx1,PerDx2,PerDy1,PerDy2,temp
      real(8) discx_xp2,discx_xp1,discx_xm2,discx_xm1
      real(8) discy_xp2,discy_xp1,discy_xm2,discy_xm1
      real(8) discx_yp2,discx_yp1,discx_ym2,discx_ym1
      real(8) discy_yp2,discy_yp1,discy_ym2,discy_ym1
```

```
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h(x+1,y)-h(x,y))/dx
dhdx2=abs(h(x,y)-h(x-1,y))/dx
dhdy1=abs(h(x,y+1)-h(x,y))/dy
dhdy2=abs(h(x,y)-h(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
x0=x
y0=y

if (Q(x,y).ne.0) then
        x=x0+2
        y=y0
        a=aa(x,y)
        b=bb(x,y)
        ax1=(aa(x+1,y)+a)/2
        bx1=(bb(x+1,y)+b)/2
        ax2=(aa(x-1,y)+a)/2
        bx2=(bb(x-1,y)+b)/2
        ay1=(aa(x,y+1)+a)/2
        by1=(bb(x,y+1)+b)/2
        ay2=(aa(x,y-1)+a)/2
        by2=(bb(x,y-1)+b)/2
        dhdx1=abs(h(x+1,y)-h(x,y))/dx
        dhdx2=abs(h(x,y)-h(x-1,y))/dx
        dhdy1=abs(h(x,y+1)-h(x,y))/dy
        dhdy2=abs(h(x,y)-h(x,y-1))/dy
        discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
        discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
        discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
        discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
        discx_xp2=(discrimx1+discrimx2)/2
        discy_xp2=(discrimy1+discrimy2)/2
```

```
x=x0+1
y=y0
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h(x+1,y)-h(x,y))/dx
dhdx2=abs(h(x,y)-h(x-1,y))/dx
dhdy1=abs(h(x,y+1)-h(x,y))/dy
dhdy2=abs(h(x,y)-h(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
discx_xp1=(discrimx1+discrimx2)/2
discy_xp1=(discrimy1+discrimy2)/2

x=x0-1
y=y0
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h(x+1,y)-h(x,y))/dx
dhdx2=abs(h(x,y)-h(x-1,y))/dx
dhdy1=abs(h(x,y+1)-h(x,y))/dy
dhdy2=abs(h(x,y)-h(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
```

```
discx_xm1=(discrimx1+discrimx2)/2
discy_xm1=(discrimy1+discrimy2)/2

x=x0-2
y=y0
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h(x+1,y)-h(x,y))/dx
dhdx2=abs(h(x,y)-h(x-1,y))/dx
dhdy1=abs(h(x,y+1)-h(x,y))/dy
dhdy2=abs(h(x,y)-h(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
discx_xm2=(discrimx1+discrimx2)/2
discy_xm2=(discrimy1+discrimy2)/2

x=x0
y=y0+2
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h(x+1,y)-h(x,y))/dx
dhdx2=abs(h(x,y)-h(x-1,y))/dx
dhdy1=abs(h(x,y+1)-h(x,y))/dy
dhdy2=abs(h(x,y)-h(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
```

```
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
discx_yp2=(discrimx1+discrimx2)/2
discy_yp2=(discrimy1+discrimy2)/2

x=x0
y=y0+1
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h(x+1,y)-h(x,y))/dx
dhdx2=abs(h(x,y)-h(x-1,y))/dx
dhdy1=abs(h(x,y+1)-h(x,y))/dy
dhdy2=abs(h(x,y)-h(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
discx_yp1=(discrimx1+discrimx2)/2
discy_yp1=(discrimy1+discrimy2)/2

x=x0
y=y0-1
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h(x+1,y)-h(x,y))/dx
dhdx2=abs(h(x,y)-h(x-1,y))/dx
dhdy1=abs(h(x,y+1)-h(x,y))/dy
dhdy2=abs(h(x,y)-h(x,y-1))/dy
```

```
        discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
        discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
        discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
        discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
        discx_ym1=(discrimx1+discrimx2)/2
        discy_ym1=(discrimy1+discrimy2)/2

        x=x0
        y=y0-2
        a=aa(x,y)
        b=bb(x,y)
        ax1=(aa(x+1,y)+a)/2
        bx1=(bb(x+1,y)+b)/2
        ax2=(aa(x-1,y)+a)/2
        bx2=(bb(x-1,y)+b)/2
        ay1=(aa(x,y+1)+a)/2
        by1=(bb(x,y+1)+b)/2
        ay2=(aa(x,y-1)+a)/2
        by2=(bb(x,y-1)+b)/2
        dhdx1=abs(h(x+1,y)-h(x,y))/dx
        dhdx2=abs(h(x,y)-h(x-1,y))/dx
        dhdy1=abs(h(x,y+1)-h(x,y))/dy
        dhdy2=abs(h(x,y)-h(x,y-1))/dy
        discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
        discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
        discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
        discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
        discx_ym2=(discrimx1+discrimx2)/2
        discy_ym2=(discrimy1+discrimy2)/2

        discrimx1=(discx_xp1+discx_xm1+discx_yp1+discx_ym1)/4
        discrimx2=(discx_xp2+discx_xm2+discx_yp2+discx_ym2)/4
        discrimy1=(discy_xp1+discy_xm1+discy_yp1+discy_ym1)/4
        discrimy2=(discy_xp2+discy_xm2+discy_yp2+discy_ym2)/4
        discrimx=2/(C/discrimx1+(2-C)/discrimx2)
        discrimy=2/(C/discrimy1+(2-C)/discrimy2)
        discrimx1=discrimx
        discrimx2=discrimx
        discrimy1=discrimy
        discrimy2=discrimy
end if

discrimx=(discrimx1+discrimx2)/2
discrimy=(discrimy1+discrimy2)/2
```

```
x=x0
y=y0
T1=(h(x+1,y)+h(x-1,y))/discrimx
T2=(h(x,y+1)+h(x,y-1))/discrimy
T3=Q(x,y)/M
T4=1/(2/discrimx+2/discrimy)
h(x,y)=(T1+T2+T3)*T4
h(x,y)=omega*h(x,y)+(1-omega)*hOld

return
end
```

A3.4 DARCY-BASED HEAD CALCULATION

```
c     ************************************************************
c      Calculates finite-difference node head values based on
c      Darcy's Law.
c     ************************************************************

      Subroutine DarcyHead (x,y,Dh,aa,  Q,dx,dy,M,length,DhOld,omega)

      implicit none
      integer x,y,length,test
      real(8) Dh(length,length)
      real(8) aa(length,length)
      real(8) Q(length,length)
      real(8) dx,dy,M,DC
      real(8) omega
      real(8) a,DhOld
      real(8) ax1,ax2,ay1,ay2,ax,ay,Dhtemp
      real(8) D1,D2,D3,D4

      a=aa(x,y)
      ax1=(aa(x+1,y)+a)/2
      ax2=(aa(x-1,y)+a)/2
      ay1=(aa(x,y+1)+a)/2
      ay2=(aa(x,y-1)+a)/2
      ax=(ax1+ax2)/2
      ay=(ay1+ay2)/2

      D1=(Dh(x+1,y)+Dh(x-1,y))/ax
      D2=(Dh(x,y+1)+Dh(x,y-1))/ay
```

```
      D3=Q(x,y)/M
      D4=1/(2/ax+2/ay)
      Dh(x,y)=(D1+D2+D3)*D4
      Dh(x,y)=omega*Dh(x,y)+(1-omega)*DhOld

      return
      end
```

A3.5 ITERATION STATISTICS DISPLAY

```
c     ************************************************
c     Displays Iteration Statistics
c     ************************************************

      Subroutine DisplayIterStats2d (h,Dh,length,ItCounter,
     &                               AbError2,DAbError2,w1x,w1y)

      implicit none
      integer ItCounter,w1x,w1y,length
      real(8) h(length,length)
      real(8) Dh(length,length)
      real(8) AbError2,DAbError2

      write(*,7)itcounter,'TChng:',AbError2,
     &                    'DChng:',DAbError2,'---- Th:',
     &                    h(w1x,w1y),'Dh:',Dh(w1x,w1y)

7     format(I5,A7,F10.6,A9,F10.6,A9,F8.3,A6,F8.3)
      return
      end
```

A3.6 ARRAY INITIALIZATION

```
c     ************************************************************
c     Sets elements of a square array to Val
c     ************************************************************

      Subroutine Init (Matrix,length,X1,X2,Y1,Y2,Val)

      implicit none
```

```
integer length
real(8) matrix(length,length)
integer X1,X2,Y1,Y2,x,y
real(8) Val

do x=X1,X2
        do y=Y1,Y2
                Matrix(x,y)=Val
        end do
end do
return
end
```

A3.7 NO-FLOW BOUNDARY

```
c       ***********************************************
c       Reset edge nodes for no-flow boundary conditions
c       ***********************************************

        Subroutine NoFlow2d (Matrix,length)
        integer length
        real(8) matrix(length,length)
        integer x,y

        do x=2,length-1
                Matrix(x,1)=Matrix(x,3)
                Matrix(x,length)=Matrix(x,length-2)
        end do
        do y=2,length-1
                Matrix(1,y)=Matrix(3,y)
                Matrix(length,y)=Matrix(length-2,y)
        end do

        return
        end
```

A3.8 WRITE SIMULATION INFORMATION

```
c       ************************************************************
c       Writes simulation parameters and information to a text file.
c       ************************************************************

        Subroutine SimInfo (Md,Ma,Mb,Hetd,Heta,Hetb,InitialHead,W1Q,
     &                      C,dx,Tol,M,h,Dh,BType,w1x,w1y,length,InfoUnitNum)

        implicit none
        integer w1x,w1y,length,InfoUnitNum
        real(8) Ma,Mb,Md,MK
        real(8) Hetd,Heta,Hetb
        real(8) InitialHead,W1Q
        real(8) C,dx,Tol,M
        real(8) h(length,length)
        real(8) Dh(length,length)
        character *6 BType

        MK=1/Ma

        write(InfoUnitNum,14)' Md ','Ma ','Mb ','Hetd ','Heta ','Hetb ',
     &                       'M ','dx ','InitialHead ','Q ','C ',
     &                       'TWellHead ','DWellHead ',
     &                       'length ','Tol ','Boundary '

        write(InfoUnitNum,15)Md,Ma,Mb,
     &                  Hetd,Heta,Hetb,M,dx,InitialHead,W1Q,
     &                  C,h(w1x,w1y),Dh(w1x,w1y),length,Tol,BType

14      format (A,A15,A15,A15,A15,A15,A15,A15,A15,A15,
     &          A15,A15,A15,A15,A15,A15)

15      format (F15.7,F15.5,F15.5,F15.5,F15.5,F15.5,F15.3,F15.3,
     &          F15.3,F15.3,F15.4,F15.3,F15.3,I15,F15.5,A7)

        return
        end
```

A3.9 SPECIFIC DISCHARGE CALCULATION

```
c       ****************************************
c       Calculates Specific Discharge of All Nodes
c       ****************************************

        Subroutine Velocity2d (x,y,h,Dh,aa,bb,Vx,Vy,
     &                         DVx,DVy,MagV,dx,dy,length)

        implicit none
        integer length,x,y,test
        real(8) h(length,length)
        real(8) Dh(length,length)
        real(8) aa(length,length)
        real(8) bb(length,length)
        real(8) Vx(length,length)
        real(8) Vy(length,length)
        real(8) DVx(length,length)
        real(8) DVy(length,length)
        real(8) dx,dy,MagV
        real(8) a,b,discrimx,discrimy
        real(8) dhdx,dhdy,Darcydhdx,Darcydhdy
        real(8) North,South,East,West

        a=aa(x,y)
        b=bb(x,y)

c               _____Calculate Non-linear Velocities_____

        dhdy=(h(x,y+1)-h(x,y))/dy
        if (dhdy.gt.0.0) then
                discrimy=sqrt(a*a+4*b*dhdy)
                North=(a-discrimy)/(2*b)
        else
                discrimy=sqrt(a*a-4*b*dhdy)
                North=(-a+discrimy)/(2*b)
        end if

        dhdy=(h(x,y)-h(x,y-1))/dy
        if (dhdy.gt.0.0) then
                discrimy=sqrt(a*a+4*b*dhdy)
                South=(a-discrimy)/(2*b)
        else
```

```
            discrimy=sqrt(a*a-4*b*dhdy)
            South=(-a+discrimy)/(2*b)
end if

dhdx=(h(x+1,y)-h(x,y))/dx
if (dhdx.gt.0.0) then
            discrimx=sqrt(a*a+4*b*dhdx)
            East=(a-discrimx)/(2*b)
else
            discrimx=sqrt(a*a-4*b*dhdx)
            East=(-a+discrimx)/(2*b)
end if

dhdx=(h(x,y)-h(x-1,y))/dx
if (dhdx.gt.0.0) then
            discrimx=sqrt(a*a+4*b*dhdx)
            West=(a-discrimx)/(2*b)
else
            discrimx=sqrt(a*a-4*b*dhdx)
            West=(-a+discrimx)/(2*b)
end if

Vx(x,y)=(East+West)/2
Vy(x,y)=(North+South)/2
MagV=sqrt(Vx(x,y)**2+Vy(x,y)**2)


c           _________Calculate Darcian Velocities_________

            Darcydhdy=(Dh(x,y+1)-Dh(x,y))/dy
            North=(-1/a)*Darcydhdy

            Darcydhdy=(Dh(x,y)-Dh(x,y-1))/dy
            South=(-1/a)*Darcydhdy

            Darcydhdx=(Dh(x+1,y)-Dh(x,y))/dx
            East=(-1/a)*Darcydhdx

            Darcydhdx=(Dh(x,y)-Dh(x-1,y))/dx
            West=(-1/a)*Darcydhdx

            DVx(x,y)=(East+West)/2
            DVy(x,y)=(North+South)/2
            return
```

```
      end
```

A3.10 WRITE RESULTS

```
c     ******************************************************
c     Writes simulation results to a text file.
c     ******************************************************

      Subroutine WriteGrid (h,Dh,aa,bb,d,Reynolds,
     &                  Vx,Vy,DVx,DVy,discX,discY,
     &                  dx,dy,length,UnitNum)

      implicit none
      integer length,x,y,UnitNum,test
      real(8) h(length,length)
      real(8) Dh(length,length)
      real(8) aa(length,length)
      real(8) bb(length,length)
      real(8) d(length,length)
      real(8) Reynolds(length,length)
      real(8) Vx(length,length)
      real(8) Vy(length,length)
      real(8) DVx(length,length)
      real(8) DVy(length,length)
      real(8) discX(length,length)
      real(8) discY(length,length)
      real(8) dhdx,dhdy,dhdL,dx,dy,dhdx1,dhdx2,dhdy1,dhdy2

      write(UnitNum,50)'X ','Y ','h ','Dh ','h-Dh','Vx ','Vy ',
     &                  'MagV ','Reynolds ','DVx ', 'DVy ',
     &                  'Size ','dhdx ','dhdy ',' dhdL ',
     &                  'discX/a ','discY/a ','Magdisc/a ','MagV-MagDV '

      do x=2,length-2
            do y=2,length-2
                  dhdx1=abs(h(x+1,y)-h(x,y))/dx
                  dhdx2=abs(h(x,y)-h(x-1,y))/dx
                  dhdy1=abs(h(x,y+1)-h(x,y))/dy
                  dhdy2=abs(h(x,y)-h(x,y-1))/dy
                  dhdx=(dhdx1+dhdx2)/2
                  dhdy=(dhdy1+dhdy2)/2
                  dhdL=sqrt(dhdx**2+dhdy**2)
```

```
                write(UnitNum,5)x,y,h(x,y),Dh(x,y),
     &                  h(x,y)-Dh(x,y),Vx(x,y),Vy(x,y),
     &                  sqrt(Vx(x,y)**2+Vy(x,y)**2),Reynolds(x,y),
     &                  DVx(x,y),DVy(x,y),d(x,y),dhdx,dhdy,
     &                  dhdL,discX(x,y)/aa(x,y),discY(x,y)/aa(x,y),
     &                  sqrt(discX(x,y)**2+discY(x,y)**2)/sqrt(2*aa(x,y)),
     &                  sqrt(Vx(x,y)**2+Vy(x,y)**2)
     &                  -sqrt(DVx(x,y)**2+DVy(x,y)**2)
              end do
        end do

5       format (I25,I25,F25.5,F25.5,F25.5,F25.5,F25.5,
     &          F25.5,F25.5,F25.5,F25.5,F25.5,F25.5,
     &          F25.5,F25.5,F25.5,F25.5,F25.5,F25.5)
50      format (A10,A10,A10,A10,A10,A10,A10,A10,A10,A10,
     &          A10,A10,A10,A10,A10,A10,A10,A10,A10)

        return
        end
```

Appendix 4. Transient Model Code

In this section, the FORTRAN 77 code for the two-dimensional, transient model is presented. Some of the subroutines called by the transient main block are identical to those used in the steady-state model; those sections are listed in Appendix 3.

A4.1 Trasient Main Block

```
c     ************************************************************
c     2D Transient Flow through a porous matrix
c     James Bene'  Spring 1999
c     ************************************************************

      Program Turb2D_transient

c     __________Declare all arrays, parameters and variables_________

      implicit none
      integer length,W1X,W1Y
      integer UnitNum,SeriesUnitNum,InfoUnitNum
      integer Itcounter,x,y,test,XE,YE,timestep,TotTimestep
      integer XStart,XEnd,YStart,YEnd
      character *6 BType
      character *3 TimeUnit
      real(8) a,b,discrimx,discrimy,MK,HetK
      real(8) AbError1,AbError2,DAbError1,DabError2
      real(8) W1Q,TQMult,TQMultSlope,Qsign,MagV
      real(8) dhdx,dhdy,dhdx1,dhdx2,dhdy1,dhdy2
      real(8) Heta,Hetb,Hetd
      real(8) Err,DErr,MaxError,DMaxError
      real(8) Xdist,Ydist,Totdist,Tempdist,Radius
      real(8) Tol,Rho,Mu,omega,temph,tempDh
      real Het1XStart,Het1XEnd,Het2XStart,Het2XEnd,Het3XStart,Het3XEnd
      real Het1YStart,Het1YEnd,Het2YStart,Het2YEnd,Het3YStart,Het3YEnd
      real(8) Ma,Mb,Md,No_Flow,TotTime,Dhtemp,hOld,Radial,Spots
```

```
real(8) InitialHead,dx,dy,dt,C,T,M
real(8) g,n,Ss
parameter(UnitNum=10,InfoUnitNum=11,SeriesUnitNum=12)

parameter(length=82)
real(8) h(length,length)
real(8) h12(length,length)
real(8) hLast(length,length)
real(8) aa(length,length)
real(8) bb(length,length)
real(8) d(length,length)
real(8) Reynolds(length,length)
real(8) Vx(length,length)
real(8) Vy(length,length)
real(8) DVx(length,length)
real(8) DVy(length,length)
real(8) Dh(length,length)
real(8) DhOld(length,length)
real(8) Q(length,length)
real(8) discX(length,length)
real(8) discY(length,length)
integer KBeg(length)
integer KEnd(length)

parameter(Rho=998.0, g=9.81, Mu=0.001, n=0.3)
parameter(Ss=Rho*g*(1.5e-8 + n*4.8e-10))
parameter(TimeUnit='sec')
parameter(No_Flow=0)
parameter(Radial=1)
parameter(Spots=0)
parameter(Tol=0.00005)
parameter(W1X=length/2, W1Y=length/2)
parameter(InitialHead=10)
parameter(Md=0.0001)
parameter(Hetd=0.000001)
parameter(dt=60)
parameter(omega=1.6)
parameter(TotTimestep=20)
parameter(dx=2.5,dy=dx)
parameter(W1Q=0.004)
parameter(M=10)

parameter(Het1XStart=1)
parameter(Het1XEnd=w1x+10)
```

```
      parameter(Het1YStart=w1y+20)
      parameter(Het1YEnd=w1y+20)
      parameter(Het2XStart=w1x-2)
      parameter(Het2XEnd=length)
      parameter(Het2YStart=w1y+7)
      parameter(Het2YEnd=w1y+7)
      parameter(Het3XStart=w1x-20)
      parameter(Het3XEnd=w1x+25)
      parameter(Het3YStart=w1y-10)
      parameter(Het3YEnd=w1y-10)

      open (unit=UnitNum,file='Turb2D_trans_het.dat')
      open (unit=InfoUnitNum,file='Turb2Dinfo_trans_het.dat')
      open (unit=SeriesUnitNum,file='Turb2Series_trans_het.dat')

      Call CalcCoefs (Md,Ma,Mb,TimeUnit)
      Call CalcCoefs (Hetd,Heta,Hetb,TimeUnit)

      MK=1/Ma

c     _____Set Initial Matrix Values_____

      write(*,*)'Initializing Matrix:',length,length

      h=InitialHead
      h12=InitialHead
      hLast=Initialhead
      Dh=InitialHead
      aa=Ma
      bb=Mb
      d=Md
      Vx=0.0
      Vy=0.0
      DVx=0.0
      DVy=0.0
      Reynolds=0.0
      discX=0.0
      discY=0.0
      Q=0.0

c     _____Determine Radial Constant-Head Boundary Location_____

      if (Radial.eq.1) then
            Radius=sqrt((real(length)/2-1)**2)
```

```
            KBeg=0
            KEnd=0
            do x=1,length
                  tempdist=0
                  do y=1,length
                        Totdist=sqrt((real(w1x)-x)**2+(real(w1y)-y)**2)
                        if ((Totdist.lt.Radius).and.(KBeg(x).eq.0)) then
                              KBeg(x)=y
                        end if

                        if (Totdist.lt.Radius) then
                              KEnd(x)=y
                        end if
                  end do
            end do
      end if

c     __________Insert Heterogeneity Parameters into Matrix__________

      XStart=int(Het1XStart)
      XStart=int(Het1XStart)
      XEnd=int(Het1XEnd)
      YStart=int(Het1YStart)
      YEnd=int(Het1YEnd)
      call Init(aa,length,XStart,XEnd,YStart,YEnd,Heta)
      call Init(bb,length,XStart,XEnd,YStart,YEnd,Hetb)
      call Init(d,length,XStart,XEnd,YStart,YEnd,Hetd)
      XStart=int(Het2XStart)
      XEnd=int(Het2XEnd)
      YStart=int(Het2YStart)
      YEnd=int(Het2YEnd)
      call Init(aa,length,XStart,XEnd,YStart,YEnd,Heta)
      call Init(bb,length,XStart,XEnd,YStart,YEnd,Hetb)
      call Init(d,length,XStart,XEnd,YStart,YEnd,Hetd)
      XStart=int(Het3XStart)
      XEnd=int(Het3XEnd)
      YStart=int(Het3YStart)
      YEnd=int(Het3YEnd)
      call Init(aa,length,XStart,XEnd,YStart,YEnd,Heta)
      call Init(bb,length,XStart,XEnd,YStart,YEnd,Hetb)
      call Init(d,length,XStart,XEnd,YStart,YEnd,Hetd)

c     _______Place Distributed Heterogeneities (Ex. 1 and 2 only)_______
```

```
      if (Spots.eq.1) then
            y=7
            do while(y.lt.length-4)
                  do x=4,length-4,8
                        XStart=x
                        XEnd=x+2
                        YStart=y
                        YEnd=y+2
                        call Init(aa,length,XStart,XEnd,YStart,YEnd,Heta)
                        call Init(bb,length,XStart,XEnd,YStart,YEnd,Hetb)
                        call Init(d,length,XStart,XEnd,YStart,YEnd,Hetd)
                  end do
                  y=y+8
            end do
      end if

      T=(W1Q/dx/M)*(Mb/Ma)**0.5
      C=exp(-0.15*T)
      Q(w1x,w1y)=-W1Q

c     _____Display model parameter values_____

      if (No_Flow.eq.1) then
            BType='NoFlow'
      else
            BType=' Const'
      end if

      write(*,*)'_____'
      write(*,8)'Matrix a:',Ma,'Matrix b:',Mb,'Matrix d:',Md
      write(*,26)'MT:',MK*M,'Tol:',Tol,
     &            'Matrix Head:',InitialHead
      write(*,*)'_____'
      write(*,*)'Beginning Iterations...'

c     _____
c       _____Begin Darcy Iteration Loop_____

      TotTime=0
      do timestep=1,int(TotTimestep)
      TotTime=TotTime+dt
      DMaxError=Tol*100
      ItCounter=0
      DhOld=Dh
```

```
      write(*,*)
      write(*,6)' Timestep:',timestep,'  Turbulent Well Head:',h(w1x,w1y),
     &                  'Darcian Well Head:',Dh(w1x,w1y)

      write(*,*)'_____________________________________________________',


     do while (DMaxError.gt.Tol)
      DMaxError=0
      Itcounter=Itcounter+1
      do x=2,length-2
            do y=KBeg(x),KEnd(x)
                  Dhtemp=Dh(x,y)
                  Call DarcyHead (x,y,Dh,aa,
     &                        Q,dx,dy,dt,Ss,M,Dhtemp,
     &                        length,DhOld,omega)

c                       ....Calculate max change from previous iteration....

                        DErr=abs(Dh(x,y)-Dhtemp)
                        DAbError1=Dh(x,y)-Dhtemp
                        if (DErr .gt. DMaxError) then
                              DMaxError=DErr
                              DAbError2=DAbError1
                        end if

            end do
      end do

            if (No_Flow.eq.1) then
                  Call NoFlow2d (Dh,length)
            end if

c           ....Display iteration statistics and well heads....

      if ((mod(Itcounter,50) .eq.0).or.(ItCounter.eq.1)
     &            .or.(DMaxError.lt.Tol)) then
                  Call DisplayIterStats2d (h,Dh,length,ItCounter,
     &                        AbError2,DAbError2,w1x,w1y)


      end if
      end do
```

```
c          ____________________________________________________________
c            ______________Begin Turb Iteration Loop______________
          if (timestep.eq.1) then
                    h=Dh
                    hLast=h
          end if
          MaxError=Tol*100
          ItCounter=0
          h12=(h+hLast)/2
          hLast=h

          do while (MaxError.gt.Tol)
                    Itcounter=Itcounter+1
                    MaxError=0
                    do x=2,length-2
                              do y=KBeg(x),KEnd(x)
                                        hOld=h(x,y)
                                        Call CalcHead2d (x,y,h,aa,bb,
    &                                                   Q,dx,dy,M,dt,Ss,h12,
    &                                                   length,hOld,hLast,omega,C)

c                             ....Calculate max change from previous iteration....

                                                        Err=abs(h(x,y)-hOld)
                                                        AbError1=h(x,y)-hOld
                                                        if (Err .gt. MaxError) then
                                                                  MaxError=Err
                                                                  AbError2=AbError1
                                                        end if
                              end do
                    end do

                     if (No_Flow.eq.1) then
                              Call NoFlow2d (h,length)
                     end if

c                      ....Display iteration statistics and well heads....

          if ((mod(Itcounter,50) .eq.0).or.(ItCounter.eq.1)
    &               .or.(MaxError.lt.Tol)) then
                              Call DisplayIterStats2d (h,Dh,length,ItCounter,
    &                                           AbError2,DAbError2,w1x,w1y)
```

```
      end if

c     _____Reverse Forchheimer-based Iteration_____

      Itcounter=Itcounter+1
      MaxError=0
      do x=length-2,2,-1
            do y=KEnd(x),KBeg(x),-1

                  hOld=h(x,y)
                  Call CalcHead2d (x,y,h,aa,bb,
     &                        Q,dx,dy,M,dt,Ss,h12,
     &                        length,hOld,hLast,omega,C)

c                 ....Calculate max change from previous iteration....

                                    Err=abs(h(x,y)-hOld)
                                    AbError1=h(x,y)-hOld
                                    if (Err .gt. MaxError) then
                                          MaxError=Err
                                          AbError2=AbError1
                                    end if
            end do
      end do

            if (No_Flow.eq.1) then
                  Call NoFlow2d (h,length)
            end if

c           ....Display iteration statistics and well heads....

      if ((mod(Itcounter,50) .eq.0).or.(ItCounter.eq.1)
     &      .or.(MaxError.lt.Tol)) then
            Call DisplayIterStats2d (h,Dh,length,ItCounter,
     &                  AbError2,DAbError2,w1x,w1y)
      end if


      end do
      end do

c     _____End Timestep/Iteration Loop_____
```

```
c           _____Calculate Velocities and Reynolds Numbers_____

      write(*,*)
      write(*,*)'Calculating Velocities and Reynolds Numbers...'
      do x=2,length-1
            do y=2,length-1
                  Call Velocity2d (x,y,h,Dh,aa,bb,Vx,Vy,
     &                        DVx,DVy,MagV,dx,dy,length)

                  Reynolds(x,y)=abs(Rho*sqrt(Vx(x,y)**2
     &                        +Vy(x,y)**2)*d(x,y)/Mu)
                  if (Q(x,y).ne.0) then
                        dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
                        dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
                        dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
                        dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
                        dhdx=(dhdx1+dhdx2)/2
                        dhdy=(dhdy1+dhdy2)/2
                  else
                        dhdx=abs(h(x+1,y)-h(x-1,y))/(2*dx)
                        dhdy=abs(h(x,y+1)-h(x,y-1))/(2*dx)
                  end if
                  a=aa(x,y)
                  b=bb(x,y)
                  discX(x,y)=sqrt(a*a+4*b*dhdx)
                  discY(x,y)=sqrt(a*a+4*b*dhdy)
            end do
      end do

c           _____Write Results to Screen_____

      write(*,*)'_____'
      write(*,8)'Matrix a:',Ma,'Matrix b:',Mb,'Matrix d:',Md
      write(*,26)'MT:',MK*M,'Tol:',Tol,'Matrix Head:',InitialHead
      write(*,*)'_____'
      write(*,*)'Delta X:',dx,'M:',M
      write(*,*)'T=',T
      write(*,*)'C=',C
      write(*,*)'Total Simulation Time: ',TotTime
      write(*,*)'_____'
      write(*,*)

c     _____Write cross-section head values to series file_____
```

```
      do y=length/2,length-1
            dhdy=abs(h(w1x,y-1)-h(w1x,y))/dx
        write(12,30)y,h(w1x,y),Dh(w1x,y),dhdy,h(w1x,y)-Dh(w1x,y)
      end do

      close(unit=SeriesUnitNum)
30    format(I15,F15.5,F15.5,F15.5,F15.5)

c     _____Write full 2D grid to data file_____

      write(*,*)'Writing simulation to file...'
      Call WriteGrid (h,Dh,aa,bb,d,Reynolds,Vx,Vy,DVx,DVy,discX,discY,
     &                dx,dy,length,UnitNum)
      close(unit=UnitNum)

c     _____Write simulation info to data file_____

      Call SimInfo (Md,Ma,Mb,Hetd,Heta,Hetb,InitialHead,W1Q,
     &              C,dx,Tol,M,h,Dh,BType,dt,TotTime,
     &              w1x,w1y,length,InfoUnitNum)
      close(unit=InfoUnitNum)

c     _____Output Format List_____

6     format(A,I4,A25,F8.3,A23,F8.3)
8     format(A10,G11.4,A11,G11.4,A11,F11.8)
9     format(A14,F12.5,A14,F12.3,A8)
25    format(A10,G13.2,G13.2,A6)
26    format(A10,F10.4,A10,F10.7,A15,F8.2)
      Call beepQQ(1000,230)
      Call beepQQ(500,130)
      stop
      end
```

A4.2 FORCHHEIMER-BASED HEAD CALCULATION

```
c     ************************************************************
c      Calculates finite-difference node head values based on
c      the Forchheimer relation
c     ************************************************************

      Subroutine CalcHead2d (x,y,h,aa,bb,Q,dx,dy,M,dt,Ss,
```

```
&                                h12,length,hOld,hLast,omega,C)

      implicit none
      integer x,y,length,test,x0,y0
      real(8) h(length,length)
      real(8) h12(length,length)
      real(8) hLast(length,length)
      real(8) aa(length,length)
      real(8) bb(length,length)
      real(8) Q(length,length)
      real(8) dx,dy,dt,M,C,Ss,hOld,omega
      real(8) dhdx,dhdy,dhdx1,dhdx2,dhdy1,dhdy2
      real(8) discrimx,discrimy,discrimx1,discrimx2,discrimy1,discrimy2
      real(8) a,b,ax1,ax2,bx1,bx2,ay1,ay2,by1,by2
      real(8) discx_xp2,discx_xp1,discx_xm2,discx_xm1
      real(8) discy_xp2,discy_xp1,discy_xm2,discy_xm1
      real(8) discx_yp2,discx_yp1,discx_ym2,discx_ym1
      real(8) discy_yp2,discy_yp1,discy_ym2,discy_ym1
      real(8) T1,T2,T3,T4,T5

      a=aa(x,y)
      b=bb(x,y)
      ax1=(aa(x+1,y)+a)/2
      bx1=(bb(x+1,y)+b)/2
      ax2=(aa(x-1,y)+a)/2
      bx2=(bb(x-1,y)+b)/2
      ay1=(aa(x,y+1)+a)/2
      by1=(bb(x,y+1)+b)/2
      ay2=(aa(x,y-1)+a)/2
      by2=(bb(x,y-1)+b)/2

      dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
      dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
      dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
      dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
      discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
      discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
      discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
      discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)

      x0=x
      y0=y
```

```
if (Q(x,y).ne.0) then
        x=x0+2
        y=y0
        a=aa(x,y)
        b=bb(x,y)
        ax1=(aa(x+1,y)+a)/2
        bx1=(bb(x+1,y)+b)/2
        ax2=(aa(x-1,y)+a)/2
        bx2=(bb(x-1,y)+b)/2
        ay1=(aa(x,y+1)+a)/2
        by1=(bb(x,y+1)+b)/2
        ay2=(aa(x,y-1)+a)/2
        by2=(bb(x,y-1)+b)/2
        dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
        dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
        dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
        dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
        discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
        discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
        discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
        discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
        discx_xp2=(discrimx1+discrimx2)/2
        discy_xp2=(discrimy1+discrimy2)/2

        x=x0+1
        y=y0
        a=aa(x,y)
        b=bb(x,y)
        ax1=(aa(x+1,y)+a)/2
        bx1=(bb(x+1,y)+b)/2
        ax2=(aa(x-1,y)+a)/2
        bx2=(bb(x-1,y)+b)/2
        ay1=(aa(x,y+1)+a)/2
        by1=(bb(x,y+1)+b)/2
        ay2=(aa(x,y-1)+a)/2
        by2=(bb(x,y-1)+b)/2
        dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
        dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
        dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
        dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
        discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
        discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
        discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
        discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
```

```
discx_xp1=(discrimx1+discrimx2)/2
discy_xp1=(discrimy1+discrimy2)/2

x=x0-1
y=y0
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
discx_xm1=(discrimx1+discrimx2)/2
discy_xm1=(discrimy1+discrimy2)/2

x=x0-2
y=y0
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
```

```
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
discx_xm2=(discrimx1+discrimx2)/2
discy_xm2=(discrimy1+discrimy2)/2

x=x0
y=y0+2
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
discx_yp2=(discrimx1+discrimx2)/2
discy_yp2=(discrimy1+discrimy2)/2

x=x0
y=y0+1
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
```

```
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
discx_yp1=(discrimx1+discrimx2)/2
discy_yp1=(discrimy1+discrimy2)/2

x=x0
y=y0-1
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
discx_ym1=(discrimx1+discrimx2)/2
discy_ym1=(discrimy1+discrimy2)/2

x=x0
y=y0-2
a=aa(x,y)
b=bb(x,y)
ax1=(aa(x+1,y)+a)/2
bx1=(bb(x+1,y)+b)/2
ax2=(aa(x-1,y)+a)/2
bx2=(bb(x-1,y)+b)/2
ay1=(aa(x,y+1)+a)/2
by1=(bb(x,y+1)+b)/2
ay2=(aa(x,y-1)+a)/2
by2=(bb(x,y-1)+b)/2
dhdx1=abs(h12(x+1,y)-h12(x,y))/dx
dhdx2=abs(h12(x,y)-h12(x-1,y))/dx
```

```
        dhdy1=abs(h12(x,y+1)-h12(x,y))/dy
        dhdy2=abs(h12(x,y)-h12(x,y-1))/dy
        discrimx1=sqrt(ax1*ax1+4*bx1*dhdx1)
        discrimx2=sqrt(ax2*ax2+4*bx2*dhdx2)
        discrimy1=sqrt(ay1*ay1+4*by1*dhdy1)
        discrimy2=sqrt(ay2*ay2+4*by2*dhdy2)
        discx_ym2=(discrimx1+discrimx2)/2
        discy_ym2=(discrimy1+discrimy2)/2

        discrimx1=(discx_xp1+discx_xm1+discx_yp1+discx_ym1)/4
        discrimx2=(discx_xp2+discx_xm2+discx_yp2+discx_ym2)/4
        discrimy1=(discy_xp1+discy_xm1+discy_yp1+discy_ym1)/4
        discrimy2=(discy_xp2+discy_xm2+discy_yp2+discy_ym2)/4
        discrimx=2/(C/discrimx1+(2-C)/discrimx2)
        discrimy=2/(C/discrimy1+(2-C)/discrimy2)
        discrimx1=discrimx
        discrimx2=discrimx
        discrimy1=discrimy
        discrimy2=discrimy
end if

discrimx=(discrimx1+discrimx2)/2
discrimy=(discrimy1+discrimy2)/2
x=x0
y=y0
T1=h(x+1,y)/discrimx+h(x-1,y)/discrimx
T2=h(x,y+1)/discrimy+h(x,y-1)/discrimy
T3=Q(x,y)/M
T4=dx*dx*Ss*hLast(x,y)/dt
T5=1/(Ss*dx*dx/dt+2/discrimx+2/discrimy)
h(x,y)=(T1+T2+T3+T4)*T5
h(x,y)=omega*h(x,y)+(1-omega)*hOld

return
end
```

A4.3 DARCY-BASED HEAD CALCULATION

```
c     ************************************************************
c      Calculates finite-difference node head values based on
c      Darcy's Law.
c     ************************************************************


      Subroutine DarcyHead (x,y,Dh,aa,Q,dx,dy,dt,Ss,M,
     &                      Dhtemp,length,DhOld,omega)

      implicit none
      integer x,y,length,test
      real(8) Dh(length,length)
      real(8) DhOld(length,length)
      real(8) aa(length,length)
      real(8) Q(length,length)
      real(8) dx,dy,M
      real(8) omega,dt,Ss
      real(8) a,ax1,ax2,ay1,ay2,ax,ay,Dhtemp
      real(8) D1,D2,D3,D4

      a=aa(x,y)
      ax1=(aa(x+1,y)+a)/2
      ax2=(aa(x-1,y)+a)/2
      ay1=(aa(x,y+1)+a)/2
      ay2=(aa(x,y-1)+a)/2
      ax=(ax1+ax2)/2
      ay=(ay1+ay2)/2

      D1=Dh(x+1,y)/ax+Dh(x-1,y)/ax+Dh(x,y+1)/ay+Dh(x,y-1)/ay
      D2=Ss*dx*dx*DhOld(x,y)/dt
      D3=Q(x,y)/M
      D4=1/(Ss*dx*dx/dt+2/ax+2/ay)
      Dh(x,y)=(D1+D2+D3)*D4
      Dh(x,y)=omega*Dh(x,y)+(1-omega)*Dhtemp

      return
      end
```

A4.4 WRITE SIMULATION INFORMATION

```
c     ************************************************************
c     Writes simulation parameters and information to a text file.
c     ************************************************************

      Subroutine SimInfo (Md,Ma,Mb,Hetd,Heta,Hetb,InitialHead,W1Q,
     &                    C,dx,Tol,M,h,Dh,BType,dt,TotTime,
     &                    w1x,w1y,length,InfoUnitNum)

      implicit none
      integer w1x,w1y,length,InfoUnitNum
      real(8) Ma,Mb,Md,MK
      real(8) Hetd,Heta,Hetb
      real(8) InitialHead,W1Q
      real(8) C,dx,Tol,M,dt,TotTime
      real(8) h(length,length)
      real(8) Dh(length,length)
      character *6 BType

      MK=1/Ma
      write(InfoUnitNum,14)' Md ','Ma ','Mb ','Hetd ','Heta ','Hetb ',
     &                     'M ','dx ','InitialHead ','Q ','C ','dt ','TotTime ',
     &                     'TWellHead ','DWellHead ',
     &                     'length ','Tol ','Boundary '

      write(InfoUnitNum,15)Md,Ma,Mb,Hetd,Heta,Hetb,M,dx,
     &                  InitialHead,W1Q,C,dt,TotTime,h(w1x,w1y),
     &                  Dh(w1x,w1y),length,Tol,BType

14    format (A,A15,A15,A15,A15,A15,A15,A15,A15,A15,A15,
     &        A15,A15,A15,A15,A15,A15,A15,A15)

15    format (F15.7,F15.5,F15.5,F15.5,F15.5,F15.5,F15.3,F15.3,
     &        F15.3,F15.3,F15.4,F15.3,F15.3,F15.3,F15.3,I15,F15.5,A7)
      return
      end
```

Appendix 5. Nomenclature

A = cross-sectional area of flow. [L^2]

a = hydraulic resistivity (Irmay 1958) [t/L]

B = coefficient of drawdown due to aquifer effects [t/L^2]

b = nonlinear resistivity coefficient [t^2L^{-2}]

C = coefficient of drawdown caused by flow through the well screen and flow inside the well bore. [units variable]

d = average grain diameter [L]

dh/dL = hydraulic gradient over some distance L [-]

g = acceleration due to gravity [Lt^{-2}]

h = hydraulic head (elevation + pressure/ ρg) [L]

K = hydraulic conductivity. [L/t]

M = aquifer thickness [L]

Q = discharge (volumetric rate of flow) [L^3/t]

q = specific discharge (Q/A) [Lt^{-1}]

R = Reynolds number [-]

S_s = specific storage $\left(\rho g(\alpha + \beta\phi)\right)$ [L^{-1}]

s_{well} = drawdown at the well screen [L]

t = time [t]

α = compressibility of the matrix [ML^3t^{-2}]

β = compressibility of the fluid [ML^3t^{-2}]

Δx = nodal spacing [L]

μ = viscosity [$Mt^{-1}L^{-1}$]

ϕ = porosity of the aquifer matrix [-]

ρ = fluid density [ML^{-3}]

References

Abou-Kassem, J.H. and Aziz, K., Analytical Well Models for Reservoir Simulation. *Society of Petroleum Engineers Journal*, (August), p. 573-579, 1985.

Basak, P., Steady Non-Darcian Seepage through Embankments. *Journal of the Irrigation and Drainage Division*, December (IR4), p. 435-443.

Basak, P. and Rajogopalan, S.P., Effect of Non-Darcy Flow on Computing Seawater Intrusion Lengths. *Journal of Hydrology*, v. 58, p. 83-87, 1982.

Bowen, Robert, *Groundwater*, 2nd ed., New York : Elsevier Applied Science Publishers, p. 427, 1986

Calhoun, John C., *Fundamentals of Reservoir Engineering*. Rev. ed., Norman : University of Oklahoma Press, p. 426, 1960.

Darcy, H., *Les Fontaines Publiques de la Villa da Dijon.* Victor Dalmont, Paris, 1856.

Domenico, P. A. And Schwartz, Franklin W., *Physical and Chemical Hydrogeology*. p. 824, 1990.

Fetter, C.W., Applied Hydrogeology, 3rd ed., New Jersey : Prentice Hall, p. 691, 1988.

Forchheimer, P.H., Wasseerbewegung Durch Boden. *Zeitschrift ver Deutsch Ingenieur*, (45), 1901.

Gill, M.A., Analysis of Permeameter Data to Obtain Parameters of Non-Darcy Flow. *Journal of Hydrology*, v. 32, p. 165-173, 1977.

Halihan, T., *Personal Communication*, September, 1999.

Hansen, D., Garga, V.K. and Townsend, D.R., Selection and Application of a One-dimensional Non-Darcy Flow Equation for Two-dimensional Flow through Rockfill Embankments. *Canadian Geotechnical Journal (Revue Canadienne de Geotechnique),* v. 33 (1), p. 199-201, 1996.

Hayes, L.J., Kendall, R.P. and Wheeler, M.F., The Treatment of Sources and Sinks in Steady-state Reservoir Engineering Simulations. Vichnevetsky, R. (editor), *Advances in Computer Methods for Partial Differential Equations II*: proceedings of the second IMACS (AICA) International Symposium on Computer Methods for Partial Differential Equations, held at Lehigh University, Bethlehem, Pennsylvania, U.S.A., June 22-24, 1977.

Irmay, S., On the Theoretical Derivation of Darcy and Forchheimer Formulas. *Transactions, American Geophysical Union,* vol. 39 (4), p. 702-707, 1958.

Jacob, C.E., Drawdown Test to Determine Effective Radius of Artesian Wells. *Transactions of the American Society of Civil Engineers*, v. 112, p. 1047-1070, 1947.

Lage, J.L., The Fundamental Theory of Flow through Permeable Media from Darcy to Turbulence. Ingham, Derek B. (editor), Pop, Ioan (editor), *Transport Phenomena in Porous Media*. Danvers, MA : Pergamon, p. 1-30, 1998.

Lee, R.L., Logan, R.W. and Tek, M.R., Effect of Turbulence on Transient Flow of Real Gas through Porous Media. *Society of Petroleum Engineers (SPE): Formation Evaluation*, (March), p. 108-120, 1987.

Peaceman, D.W., Interpretation of Well-Block Pressures in Numerical Reservoir Simulation. *Society of Petroleum Engineers Journal*, (June), p. 183-194, 1978.

Peaceman, D.W., Interpretation of Well-Block Pressures in Numerical Reservoir Simulation: Part 3 -- Off-Center and Multiple Wells Within a Wellblock. *Society of Petroleum Engineers (SPE): Reservoir Engineering,* (May), p. 227-232, 1990.

Quinney, D., *An Introduction to the Numerical Solution of Differential Equations*. New York : Wiley, p. 283, 1985.

Rama Rao, B.S. and Das, R.N., A Study of Unconfined Non-Darcy Seepage to a Well. *Journal of Hydrology*, v.38, p.161-178, 1978.

Rorabaugh, M.I., Graphical and Theoretical Analysis of Step-Drawdown Test of Artesian Well. *Proceedings of the American Society of Civil Engineers*, v. 79, sep. 362, p. 23, 1953.

Sen, Z., Volumetric Approach to Non-Darcy Flow in Confined Aquifers. *Journal of Hydrology*, v. 87, p.337-350, 1986.

Sen, Z., Non-Darcy Flow in Fractured Rocks with a Linear Flow Pattern. *Journal of Hydrology*, v. 92, p.43-57, 1987.

Sen, Z., Analytical Solution Incorporating Nonlinear Radial Flow in Confined Aquifers. *Water Resources Research*, v. 24 (4), p. 601-606, 1988.

Subramanya, K. And Madhav, M.R., Linear amd Non-linear Flows Through Porous Media. Tech. Rep. No. 22, *Central Board of Irrigation and Power,* New Delhi, India, 1978.

Theim, G., *Hydrologische Methoden*. Leipzig, Gebhardt, p. 56, 1906.

Tiss, M. and Evans, R.D., Measurement and Correlation of Non--Darcy Flow Coefficient in Consolidated Porous Media. *Journal of Petroleum Science and Engineering*, v. 3, p.19-33, 1989.

Todd, David K., *Groundwater Hydrology*. 2nd ed., New York : Wiley, p. 535, 1959.

Venkataraman, P. and Rama Mohan Rao, P., Darcian, Transitional and Turbulent Flow through Porous Media. *Journal of Hydraulic Engineering,* (August), p. 840-846, 1998.

Volker, R.E., Solutions for Unconfined Non-Darcy Seepage. *Journal of the Irrigation and Drainage Division*, March (IR1), p. 53-65, 1975.

Von Rosenberg, Dale U., Methods for the Numerical Solution of Partial Differential Equations. Bellman, R.(editor), *Modern Analytic and Computational Methods in Science and Mathematics,* vol. 16, p. 128, illus., 1969.

Wang, H.F., Anderson, M.P., *Introduction to Groundwater Modeling: Finite Difference and Finite Element Methods*. San Francisco : W.H. Freeman, p.237, 1982.

Williamson, A.S. and Chappelear, J.E., Representing Wells in Numerical Reservoir Simulation: Part 1 -- Theory. *Society of Petroleum Engineers Journal,* (June), p.323-338, 1981.

Williamson, A.S. and Chappelear, J.E., Representing Wells in Numerical Reservoir Simulation: Part 2 -- Implementation. *Society of Petroleum Engineers Journal,* (June), p.339-344, 1981.

The vita has been removed from the digitized version of this document.