Use of Picard and Newton Iteration for Solving Nonlinear Ground Water Flow Equations
by Steffen Mehl

Abstract
This study examines the use of Picard and Newton iteration to solve the nonlinear, saturated ground water flow equation. Here, a simple three-node problem is used to demonstrate the convergence difficulties that can arise when solving the nonlinear, saturated ground water flow equation in both homogeneous and heterogeneous systems with and without nonlinear boundary conditions. For these cases, the characteristic types of convergence patterns are examined. Viewing these convergence patterns as orbits of an attractor in a dynamical system provides further insight. It is shown that the nonlinearity that arises from nonlinear head-dependent boundary conditions can cause more convergence difficulties than the nonlinearity that arises from flow in an unconfined aquifer. Furthermore, the effects of damping on both convergence and convergence rate are investigated. It is shown that no single strategy is effective for all problems and how understanding pitfalls and merits of several methods can be helpful in overcoming convergence difficulties. Results show that Picard iterations can be a simple and effective method for the solution of nonlinear, saturated ground water flow problems.

Introduction
Addressing the complex problems faced by hydrogeologists requires versatile models. In addition to the hydrological concerns, hydrogeologists often have to consider economic, societal, and ecological issues as well. Thus, capabilities of ground water models have evolved to deal with complicated processes that connect various components of the hydrological system. Furthermore, increased computer capabilities have allowed hydrogeologists to represent processes in their models with improved accuracy by using greater spatial and temporal resolutions and better approximations to phenomena that were previously simplified or ignored. Incorporating these processes into a numerical model often produces a system of nonlinear equations. The improved capabilities are substantially hindered if (1) data are not available to characterize the more complicated systems (Halford 2004) or (2) solutions cannot be obtained for these complicated models. There is perhaps nothing more frustrating to the practicing hydrogeologist than spending many hours building a complex, numerical ground water model only to spend many more hours trying to get the solver to produce a suitable solution. Solving linear systems of equations can be difficult. Solving nonlinear systems of equations is generally much more challenging. The convergence problems often arise from nonlinear phenomena typically present in complicated ground water models, a common but unfortunate situation in that the very things that make our models better in the sense that physics are better represented often make them worse in the sense that they are more difficult to solve numerically.

Techniques for solving systems of nonlinear equations have been investigated in general (for example, Ortega and Rheinboldt 1970; Dennis and Schnabel 1996) and in the context of ground water modeling (for example, Cooley 1983; Kuiper 1987; Paniconi and Putti 1994; Farthing et al. 2003). Typically, the methods used are iterative in that they approach the solution through a series of steps. Of concern is both the algorithm used within each step and how the results of one or more steps are used to proceed to the next step. Newton-type methods (also known as gradient-based methods) are generally regarded...
as being very efficient, but Picard iterations (also known as successive substitution) continue to be used in many numerical models, such as MODFLOW (Harbaugh et al. 2000). The reasons for this are both the simplicity of Picard iterations and that available comparisons to Newton-type methods for saturated ground water flow equations (see Kuiper 1987) suggest that Picard iterations generally perform well. Newton-type methods require either evaluation of the Jacobian or an approximation of the Jacobian (the derivative of the system of equations with respect to heads). Depending on how Newton-type methods are implemented, they can result in a nonsymmetric coefficient matrix that requires more storage and specialized linear solvers compared to symmetric coefficient matrices that result from discretization of the ground water flow equation. Conversely, Picard iterations directly use the standard coefficient matrix and therefore are easier to implement (Paniconi and Putti 1994). The drawback of Picard iterations is their linear rate of convergence, while Newton-type methods converge superlinearly or quadratically. Thus, a trade-off exists of more iterations of a cheaper method or fewer iterations of a more expensive method.

However, perhaps the most important issue is not the rate (order) of convergence, rather the ability to converge as indicated by Ralston and Rabinowitz (2001, 360): “But with systems of [nonlinear] equations convergence itself is such a serious problem that usually we shall be satisfied with any order of convergence if only the method will converge.” This issue is related to the radius of convergence and basin of attraction. For initial guesses within this basin, and this region is not necessarily continuous, the method will converge. Techniques that increase this region increase the robustness of the Picard or Newton iteration so that initial guesses that are poor may still converge. Some of these techniques will be examined closely and their strengths and weaknesses discussed.

To examine the characteristic convergence patterns that result from solving the nonlinear, saturated ground water flow equation using both Picard and Newton iteration, this study first uses a simple three-node problem, solved using a spreadsheet. The three-node problem allows clear demonstration of typical convergence difficulties. Analogies from attractors of dynamical systems are used to provide further insight into the convergence patterns of the Picard iteration, and examples are shown of the three main patterns: cobweb, staircase, and periodic attractor. Understanding the characteristics of these convergence patterns can help develop strategies to overcome convergence difficulties. For the first two examples, results of Newton’s method also are shown and used to demonstrate that, while Newton’s method converges more rapidly, it often requires a better initial guess than Picard’s method in order to converge at all. Furthermore, it is shown that the nonlinearities that arise from nonlinear (piecewise linear), head-dependent boundary conditions can cause more difficulties than the nonlinearity associated with flow in an unconfined aquifer. But the combination of unconfined flow, piecewise linear, head-dependent boundary conditions, and heterogeneity make obtaining a viable solution more difficult. Marching transient solutions toward a steady-state solution and damping (relaxation) also are examined as methods for improving the ability to converge and the convergence rate.

While the extension is not obvious from a simple three-node problem to the complex three-dimensional problems that occur in practice, many of the general features that occur in the simple problem will manifest in more complicated problems. This will be demonstrated using a complicated three-dimensional MODFLOW model where the importance of damping is investigated.

This work uses the results based on the convergence patterns of several examples to analyze, discuss, and describe strategies for alleviating some of the convergence difficulties associated with nonlinear, saturated ground water flow problems.

A Three-Node Model

A schematic of the three-node system is shown in Figure 1. Specified-head boundaries are used on the left and right nodes, and the cells have a uniform spacing and unit width. Additional considerations include unconfined conditions and evapotranspiration. Here, the equations applicable to these conditions are presented. This three-node system requires a solution only at the central node.

Performing a steady-state mass balance on the central cell results in

\[ Q_L - Q_R - ET = 0 \]  

(1)

where, \( Q_L \) = flow per unit width into the central cell through the left face, \( Q_R \) = flow per unit width out of the central cell through the right face, \( ET \) = evapotranspiration per unit width out of the central cell.

The flow across the cell faces can be expressed in terms of conductances and head differences, as

\[ Q_L = C_L (h_L - h) \]  

(2a)

\[ Q_R = C_R (h - h_R) \]  

(2b)

where, \( C_L \) = the conductance of the aquifer material between the left node and the central node, \( C_R \) = the conductance of the aquifer material between the right node

\[ ET = \text{evapotranspiration} \]

\[ h_L = \text{specified head} \]

\[ h_R = \text{specified head} \]

Figure 1. Schematic of a three-node block-centered model with uniform grid spacing and specified head boundaries at the left and right nodes (\( h_L \) and \( h_R \), respectively). Evapotranspiration from the central cell varies piecewise linearly with water table elevation from zero at the extinction depth to the maximum rate at the evaporation surface.
and the central node, \( h_L \) = head specified as a boundary condition at the left node, \( h_R \) = head specified as a boundary condition at the right node, \( h \) = head at the central node (unknown).

For a confined aquifer, conductance for a unit width is calculated as

\[
C_L = \frac{(2 \cdot K_L \cdot b_L \cdot K \cdot b)}{(K_L \cdot b_L \cdot \Delta x + K \cdot b \cdot \Delta x)}
\]  
(3a)

\[
C_R = \frac{(2 \cdot K_R \cdot b_R \cdot K \cdot b)}{(K_R \cdot b_R \cdot \Delta x + K \cdot b \cdot \Delta x)}
\]  
(3b)

For an unconfined aquifer, conductance is a function of head and for a unit width is calculated as

\[
C_L = \frac{(2 \cdot K_L \cdot h_L \cdot K \cdot h)}{(K_L \cdot h_L \cdot \Delta x + K \cdot h \cdot \Delta x)}
\]  
(3c)

\[
C_R = \frac{(2 \cdot K_R \cdot h_R \cdot K \cdot h)}{(K_R \cdot h_R \cdot \Delta x + K \cdot h \cdot \Delta x)}
\]  
(3d)

where \( K_L \) = the hydraulic conductivity of the left cell, \( K_R \) = the hydraulic conductivity of the right cell, \( K \) = the hydraulic conductivity of the central cell, \( b_L \) = the saturated thickness of the left cell, \( b_R \) = the saturated thickness of the right cell, \( b \) = the saturated thickness of the central cell, \( \Delta x \) = the grid spacing or cell dimension.

The only difference between the confined and unconfined conductances is the saturated thickness; in the unconfined case, it is equal to the head in the aquifer. Thus, in the unconfined case, the datum for head is the bottom of the cell.

A common representation for evapotranspiration is a piecewise-linear function that varies linearly from the maximum rate (\( ET_{\text{max}} \)) to zero with water table level (McDonald and Harbaugh 1988; Bredehoef 2002).

\[
ET = \begin{cases} 
ET_{\text{max}} & \text{if } h > h_s \\
ET_{\text{max}}[h - (h_s - d)]/d & \text{if } h_s - d < h < h_s \\
0 & \text{if } h < h_s - d 
\end{cases}
\]  
(4a)

where \( h_s \) = the evapotranspiration surface elevation where maximum evapotranspiration applies, \( d \) = the distance below \( h_s \) where evapotranspiration ceases (hereafter called extinction depth).

Writing Equation 1 in terms of Equations 2 and 4 and gathering terms that multiply \( h \) results in

\[
h(-C_L - C_R) = -C_L \cdot h_L - C_R \cdot h_R + ET_{\text{max}}
\]  
(5a)

\[
h(-C_L - C_R - ET_{\text{max}}/d) = -C_L \cdot h_L - C_R \cdot h_R - ET_{\text{max}}(h_s - d)/d
\]  
(5b)

\[
h(-C_L - C_R) = -C_L \cdot h_L - C_R \cdot h_R
\]  
(5c)

where Equations 5a, 5b, and 5c are derived from Equation 4a, 4b, and 4c, respectively. This single equation for \( h \) is completely analogous to the system of equations (often written in matrix form) that is typically formulated (for example, McDonald and Harbaugh 1988 2–26)—the left-hand side contains conductance terms that connect the node of interest to the neighboring nodes as well as head-dependent boundary conditions. The right-hand side contains terms related to boundary conditions and sinks/sources.

Solving for \( h \) results in

\[
h = \frac{-C_L \cdot h_L - C_R \cdot h_R + ET_{\text{max}}}{(-C_L - C_R)} = f_p(h)
\]  
(6a)

\[
h = \frac{-C_L \cdot h_L - C_R \cdot h_R - ET_{\text{max}}(h_s - d)/d}{(-C_L - C_R - ET_{\text{max}}/d)} = f_p(h)
\]  
(6b)

\[
h = \frac{-C_L \cdot h_L - C_R \cdot h_R}{(-C_L - C_R)} = f_p(h)
\]  
(6c)

Equation 6 has two types of nonlinearity arising from the head, \( h \), being a function of terms that are themselves functions of head, \( f_p(h) \) (the subscript "p" is used to indicate this is the functional form used in the Picard iteration). The first type of nonlinearity arises from the evapotranspiration term. Depending on the head at the central node, Equation 6a, 6b, or 6c is applicable. However, we do not know the correct head solution at the central node yet, so we do not know which equation applies. The second type of nonlinearity is related to the conductance terms. For unconfined aquifers, the conductances depend on the head, as shown in Equations 3c and 3d. Again, because we do not know the correct head yet, we do not know the correct value of conductance. Generally, iterative methods are used to solve nonlinear equations, and two typical approaches, Picard and Newton iterations, are discussed.

**Picard Iteration**

Picard iteration is a simple method for dealing with the nonlinearities in Equation 6. The method proceeds as follows for the three-node system: guess a value for head, calculate conductances using Equation 3, calculate the evapotranspiration using the appropriate parts of Equation 4, and solve for a new value of head using Equation 6. If this new value of head matches (to some tolerance) the value of head that started this iteration, then stop (convergence was achieved). If not, use this new value of head to update the nonlinear terms (conductances of Equation 3 and evapotranspiration of Equation 4) and solve Equation 6 again. That is, continuously substitute new values of head into the equation to solve for yet newer values of head. The procedure stops when convergence is achieved (there is little difference between old and new heads) or a maximum iteration count is reached.

In general numerical ground water flow models, Picard iterations are implemented in a similar fashion. Using the current head values, the coefficient matrix (left-hand side) and the right-hand side vector are updated. The resulting system of equations is solved with a linear solver to obtain new head values, and the procedure continues as before. This procedure is used in MODFLOW-2000.
A Closer Look at Picard Iteration

In the context of dynamical systems, the Picard iteration defined in Equation 6 can be viewed as a mapping rule. By inputting a value of head on the right-hand side, we get a value of head on the left-hand side. Therefore, when given a value of head, Equation 6 maps it to a new value of head. Tracing the sequence of heads that are generated by successively applying this rule results in an orbit (Alligood et al. 1996). This can be depicted graphically, as shown in Figure 2, for the three-node problem without evapotranspiration. In Figure 2, starting at an initial guess of \( h = 0.10 \) (point A), the value of the right-hand side of Equation 6 is \( f_p(h) = 1.3 \) (point B). Because \( h = f_p(h) \), this new value of \( h \) from the left-hand side is substituted into the right-hand side of Equation 6 to prepare for the next iteration. This maps the solution through an orbit shown in Figure 2. In this case, the orbit is attracted to a fixed point and eventually ends up there. The fixed point is the solution to the nonlinear ground water flow equation and corresponds to the intersection of the graphs of \( h = h \) and \( h = f_p(h) \), where \( f_p(h) \) is defined by the right-hand side of Equation 6. At this point, the value of head input on the right-hand side of Equation 6 and the value output on the left-hand side are the same.

The orbit shown in Figure 2 is the convergence pattern that results from using Picard iterations to solve the nonlinear ground water flow equation. This type of convergence pattern is typical of Picard iterations and often is called a “cobweb plot.” It is characterized by the direction of head change alternating with each iteration. The head is successively over- and underestimated by decreasing amounts until it converges to the solution. That is, it overshoots the solution at each iteration, but the amount of overshoot decreases with each iteration. As shown in Figure 2, all starting heads greater than zero converge to the solution. Starting heads less than zero are physically unrealistic.

Newton Iteration

Newton’s method is used to find a solution of nonlinear equations. It is based on finding the root of equations. The three-node example can be formulated into a root-finding problem by subtracting the right-hand side of Equation 5 from both sides of the equation. This results in

\[
\begin{align*}
f_p(h) &= h(-C_L - C_R) + C_L \cdot h_L + C_R \cdot h_R - ET_{\text{max}} \\
f_p(h) &= h(-C_L - C_R - ET_{\text{max}}/d) + C_L \cdot h_L + C_R \cdot h_R + ET_{\text{max}}(h_i - d)/d \\
f_p(h) &= h(-C_L - C_R) + C_L \cdot h_L + C_R \cdot h_R
\end{align*}
\]

Newton’s method finds the value of \( h \) that results in \( f_p(h) = 0 \). One begins by guessing a value and calculating the derivative of the equation—slope of the line tangent to the curve—at this point. Following the tangent line until it crosses zero provides an approximation of the root. The value of \( h \) where the tangent crosses zero is taken as the updated value, and the procedure repeats until convergence is achieved or a maximum iteration count is reached. This procedure is shown in Figure 3 for the same three-node model without evapotranspiration.

Alternatively, Newton’s method can be derived using a Taylor series expansion of Equation 7 truncated after

![Figure 2. “Cobweb” plot showing the convergence of Picard iterations. The iterations begin at \( h = 0.1 \) and converge to a fixed point at \( h = 0.547 \). Markings along the top indicate initial guesses that will produce convergence. Markings along the bottom indicate initial guesses that do not converge. Here, the markings only appear along the top, indicating convergence from the entire range of initial guesses. The aquifer is unconfined, heterogeneous, \( h_L = 4, h_R = 0.1, K_L = 0.005, K = 0.25, K_R = 100 \), and \( ET_{\text{max}} = 0 \).](image)

![Figure 3. Convergence pattern for Newton’s method applied to the three-node problem in Figure 1. The iterations begin at \( h = 1.7 \) and converge to the solution of \( h = 0.547 \). Markings along the top indicate initial guesses that will produce convergence. Markings along the bottom indicate initial guesses that do not converge. The aquifer is unconfined, heterogeneous, \( h_L = 4, h_R = 0.1, K_L = 0.005, K = 0.25, K_R = 100 \), and \( ET_{\text{max}} = 0 \).](image)
the first derivative and solving for $\Delta h$ such that the function at $f(h + \Delta h)$ is zero. In this form, it can be shown (see Panconi and Putti 1994) that Picard’s method is equivalent to Newton’s method if the derivative terms other than $dh/dh$ are neglected. Therefore, Picard’s method can be viewed as a Newton-type method that uses a simplified approximation to the Jacobian (derivative). This is shown graphically in Figure 4.

Characteristics of Nonlinearities and Convergence Patterns

The nonlinearity present in $f_p(h)$ and $f_d(h)$ of Figures 2 through 4, respectively, is from flow in a heterogeneous unconfined aquifer; evapotranspiration was set to zero. The nonlinearity from the unconfined flow causes the curvature of the functions $f_p(h)$ and $f_d(h)$. Heterogeneity can either increase the curvature, creating more nonlinearity, or decrease the curvature so that the function is nearly linear, but no curvature would be present if the system were confined.

Including evapotranspiration causes a discontinuous change in the function because the evapotranspiration rate varies piecewise linearly with head (Equation 4). The piecewise-linear representation of evapotranspiration is evident in Figure 5, where the evapotranspiration surface ($h_s$) is set at 0.5 and the extinction depth is 0.25. This causes the curvature of the function caused by unconfined conditions to change abruptly within this range of head.

The orbit shown in Figure 5 results from applying Picard iterations and shows a different type of convergence pattern called a “staircase” plot. The staircase plot is characterized by the head changing in the same direction each iteration, in contrast to the cobweb plot of Figures 2 and 4.

Applying Newton’s method to the problem that is the subject of Figure 5 results in a convergence pattern shown in Figure 6. From the shape of the function in Figure 6, and recalling that Newton’s method follows the tangent of the function, it is evident that Newton’s method converges rapidly if the initial guess is between 0.25 and 0.5 and it fails to converge for initial guesses <0.25. For initial guesses >0.5, the solution will be found only if the convergence pattern lands somewhere in the region between 0.25 and 0.5. For this problem, the region of convergence is more sporadic than Picard’s method, as

![Figure 4](image)

**Figure 4.** Convergence pattern for Picard’s method plotted in the form of an approximation to Newton’s method. Because the derivative is approximated, the slope of lines tangent to the function is not exact. The iterations begin at $h = 0.1$ and converge to the solution of $h = 0.547$. Markings along the top indicate initial guesses that will produce convergence. Markings along the bottom indicate initial guesses that do not converge. The aquifer is unconfined, heterogeneous, $h_L = 4$, $h_R = 0.1$, $K_L = 0.005$, $K = 0.25$, $K_R = 100$, and $ET_{max} = 0$.

![Figure 5](image)

**Figure 5.** “Staircase” plot showing the convergence of Picard iterations applied to the three-node problem in Figure 1. The sharp discontinuity in the function results from evapotranspiration. The iterations begin at $h = 0.2$ and converge to the solution of $h = 0.415$. Markings along the top indicate initial guesses that produce convergence. Markings along the bottom indicate initial guesses that do not converge. The aquifer is unconfined, heterogeneous, $h_L = 2$, $h_R = 0.5$, $K_L = 100$, $K = 1.0$, $K_R = 0.1$, $ET_{max} = 2$, $h_s = 0.5$, and $d = 0.25$.

![Figure 6](image)

**Figure 6.** Convergence pattern for Newton’s method applied to the same problem solved with Picard’s method shown in Figure 5. The sharp discontinuity in the function results from evapotranspiration. The iterations begin at $h = 1.03$ and converge to the solution of $h = 0.415$. Markings along the top indicate initial guesses that produce convergence. Markings along the bottom indicate initial guesses that do not converge. The aquifer is unconfined, heterogeneous, $h_L = 2$, $h_R = 0.5$, $K_L = 100$, $K = 1.0$, $K_R = 0.1$, $ET_{max} = 2$, $h_s = 0.5$, and $d = 0.25$. 
indicated by the markings along the top and bottom of the Figures 5 and 6.

Comparison of Figures 2 through 6 illustrates that the characteristics of the nonlinearities influence the shape of the nonlinear function, which consequently has a large effect on the convergence patterns of both Picard and Newton iterations. This relationship between the shape of the nonlinearity and the convergence pattern is fundamental for solving nonlinear ground water flow equations and is further explored in subsequent sections of this article.

**Picard’s Method vs. Newton’s Method**

Three important features can be seen by comparing Figures 2 and 3. (1) The graphs of the functions that are operated on by Picard’s method and Newton’s method differ, $f_p(h)$ vs. $f_h(h)$, (Equation 6 vs. Equation 7). The former finds the intersection with the graph of $h = h$ (fixed point), while the latter finds the intersection with zero (root), which are mathematically equivalent. (2) The fixed point in the Picard iteration attracts all the initial guesses greater than zero. However, Newton’s method fails if the initial guess is <0.19; following the slope of the tangent line from points to the left of 0.19 results in heads that are either equal to or less than zero, which is physically unrealistic. (3) Newton’s method converges to the solution in fewer iterations than Picard’s method. After three iterations of Newton’s method, the approximation is within ~1% of the solution, while Picard’s method requires 10 iterations for the equivalent accuracy. However, Newton’s method requires more work per iteration because evaluation of the derivative is required. While this is problem and implementation specific, Paniconi and Putti (1994, 3359) found that, independent of dimensionality, the per-iteration computational cost of Newton’s method was about twice that for Picard’s method for the variably saturated flow problem.

Based on the results shown in Figures 2 through 6, it is not clear which method, Picard or Newton, is superior for solution of nonlinear, saturated ground water flow equations. Previous comparisons between the methods have resulted in similar unresolved conclusions. For example, Paniconi et al. (1991) and Putti and Paniconi (1992) state that Newton’s method is more robust in that it has a wider region of convergence than Picard’s method. Later, however, Paniconi and Putti (1994) indicate that Newton’s method is more sensitive to initial guesses than Picard’s method and Picard’s method generally does not diverge (which provides motivation for their mixed Picard-Newton approach). These findings contradict one another and indicate the difficulty of dealing with the wide variety of nonlinearities encountered in typical ground water flow problems. Consequently, a method that performs well in one circumstance may fail miserably in another, which makes it difficult to draw clear conclusions about the superiority of one method over another. Therefore, it is beneficial to understand the relative advantages and drawbacks of each method and characteristics of their convergence so that modelers can select the appropriate method for a given circumstance.

The aim of the remainder of this article is to increase this understanding by further analyzing the nonlinearities, their corresponding convergence difficulties, and exploring strategies for achieving convergence.

**What Can Go Wrong—Convergence Failure and Nonuniqueness**

Examination of the cobweb and staircase plots of Figure 2 and Figure 5, respectively, leads to the question of whether Picard’s method is guaranteed to converge. For example, in Figure 5, if the convergence pattern did not land on the ledge between 0.5 and 0.25, it would have been mapped to a value below 0.25, which would then map it back to the top of the staircase for another descent. Can the Picard iteration produce orbits that are attracted to a fixed point, but never reach it? The answer, unfortunately, is “yes,” as shown by Figure 7, where the only nonlinearity is from evapotranspiration. There is no curvature in the function because it is a confined system and evapotranspiration is piecewise linear (Equation 4). The evaporation surface and extinction depth are set to 1.2 and 0.4, respectively.

**How Good Does the Initial Guess Have to Be?**

In the context of dynamical systems, the convergence pattern shown in Figure 7 is a period 2 orbit (Alligood et al. 1996). Every other iteration, the orbit is attracted to 0.5 and 1.5. This is because the function that the Picard iteration is operating on has a cyclical mapping. All values $<0.8$ are mapped to 1.5, and all values $>1.2$ are mapped to 0.5, resulting in the endless oscillation for orbits that land anywhere in these regions. Only initial guesses between 0.8 and 1.2 produce orbits that converge to the solution at 1.0.

![Figure 7. Convergence pattern of Picard iterations showing a period 2 orbit. The iteration begins at $h = 0.7$ and oscillates between periodic attractors at 0.5 and 1.5. Initial guesses $>0.8$ and $<1.2$ converge to the solution at $h = 1.0$, while other initial guesses result in a periodic orbit. Nonlinearity is caused by evaporation only. Markings along the top indicate initial guesses that produce convergence. Markings along the bottom indicate initial guesses that do not converge. The aquifer is confined, homogeneous, $h_L = 2.5$, $h_R = 0.5$, $b = 1$, $K_L = K = K_R = 1$, ET$_{max} = 2$, $h_s = 1.2$, and $d = 0.4$.](image-url)
In the context of a ground water system, the convergence pattern in Figure 7 has a physical explanation. When the head is greater than the evaporation surface \( h_s = 1.2 \), evapotranspiration is at its maximum. Evapotranspiration at its maximum rate causes the head to drop to 0.5. This head drop is large enough that head falls below the extinction depth. The extinction depth \( d \) is 0.4, so when the head drops below 0.8 \( (h_s - d) \), evapotranspiration ceases, and the head increases to a level greater than \( h_s \). Thus, the system cycles between a maximum evapotranspiration rate and no evapotranspiration without experiencing the linear regime in between, and convergence is only achieved if the head moves into this linear regime. In this simple model, the linear regime can be made arbitrarily small by making the extinction depth arbitrarily small, which implies that the initial guess needed for convergence would need to be infinitely precise. Physically, this implies that evapotranspiration can go from its maximum to zero with an arbitrarily small change in head—a situation that should be avoided in practice. Although this particular example is for a confined homogeneous system, this same type of oscillation can occur for unconfined and heterogeneous systems too.

What About Uniqueness?

When solving nonlinear equations, the possibility of nonunique solutions should be considered. The nonlinear scenarios examined previously always had a unique solution, and when given a good initial guess, the Picard iteration was able to find this solution because the orbits were attracted toward it. This leads to the question, “can the Picard iteration produce orbits that are repelled from, rather than attracted to, the fixed point?” Unfortunately, the answer to this question is “yes.” Figure 8 shows an example where the orbit diverges from a valid solution and instead converges to a different valid solution.

Valota et al. (2002) posed the question of whether the forward problem is well posed. Their examples showed that it is not, but in certain circumstances, physical reasoning could be used to eliminate invalid solutions. The simple example in Figure 8 shows three valid solutions, none of which can be eliminated by physical reasoning. Unfortunately, the Picard iteration will only converge to two of the valid solutions because the solution at 0.721 repels orbits of the Picard iteration rather than attracting them. Therefore, situations can arise where Picard iteration might not converge at all (Figure 7) or it might not converge to the intended solution even from a very good initial guess.

Is There Any Hope for Obtaining Solutions?

The previous examples perhaps give a bleak outlook for solving nonlinear ground water flow equations. The issue of nonuniqueness is an intrinsic property of nonlinear equations, as pointed out by Valota et al. (2002) for the case of a multilayered aquifer system with an upper unconfined aquifer. In the three-node model considered here, nonuniqueness is only present when unconfined flow is combined with nonlinear boundary conditions—in this case, evapotranspiration. Although “drying” or “rewetting” of cells often is a source of nonuniqueness (McDonald et al. 1991; Naff et al. 2003), nonuniqueness can arise without cells or layers going “dry” or becoming “rewetted.” Unfortunately, the only way to deal with nonunique solutions is to (1) explore the possibility of multiple valid solutions by varying the initial guess and (2) try to rule out solutions using physical arguments and observations.

The issue of convergence is separate from uniqueness, and often convergence problems can be addressed by using different initial values, as shown in Figures 2 through 6, or damping. Some strategies for generating initial guesses and for implementing damping are examined subsequently.

Using a Transient Solution to Find a Steady-State Solution

Using a transient solution algorithm and marching out to large time can be used to achieve a steady-state solution. This approach is sometimes called pseudotransient continuation and has been applied to both unsaturated (Farthing et al. 2003) and saturated (Feinstein et al. 2002; Naff et al. 2003) ground water flow problems. While this may seem like more work, it can be advantageous for several reasons. First, as seen in the previous examples, a good initial guess is needed for convergence and because transient solution algorithms generally use the solution at the previous time step as the initial guess for the current time step, a good initial guess is typically available as the solution evolves toward steady state. Therefore, poor initial guesses can still converge because, over time, all initial guesses are evolved to a steady-state solution. Second, aquifer storage adds capacitance to the system, that is, it tends to absorb the effects of changes in head that can be caused by strong sinks/sources. In the...
transient flow equations, the storage terms are divided by the time-step size. Therefore, smaller time steps increase the influence of storage, which can help smooth out the effects of the nonlinearities and make for an easier numerical solution. Larger time steps decrease the effect of storage terms and the equations behave more like a steady-state system.

This procedure is demonstrated in Figure 9 using Newton’s method for the same problem shown in Figure 6 where initial guesses <0.25 did not converge. Starting with an initial guess of 0.05 and a time-step size of 0.01 leads to a head solution of 0.0583 (Figure 9a). Using this as the initial guess for the next time step and increasing the time-step size to 0.05 produce a head solution of 0.162, as shown in Figure 9b. The subsequent application results in a head solution of 0.283, as shown in Figure 9c. Recall that initial guesses >0.25 and <0.5 rapidly converge to the steady-state solution. Continuing to march in time with this initial guess and increasing the time-step size to 1.0 result in a head solution of 0.409 (Figure 9d), which is very close to the steady-state solution of 0.415. Using this approach has effectively increased the region of initial guesses that converge to the steady-state solution.

The effects of time-step size on increasing the capacitance of the system through the storage term are seen in the change of function $f_n(h)$ in Figure 9. For small time-step sizes, $f_n(h)$ is effectively linear (Figure 9a). As time-step size is increased, the impact of storage is diminished (the capacitance is reduced), and the nonlinearities of the function shown in Figure 9d closely resemble those of the steady-state system shown in Figure 6. Therefore, the effectiveness of this approach depends on the selection of time-step size.

Damping

Generally, when Picard iterations do not converge, it is because they are caught in some type of periodic attractor, as shown in Figure 7. This can be observed by head changes that oscillate. In this situation, the characteristic convergence pattern often is similar to that of the cobweb plot, but instead of the head changes decreasing until they converge, they continue to over- and underestimate by the same or increasing amounts. Damping (under relaxation), which reduces the head change between subsequent iterations, can be useful in restricting the head change such that the overshoot does not occur or is reduced, resulting in a converging cobweb plot. As simple as it sounds, damping can have a dramatic effect on the convergence of Picard iterations. For nonlinear ground water models, investigations of the effects of damping have been primarily limited to unsaturated-flow problems (for example, Cooley 1983; Paniconi and Putti 1994), but it has utility in saturated flow problems as well. For the example in Figure 2, applying damping of 0.8 results in convergence to within 1% of the solution in four iterations; without damping, 10 iterations were required, and Newton’s method required three iterations for similar accuracy.

---

**Figure 9.** Convergence patterns and evolution of the head solution when using a transient solution to find a steady-state solution for the same problem shown in Figure 6. Specific yield is 0.3 and the initial guess is 0.05. Marching in time using the solution of the previous time step as the initial guess for the current time step for the sequence of time-step sizes of (a) 0.01, (b) 0.05, (c) 0.05, and (d) 1.0. The aquifer is unconfined, heterogeneous, $h_L = 2$, $h_R = 0.5$, $K_L = 100$, $K = 1.0$, $K_R = 0.1$, $ET_{max} = 2$, $h_s = 0.5$, and $d = 0.25$.  

---

590  
S. Mehl  
GROUND WATER 44, no. 4: 583–594
However, one question remains: what value of damping should be used? Because damping restricts the head change, it can also slow the convergence rate. For example, if damping of 0.5 is applied to a problem where damping of 0.9 would suffice, it will take more iterations to achieve the same accuracy because damping slows down the solver’s progress toward the solution. Another complication arises when the head change between subsequent iterations is small due to the nonlinearity of the function (for example, the small “stair steps” near $h = 1.0$ on the staircase plot of Figure 5). Restricting head changes that are already small may prematurely satisfy termination criteria based on maximum head change, resulting in a false convergence. This might give the idea that the best strategy is to find the largest value of damping such that the solution converges because anything less than that amount will only slow down the solver. Unfortunately, it is not that simple in practice as demonstrated next.

**Application of Damping to a Model of the Death Valley Regional Flow System**

Figure 10a shows the convergence for increasing solver iterations with values of damping ranging from 0 to 1.5 for a MODFLOW-2000 version of the three-layer Death Valley regional flow system model described by D’Agnese et al. (1997). Because the model solves for several thousand unknown heads, plotting a surrogate instead of head is needed to assess the convergence to the solution. Here, the convergence is measured by the $L_2$ norm of the residuals of the matrix equations (the square root of the error in the flow mass balance in each cell summed for all cells in the model). A residual of zero indicates a perfect mass balance, and for this system a residual of 10 has negligible mass balance error; therefore, after log transforming for easier plotting, values <1 indicate an adequate mass balance. Figure 10b shows the residuals after 80 iterations for the different values of damping. Using a value of 0.81 leads to the fewest Picard iterations for convergence, and is therefore, in some sense, optimal. Notice convergence is achieved with damping of 0.85 to 0.83 and 0.81 but not with damping of 0.82. The same phenomenon occurs with damping around 0.63 as seen by the narrow valleys. Values of damping between 0.52 and 0.17 converge in 80 iterations. Values <0.17 appear to be converging although the small values of damping severely hinder the solver’s progress and more than 80 iterations are required. Values of damping >0.85 do not converge; they initially reduce the residuals, but plateau after about five iterations. Some areas of the plateau have bumps that appear at a regular spacing. These correspond to periodic attractors such as those seen in the simple three-node model in Figure 7; the head is oscillating about the solution, but not able to converge to it.

**Adaptive Damping**

For complicated nonlinear models, finding a good value of damping can be as difficult as finding a good starting guess. Figure 10a shows two features: (1) initially, all damping values work fairly well, then some plateau, while others continue to lower the residuals and (2) once the intermediate solution moves down into the valley, further damping only restricts the solver’s progress toward the final solution. Figures 10a and 10b show that finding the optimal damping value of 0.81 beforehand is difficult because damping values on either side will not converge, so there is no indicator that this value should work. Moreover, changing the model slightly can shift the narrow valleys so that the value of damping that worked previously no longer converges. Consequently, the optimal value of damping can be highly sensitive to model changes.

This leads to the idea that perhaps the value of damping should be allowed to change based on the characteristics of the convergence to the solution. This has been done for general nonlinear equations (for example, Wegstein 1958, which actually leads to a secant method), for nonlinear circuits (Neill and Stefani 1975), for chemical processes (Crowe and Nishio 1975), and in the context of ground water models (for example, Cooley 1983; Huyakorn et al. 1986; Mehl and Hill 2001). If the adaptive damping strategy developed by Cooley and modified by Huyakorn et al. is applied to this problem, convergence is achieved after 8 Picard iterations, while 20 Picard iterations are needed if the damping is fixed at 0.5, and the optimal fixed value of 0.81 required 9 Picard iterations. Therefore, for some problems, adaptive damping strategies can achieve convergence similar to the optimal fixed damping value. However, because these
methods are empirically based, they are not guaranteed to work or perform better than a fixed level of damping.

Discussion and Strategies

When convergence problems occur for complicated nonlinear ground water models, often it can be challenging to determine what is causing the convergence difficulties. Typically, “drying” or “rewetting” of cells gets blamed for the convergence and nonuniqueness problems, but this work demonstrates that these difficulties can arise without any drying/rewetting phenomena. Oscillations can occur with a confined ground water flow model with a nonlinear boundary condition, such as ET, as shown in Figure 7. If the flow is unconfined and ET is present, then nonuniqueness can result, as shown in Figure 8. Heterogeneity can make this worse by increasing the curvature and creating more than one solution—the only difference between Figure 5 and Figure 8 is the heterogeneity with the hydraulic conductivity of the center cell increased by 10%. However, heterogeneity is not required for nonuniqueness and instead of increasing the curvature, it can also decrease it, creating a more linear and easier-to-solve problem. Therefore, effects of heterogeneity are difficult to predict beforehand.

The nonlinearity that arises from a boundary condition like ET is different from the nonlinearity of unconfined flow because here, ET is represented linearly within each regime (max ET, a linear drop in ET, and no ET). However, this piecewise linearity still produces nonlinearity when the head causes the ET to change from one regime to another. It can be thought of as a discretized nonlinearity. Therefore, perhaps a better strategy would be to have a single, continuous, and smooth representation of ET instead of three piecewise functions. However, the real problem in terms of causing convergence difficulties is not the discrete nature of a piecewise-linear ET function, rather the poor conditioning that results if the extinction depth is small. Regardless of whether the function representing a sink/source is continuous or discrete, oscillations can result if the source/sink goes from a maximum to a minimum over a narrow range of values. Therefore, in situations where small extinction depths cannot be avoided, the best strategy is to use initial guesses that are within the middle linear regime so that ET can either increase or decrease as necessary and hopefully avoid the oscillations between maximum and minimum ET rates. In addition, damping can be used to alleviate some of the oscillations.

Of course, the best strategy is to have a good initial guess (a guess that is close to the true steady-state solution), but if we always had a good estimate as to how the ground water system is behaving, it probably would diminish our need for building a numerical model—an unfortunate paradox indeed.

One strategy for generating good initial guesses is to introduce complexity into the model stepwise, using the solution from the previous model as the initial guess for the model of the next level of complexity. As discussed previously, boundary conditions that have large variations over a small spatial range can be problematical. For example, ET with a small extinction depth can first be modeled with a much larger extinction depth and this solution can be used as the initial guess for the model using a smaller extinction depth, and so on until the true ET is used. However, this approach may not be effective where strong, nonlinear sinks/sources are involved because they can significantly change the model, making it difficult to introduce their effects gradually.

Unfortunately, initial guess strategies that are good for some types of nonlinearity may not be good for others. For example, when convergence problems occur because of drying/wetting phenomena, often convergence can be obtained by using an initial guess for the water table that is much higher than expected. When the solver slowly lowers the water table, the oscillations between dry and wet cells can be attenuated. However, starting with a very high water table may be a poor strategy for something like ET or other nonlinear surface boundary conditions.

Using a transient solution to march toward a steady-state solution can avoid some of the difficulties of finding a good initial guess to the steady-state solution by effectively increasing the region of convergence. However, as shown in Figure 9, the shape of the nonlinear function depends on the time-step size, with the function behaving more linearly with small time-step sizes (Figure 9a) and more nonlinearly with larger time-step sizes (Figure 9d). This brings up another difficulty of selecting a small enough time-step size such that the convergence is rapid, but large enough such that adequate progress is made toward the steady-state solution. In the example presented, the nonlinear equation had to be solved four times to converge to within ~1% of the solution. However, because the initial guesses are good, few iterations are needed for convergence of each nonlinear solution. Therefore, this approach is particularly well suited for Newton’s method where rapid convergence is achieved when initial guesses are good. The solution at intermediate time steps can be nonunique, while the steady-state solution is unique. Therefore, although this approach can increase the region of convergent initial guesses, uniqueness remains an issue and the steady-state solution can still depend on the initial guess used in the transient solution.

Different linear solvers have differing characteristics that can be helpful for certain types of problems. For example, multigrid solvers, such as the LMG (Mehl and Hill 2001) and GMG (Wilson and Naff 2004) of MODFLOW, require more CPU time per Picard iteration but also tend to provide a smoother solution each iteration, which can be advantageous in mitigating oscillations. The PCG solver of MODFLOW (Hill 1990) can be very effective for nonlinear problems when run with few inner iterations (one or two) and many outer iterations (this is akin to a steepest descent approach). In this case, the Picard iteration is computationally cheap, but many Picard iterations are required for convergence. Therefore, different linear solvers should be tried to see if better convergence can be achieved.

Damping can overcome some of the oscillation problems associated with a poor initial guess. However, it creates another problem of finding an appropriate value of damping. As shown in Figure 10b, the narrow valleys
Adaptive damping strategies can help resolve this issue, however, as Anderson (1965) points out, “For the multivariate nonlinear case, relaxation [damping] procedures have only pragmatic self-justification but can be extremely effective in some instances.” In other words, when adaptive damping strategies work, they tend to work well, but there is no guarantee that they will work at all. Therefore, because it is easy to implement and can be helpful, some damping (either fixed or adaptive) is worth trying, but these methods are empirically based and certainly are not a panacea.

Newton’s method can be combined with a line-search method (Dennis and Schnabel 1996; Farthing et al. 2003) to increase the region of initial guesses that converge to the solution. The line search essentially limits the step size and is helpful when Newton’s method diverges when overshooting the solution. Therefore, its effect is similar to damping, but instead of being empirically based, the step size is adjusted based on the progress toward the solution and previous estimates of the solution. Consequently, it is more robust than fixed or adaptive damping. However, for initial guesses where the derivative (tangent line) points in the wrong direction, as in Figure 6 for initial guesses <0.25, limiting the step size will not result in convergence. Therefore, like damping, this approach is worth implementing for Newton’s method, but it is not a panacea.

Conclusion

Picard iteration with and without damping can sometimes be very effective and robust for solving nonlinear ground water flow models, while at other times it can exhibit pitfalls (for example, slow convergence or divergence from a valid solution). Unfortunately, available research for solving nonlinear, saturated ground water flow equations has not shown Newton’s method to be definitively better because it can suffer from similar problems. Because of the wide variety of nonlinearities that occur in complicated ground water models, finding a single approach that works effectively for all problems is not likely. Therefore, understanding the properties of a variety of approaches in terms of their typical convergence patterns can be useful in analyzing and developing strategies to overcome convergence difficulties. The typical convergence patterns were investigated using a simple model and analyzed in terms of orbits of a dynamical system. Based on the characteristics of these patterns, several strategies for achieving convergence were analyzed and discussed. However, there really is no substitute for a good initial guess, and strategies were given to generate good initial guesses. A summary of these strategies are

- For problems with oscillation (periodic attractor and cobweb plot), damping can be very effective. Generally, less damping is needed as the oscillations become less severe, and adaptive damping can be used to control this. Using initial guesses that are within the middle regime or widening the middle regime of nonlinear sinks/sources (such as ET) can help reduce oscillations between extremes.
- Initial guesses that are close to the true solution can be generated by introducing complexity into the model gradually and using the previous solution as the initial guess for the next model.
- Using a transient solution to march toward a steady-state solution can effectively provide good initial guesses.
- Different linear solvers have different properties that make them more/less efficient for different problems. Therefore, different solvers as well as solver settings should be explored.
- Uniqueness should be examined by varying the initial guesses and ruling out invalid solutions based on observations and physical properties of the system.

It is hoped that the examples, discussion, and strategies presented will be useful for obtaining numerical solutions to nonlinear, saturated ground water flow models.

Acknowledgments

The author is grateful for suggestions provided by Ken Kipp, Tom Reilly, and Mary Hill which improved this manuscript. Review comments by Claire Tiedeman, Mauro Giudici, and an anonymous reviewer improved the clarity of this manuscript.

References


