A Dynamic Programming Solution to Solute Transport and Dispersion Equations in Groundwater

M. Mirabzadeh¹ and K. Mohammadi²*

ABSTRACT

The partial differential equations for water flow and solute transport in a two-dimensional saturated domain are rendered discrete using the finite difference technique; the resulting system of algebraic equations is solved using a dynamic programming (DP) method. The advantage of the DP algorithm is that the problem is converted from solving an algebraic system of order NC(NL-1)×NC(NL-1) into one of solving a difference equation of order NC×NC over NL-1 steps and involving NL-1 matrix inversions of order NC×NC. The accuracy and precision of the solutions are shown by comparing the results with an analytical solution and calculation of mass the balance. In addition, the performance of the DP model was compared with the results of the MOC model developed by US Geological Survey. In all cases, the DP model showed good results with sufficient accuracy.

Keywords: Dynamic programming, Groundwater, Numerical model, Solute transport.

INTRODUCTION

In Iran, groundwater is an important source of irrigation water. In fact, it fulfills more than 50 percent of the country’s total irrigation needs. Groundwater has played a major role in increasing food production and achieving food security. Groundwater, a renewable source of water, has the remarkable distinction of being a highly dependable and safe source of water supply for agriculture, domestic and industrial needs. Increased use of chemical fertilizers coupled with improper water management practices has resulted in the deterioration of groundwater quality in several parts of the country. Every year, almost 17.5 BCM of agricultural wastewater is returned to aquifers which account for almost 35% of groundwater extraction (1). This recycling of water causes secondary salinization in irrigated areas.

Solute transport models are being used to study groundwater quality and water pollution in the subsurface. Several methods have been used to generate a better and faster model, for example: Reddell and Sunada (20); Chatwal et al. (6); Cheng and Hodge (7); Gray and Pinder (11); Grove (12); Huyakorn et al. (13); Konikow and Bredehoeft (14); and Bouhroum and Bai (5). In addition, certain advances in analytical solutions have been made by some other researchers such as Basha and El Habel (2) and Sim and Chrysiopoulos (21). They have developed one-, two-, and three-dimensional analytical solutions for solute transport in saturated, homogeneous porous media. However, numerical algorithms are still the only solution to complex problems in the field of solute transport in groundwater.

In this paper, a numerical model to solve the solute transport and dispersion equation in groundwater is developed. The purpose of
the simulation model is to compute the concentration of a non-reactive dissolved chemical species in an aquifer at any given time and place. The partial differential equations are made discrete using the finite difference technique and the resulting system of algebraic equations is solved using a dynamic programming (DP) method which is used here for the first time. Bellman (4), Collins (8) and Collins and Angel (9) applied this technique to the solution of parabolic and elliptic partial differential equations. A dynamic programming method was then used by Mirabzadeh (16, 17, 18, 19) to solve the flow equation in two and three dimensions for heterogeneous and anisotropic aquifers. The higher speed of this algorithm compared with the other matrix solutions such as Jacobi, Guass-Sidel, and Successive over Relaxation (SOR) has been the motivation for using the method in this research (9, 17).

Convective transport, hydrodynamic dispersion, and chemical reactions are the three main processes in solute transport. Among these, the first two phenomena are more important in most cases and those have been considered in this model. Because convective transport and hydrodynamic dispersion depend on the velocity of groundwater flow, the solute transport equation is a nonlinear equation and must be considered in conjunction with the groundwater flow equation. The computer program solves two simultaneous partial differential equations- one is the groundwater flow equation and the second one is the solute transport equation.

A solution to the solute transport equation in two dimensions has been obtained using a dynamic programming technique. In this paper, the analytical solution (2) and the method of characteristics were used to verify the developed algorithm. The method of characteristics was originally applied to transport in porous media by Garder et al. (11) in order to calculate miscible displacement in a reservoir simulation. This method was later implemented in a two-dimensional solute transport model by Konikow and Bredehoeft (14); their code is commonly referred to as MOC. The method of characteristics uses a conventional particle tracking technique for solving the advection term. For more detail about this method, one can refer to Zheng and Bennett (22).

**Theoretical Development**

In a Cartesian coordinates, the governing equation for two dimensional fluid flow in a saturated porous medium by using the continuity equation and Darcy’s law, is:

\[
\frac{\partial}{\partial x} \left( T_{xx} \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left( T_{yy} \frac{\partial h}{\partial y} \right) = S \frac{\partial h}{\partial t} \pm q
\]  

where \( h \) is hydraulic head [L]; \( t \) is time [T]; \( q \) is source/sink term [L/T] with a positive sign for the source and negative for sink terms; \( S \) is storage coefficient; and \( T_{xx} \) and \( T_{yy} \) are transmissivity [L²/T] in x and y direction, respectively. Transmissivity is defined as follows:

\[
[T_{ij}] = b[K_{ij}]
\]

where \( b \) is the saturated thickness of the aquifer [L]; and \( [K_{ij}] \) is the hydraulic conductivity tensor. It is assumed that the coordinate system is oriented with the conductivity tensor, so that \( K_{ij} = 0 \) for \( i \neq j \).

The solute transport equation describing convection and dispersion of a nonreactive dissolved chemical species in groundwater is as follows (14):

\[
\frac{\partial}{\partial t} \left( bn_{e} D_{xx} \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left( bn_{e} D_{yy} \frac{\partial C}{\partial y} \right) + \frac{\partial}{\partial x} \left( bn_{e} D_{xy} \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left( bn_{e} D_{xy} \frac{\partial C}{\partial y} \right) = \frac{\partial}{\partial t} \left( bV_{x} \frac{\partial C}{\partial x} + bV_{y} \frac{\partial C}{\partial y} \right) + C(n_{e}) \frac{\partial b}{\partial t} + S \frac{\partial h}{\partial t} + q(C-C)
\]

where \( C \) is the concentration of the dissolved chemical species [M/L³]; \( C' \) is the concentration of dissolved chemical in a source or sink fluid [M/L³]; \( n_{e} \) is the effective porosity; \( V_{x} \) and \( V_{y} \) are Darcy’s velocities in x and y directions, respectively [L/T]; and \( D_{ij} \) is the coefficient of the hydrodynamic dispersion [L²/T]. The dispersion coefficient for
two-dimensional flow in an isotropic aquifer may be written as (4):

\[
D_{x} = \alpha_{x} \left( \frac{V}{|V|} \right)_{j,1} + \alpha_{v} \left( \frac{V}{|V|} \right)_{j,1} \tag{4}
\]

\[
D_{y} = \alpha_{x} \left( \frac{V}{|V|} \right)_{j,1} + \alpha_{v} \left( \frac{V}{|V|} \right)_{j,1} \tag{5}
\]

\[
D_{z} = D_{n} = (\alpha_{v} - \alpha_{x}) \left( \frac{V}{|V|} \right)_{j,1} \tag{6}
\]

where \(\alpha_{x}\) and \(\alpha_{v}\) are the longitudinal and transverse dispersivities of the aquifer, respectively [L]; and \(|V|\) is the magnitude of the velocity [L/T].

**Finite Difference Formulation**

The two-dimensional model presented here is based on an implicit finite difference scheme using the dynamic programming solution.

**Flow Equation**

The compact finite difference form of equation (1) is given by:

\[
W_{x(i,j)} h_{i-1,j}^{k} + W_{x(i,j)} h_{i+1,j}^{k} + W_{y(i,j)} h_{i,j+1}^{k} + W_{y(i,j)} h_{i,j-1}^{k} + W_{c(i,j)} h_{i,j}^{k} - W_{c(i,j)} h_{i,j}^{k} = R_{0}^{k-1}
\]

where \(i = 2, \ldots, N_{L}-1\)

\(j = JD(i), \ldots, JF(i)\)

\(W_{x(i,j)}\), \(W_{y(i,j)}\), \(W_{c(i,j)}\), \(R_{0}^{k-1}\) are the column numbers at the beginning and the end of each row, respectively.

\[
W_{x(i,j)} = T_{x(i,j)} \frac{\Delta y(j)}{\Delta x(i,j)} \tag{8a}
\]

\[
W_{y(i,j)} = T_{y(i,j)} \frac{\Delta x(i,j)}{\Delta y(j)} \tag{8b}
\]

\[
W_{c(i,j)} = W_{c(i,j)} + W_{x(i,j)} + W_{y(i,j)} \tag{8c}
\]

\[
R_{0}^{k-1} = \Delta x(i,j) \Delta y(j) \left( S_{0}(j) h_{i,j+1}^{k-1} + q_{k-1}^{i,j} / \Delta t \right) \tag{8d}
\]

**Solute Transport Equation**

Using the finite difference discretization approach, equation (3) can be written as:

\[
\begin{align*}
[ B_{x(i,j)} c_{i,j}^{k} + B_{x(i,j)} c_{i,j}^{k} + B_{y(i,j)} c_{i,j}^{k} ] + B_{y(i,j)} c_{i,j}^{k} - B_{e(i,j)} c_{i,j}^{k} \]
\end{align*}
\]

\(R_{e(i,j)}^{k-1}\) \(g\)

where

\[
B_{x(i,j)}^{k} = b_{x(i,j)}^{k} n_{e(i,j)}^{k} D_{x(i,j)}^{k} \tag{10a}
\]

\[
B_{y(i,j)}^{k} = b_{y(i,j)}^{k} n_{e(i,j)}^{k} D_{y(i,j)}^{k} \tag{10b}
\]

\[
B_{c(i,j)}^{k} = \left[ B_{x(i,j)}^{k} + B_{x(i,j)}^{k} + B_{y(i,j)}^{k} + B_{y(i,j)}^{k} + \frac{\Delta x(i,j) \Delta y(j)}{\Delta t} n_{e(i,j)}^{k} b_{e(i,j)}^{k} \right] \tag{10c}
\]

\[
R_{e(i,j)}^{k-1} = R_{e(i,j)}^{k-1} + R_{e(i,j)}^{k-1} - R_{e(i,j)}^{k-1} \tag{10d}
\]

\[
B_{x(i,j)}^{k} = b_{x(i,j)}^{k} n_{e(i,j)}^{k} D_{x(i,j)}^{k} \tag{10e}
\]

\[
B_{y(i,j)}^{k} = b_{y(i,j)}^{k} n_{e(i,j)}^{k} D_{y(i,j)}^{k} \tag{10f}
\]

\[
R_{e(i,j)}^{k-1} = \left[ b_{x(i,j)}^{k} \frac{\Delta x(i,j) \Delta y(j)}{\Delta t} \right] + \left[ b_{y(i,j)}^{k} \frac{\Delta x(i,j) \Delta y(j)}{\Delta t} \right] - \left[ b_{c(i,j)}^{k} \frac{\Delta x(i,j) \Delta y(j)}{\Delta t} \right]
\]

\[
B_{c(i,j)}^{k} = b_{c(i,j)}^{k} n_{e(i,j)}^{k} D_{c(i,j)}^{k} \tag{10g}
\]

\[
B_{c(i,j)}^{k} = b_{c(i,j)}^{k} n_{e(i,j)}^{k} D_{c(i,j)}^{k} \tag{10h}
\]

where \(\alpha_{x}, \alpha_{y}, \alpha_{c}, \alpha_{e}\) are the same thing for \(\alpha_{x}, \alpha_{y}, \alpha_{c}, \alpha_{e}\).
\[
\frac{R^{k-1}_{(i,j)}}{\Delta x_{(i,j)}} = \left[ C \left( \frac{\partial b}{\partial t} + S \frac{\partial h}{\partial t} \right) + q(C^t - C) \right] = C_{(i,j)}^{k-1} \left[ n_{e(i,j)} \frac{b^k_{(i,j)} - b^{k-1}_{(i,j)}}{\Delta t} - S_{(i,j)} \frac{h^k_{(i,j)} - h^{k-1}_{(i,j)}}{\Delta t} \right] + q\left[C_{(i,j)}^t - C_{(i,j)}^{k-1}\right]
\]

\[(10i)\]

**Dynamic Programming Solution**

In this section, the proposed relationships for solving the resulting algebraic equations are presented. In order to solve equation (7) subject to the boundary conditions, we define the vectors \( H_i, W_i, \) and \( R^*_i \) with the dimension of \( N(i)= NC - 2 \) for each row:

\[
H_i = h^k_{(i,j)} \quad W_i = [W_{e(i,j)}, h^*_{(i,j)}] \\
R^*_i = R^k_{(i,j)}
\]

Using the above definitions, equation (7) may be rewritten as follows:

\[
F_{(i-1)} \cdot H_{(i-1)} + F_{(i)} \cdot H_{(i)} - G_{(i)} \cdot H_{(i)} + W_{i} = R^*_i, \quad i = 2,3,\ldots, NL - 1
\]

\[(11)\]

Vectors \( H_{(1)}, H_{(NL)}, \) and \( W_{i} \) are known from the boundary conditions, \( F_{(i)} \) and \( F_{(i+1)} \) are the diagonal matrices, and \( G_{(i)} \) is a tridiagonal matrix with the components defined by the coefficients of the system of equations (7) as follows:

\[
F_{(i)} = \text{diag}(f) \quad f = W_{x(i)} \cdot \delta_{i(j)} \begin{cases} W_{c(i,j)} & i = j \\ W_{y(i,j)} & i = j - 1 \\ 0 & \text{otherwise} \end{cases}
\]

\[
G_{(i)} = \begin{bmatrix} g_{(i,j)} \\ g_{(i,j)} \end{bmatrix} = \begin{cases} -W_{y(i,j)} & i = j - 1 \\ -W_{y(i,j)} & j = i + 1 \\ 0 & \text{otherwise} \end{cases}
\]

Using dynamic programming, equation (11) may be cast in the following form:

\[
H_{(i)} = A_{(i-1)} \cdot H_{(i-1)} + B_{(i-1)} \quad (12)
\]

Solving (11) for \( H_{(i)} \) and using \( H_{(i+1)} = A_{(i)} \cdot H_{(i)} + B_{(i)} \) gives the following equation:

\[
H_{(i)} = F_{(i-1)} \cdot (G_{(i)} - F_{(i)} \cdot A_{(i)})^t \cdot H_{(i-1)} + (G_{(i)} - F_{(i)} \cdot A_{(i)})^t \cdot (F_{(i)} \cdot B_{(i)} + W_{i} - R^*_i) \]

\[(13)\]

Comparing (12) and (13) yield the following recurrence relations:

\[
A_{(i-1)} = F_{(i-1)} \cdot (G_{(i)} - F_{(i)} \cdot A_{(i)})^t \quad (14)
\]

\[
B_{(i-1)} = A_{(i-1)} \cdot F_{(i)} \cdot B_{(i)} + W_{i} - R^*_i \quad (15)
\]

The initial conditions for (14) and (15) are derived from (12) setting \( i = NL \)

\[A_{(NL-1)} = 0 \quad B_{(NL-1)} = H_{(NL-1)}\]

Then, the backward solution of (14) and (15) will give the matrices \( A_{(i)} \) and the vectors \( B_{(i)} \). Equation (12) may now be solved for \( H_{(i)} \) by utilizing \( A_{(i)} \) and \( B_{(i)} \).

Equation (9) can be solved in the same way. In each time step, after calculating the dispersion coefficient tensor, matrices \( A_{c(i)} \) and vectors \( B_{c(i)} \) and \( C(i) \) will be computed in a similar procedure to \( A_{(i)} \) (14), \( B_{(i)} \) (15), and \( H_{(i)} \) (12), respectively.

**RESULTS AND DISCUSSION**

In order to verify that the numerical model is solving the governing equations correctly, it is necessary to compare the results to known solutions or results from other models. The solution developed by Basha and El Habel (2) was used to verify the model against the analytical solution. A one-dimensional solute transport in saturated media in the direction was solved analytically. Table 1 shows the parameters that were used in this solution. Figure 1 shows the comparison between the analytical solution and DP model. The mean absolute error was 0.8 percent and the root mean square error between two methods was 0.03 percent which shows good agreement between results.

<table>
<thead>
<tr>
<th>Table 1. Parameters used in the analytical solution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Δx</td>
</tr>
<tr>
<td>Length (L)</td>
</tr>
<tr>
<td>Initial Concentration (C)</td>
</tr>
<tr>
<td>Velocity (Vx)</td>
</tr>
<tr>
<td>Dispersivity ((\alpha))</td>
</tr>
<tr>
<td>Injected Mass (C')</td>
</tr>
</tbody>
</table>
In addition, the model was compared to the results obtained from the MOC model (14). A mass balance test was used as a measure of the numerical accuracy of the solutions in DP and MOC comparisons. Here, the model uses, as the mass residual, $M_r$, a criterion compared with the difference between the initial mass, $M_i$, and the net mass flux, $N_f$.

$$\text{Error} = \frac{100M_r}{M_i - N_f}$$  \hspace{1cm} (16)

where $M_r = \Delta M - N_f$ and $\Delta M$ is the change in mass stored in the aquifer [M].

One recharge well and one extraction well were considered in a 2192×2466 m domain. The left and right sides of the domain were impermeable and the northern and southern sides of the domain had constant head boundaries with a 22.5 m and 30 m water elevation, respectively. Figure 2 shows the result for test problem number 1 and the following parameters were used in the compu-
tations:

\[ K = 1.5 \times 10^{-3} \text{ m/s} \]
\[ \Delta x = \Delta y = 274 \text{ m} \]
\[ b = 6.1 \text{ m} \]
\[ S = 0.3 \]
\[ \alpha_L = 30.5 \text{ m} \]
\[ \alpha_f / \alpha_L = 0.3 \]
\[ C^* = 220 \text{ mg/lit} \]
\[ C_0 = 0.0 \text{ mg/lit} \]
\[ q(3,5) = 0.028 \text{ m}^3/\text{s} \]
\[ q(6,4) = -0.028 \text{ m}^3/\text{s} \]

Figure 3 presents the mass balance error for the dynamic programming method and method of characteristic.

In test problem 2, the effect of dispersion was eliminated (\( \alpha_L = 0.0 \)) and other parameters remained the same as in test problem number 1. Again, Figure 4 shows the results of the solute concentration after 2 1/2 years and Figure 5 shows the mass balance errors for this period.

Figure 3. Mass balance error for test problem 1.

Figure 4. Comparison between the computed concentrations in test problem 2.
A general two-dimensional model has been developed to simulate solute transport through heterogeneous porous media. The agreement between the results from the present model and the analytical solution and the MOC model are good and the mass balance error for the DP model is less than for the MOC model.

Using a dynamic programming method can reduce the time of computation and computer error due to inversion of the large matrices. The application of this method converts the problem from solving an algebraic system of order NC \((NL-1) \times NC(NL-1)\) into that of solving a difference equation of order \(NC \times NC\) over \(NL-1\) steps and involving \(NL-1\) matrix inversions of order \(NC \times NC\). In addition, it reduces the storage requirements, since instead of having to store \((NL \times NC)^2\), we need to store only \((NC \times NC)\) numbers at each step.

**REFERENCES**

حل برنامه ریزی پویای معادله انتقال و پخشیدگی املاح در آب زیرزمینی

م. میرآذاده و ک. محمدی

چکیده

معادلات دیفرانسیل با مشتقات جزئی برای جریان و انتقال املاح در محیط دو بعدی و اشباع آب زیرزمینی با روش تفاضل های محدود بصورت گسترده در آمد و دستگاه معادلات جبری به دست آمده به روش برنامه ریزی پویا و حل گرددند. مزیت الگوریتم برنامه ریزی پویا در این است که مسئله را از حل NC\times NC را به حل یک دستگاه معادلات از مرتبه NC(NL-1)\times NC(NL-1) می‌کنیم. در گام تبدیل کرده که شامل NL-1 معکوس کردن ماتریس مرتبه NC\times NC است. دقت روش‌های حل با مقایسه نتایج با حل تحلیلی و محاسبه بیلان جرمی نشان داده شده است. مدل برنامه ریزی پویا MOC همچنین با مدل DP که توسط مؤسساتی سه شانسی آمریکا به شده مقایسه گردید.

انتشار داد که می‌تواند نتایجی با دقت خوب ارائه دهد.