

# Computational aspects of accurately modelling salt advection beneath aquaculture ponds

D. P. Verrall<sup>1</sup>W. W. Read<sup>2</sup>K. A. Narayan<sup>3</sup>

(Received 4 December 2006; revised 23 December 2008)

## Abstract

Aquaculture is becoming an important primary industry in Australia and the Asian region. Environmental concerns are raised when above ground saline ponds are introduced over fresh groundwater. The flow field can be simply modelled using series solutions for the hydraulic potential. However, extracting the advection fronts and isochrones can be much more challenging, particularly when solutions are sought using a laptop or desktop computer. We present an efficient and accurate way to calculate the advection front and the isochrones for a realistic problem. These methods use the analytic nature of the flow field solution and are an efficient and accurate alternative to the traditional methods used.

## Contents

### 1 Introduction

C1076

---

<http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/195> gives this article, © Austral. Mathematical Soc. 2009. Published February 9, 2009. ISSN 1446-8735. (Print two pages per sheet of paper.)

<i>1 Introduction</i>	C1076
<b>2 Original methodology</b>	<b>C1077</b>
<b>3 Problems</b>	<b>C1080</b>
<b>4 Modified methodology</b>	<b>C1082</b>
<b>5 Comparison</b>	<b>C1084</b>
<b>6 Discussion</b>	<b>C1086</b>
<b>References</b>	<b>C1088</b>

## **1 Introduction**

The solution to Laplace's equation has many applications in a wide variety of fields. In the area of groundwater modelling, analytic solutions of Laplace's equation provide accurate expressions for the flow field throughout a domain. Using a series solution, Powers et al. [2], Read [3, 4] and others solved for arbitrary boundaries within a saturated domain, whereas Philip [1] successfully applied the technique to infiltration. But being able to represent streamlines, lines of potential and velocity fields only represents part of the picture when dealing with groundwater movement. When modelling the effects of contamination on a groundwater system, we need to know the location of advection fronts at particular times (isochrones). In the past, this was achieved using purely numerical techniques, but accurate solutions can be difficult to obtain and verify in this manner. However, a simple extension to the closed form solutions above provide these isochrones accurately and computationally efficiently.

Figure 1 represents a soil profile taken from an aquaculture farm in the North Queensland region, where saline ponds sit above a freshwater aquifer 40 metres deep. Given the worst case scenario of saltwater leaking freely into the groundwater beneath, we determine the contaminant advection fronts using

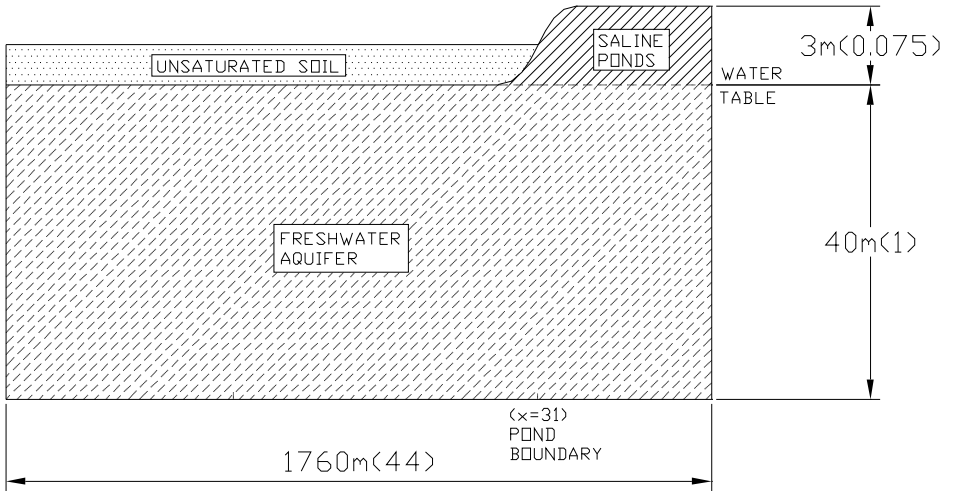


FIGURE 1: Saline ponds above freshwater aquifer (bracketed terms are dimensionless)

two different techniques. The first method is a more traditional approach for solving time dependent problems numerically. We integrate along the closed form solution of the streamlines using a small discrete time step. This technique has an advantage in simplicity, but can be slow to extract a solution, especially for low flow domains, or large isochrones. For this reason, we present a second method which uses a small discrete spatial step. The new technique is significantly quicker, but introduces some complexity which we discuss.

## 2 Original methodology

Read [3, 4] describes the method of series solutions in some detail. For our example (Figure 1), we non-dimensionalize the domain using the depth as

our scaling factor. These non-dimensional parameters (shown in brackets) constitute the domain of consideration in the rest of this article. Using the method described by Read [3, 4], we truncate after  $N + 1$  terms to obtain a potential function  $\phi(x, y)$  and a stream function  $\psi(x, y)$ ,

$$\phi(x, y) = \sum_{n=0}^N A_n \cosh \frac{n\pi y}{r} \cos \frac{n\pi x}{r}, \quad (1)$$

$$\psi(x, y) = - \sum_{n=1}^N A_n \sinh \frac{n\pi y}{r} \sin \frac{n\pi x}{r}, \quad (2)$$

where

$$A_0 = \frac{1}{r} \int_0^r h^t(x) dx, \quad (3)$$

$$A_n = \frac{2}{r} \int_0^r h^t(x) \cos \frac{n\pi x}{r} dx, \quad n \neq 0. \quad (4)$$

The partial derivatives of equations (1) and (2) (that is, the fluid velocity) satisfy the bottom and side (impermeable) boundary conditions exactly (to machine precision), with the remaining boundary condition ( $h^t(x)$ ) used to evaluate the series coefficients  $A_n$ ,  $n = 0, 1, \dots, N$ .

From the series solution of the hydraulic head  $\phi$ , we find the seepage velocity  $\bar{\mathbf{u}}$  (that is, the macroscopic velocity averaged over the void and solid area) from Darcy's equation,

$$\bar{\mathbf{u}} = -K \nabla \phi(x, y), \quad (5)$$

where the hydraulic conductivity  $K$  is a scalar. To find the pore velocity  $\bar{\mathbf{U}}$  (that is, the actual flow velocity through the void space) from equation (5), we simply divide through by the soil porosity  $\sigma$  (also a scalar),

$$\bar{\mathbf{U}} = -\frac{K}{\sigma} \nabla \phi(x, y). \quad (6)$$

We non-dimensionalize equation (6) with the scaling factor  $K/\sigma$  and separate the directional components to get an expression for the velocity field  $\mathbf{U} = (\mathbf{U}, \mathbf{V})$ ,

$$\mathbf{U} = - \left( \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y} \right). \quad (7)$$

The velocity at any point is also

$$\mathbf{U} = \left( \frac{dx}{dt}, \frac{dy}{dt} \right), \quad (8)$$

where the location at which the pore velocity is calculated is parametrised using time  $t$ . That is,  $x$  and  $y$  are no longer independent variables, but are now time dependent:  $x \equiv x(t)$ ,  $y \equiv y(t)$ . Thus, from equations (7) and (8), the time  $t$  taken for a neutrally buoyant particle to be advected along a streamline from  $(x_0, y_0)$  to  $(x(t), y(t))$  is

$$t = \int_{x_0}^{x(t)} \frac{dx}{U} = - \int_{x_0}^{x(t)} \frac{dx}{\partial \phi / \partial x} \quad (9)$$

$$= \int_{y_0}^{y(t)} \frac{dy}{V} = - \int_{y_0}^{y(t)} \frac{dy}{\partial \phi / \partial y}. \quad (10)$$

We use this result to determine isochrones for the saltwater as it is transported away from the aquaculture ponds. Re-arranging equations (9) and (10), the distances  $(\Delta x, \Delta y)$  a particle is advected in the  $x$  and  $y$  directions in a small time interval  $\Delta t$  is

$$\Delta x \approx - \frac{\partial \phi}{\partial x} \Delta t, \quad (11)$$

$$\Delta y \approx - \frac{\partial \phi}{\partial y} \Delta t. \quad (12)$$

Consider at time  $t = t_0$  a discrete set of  $I > 0$  points  $(x_{i,0}, y_{i,0})$ ,  $i = 1, \dots, I$ . Then, for any time  $t_J = t_0 + J\Delta t$ ,  $J > 0$ , we calculate the approximate

location of the advection front  $(x_{i,J}, y_{i,J})$  by calculating the intermediate points  $(x_{i,j}, y_{i,j})$ ,  $j = 1, \dots, J - 1$  using the difference equations

$$x_{i,j} \approx x_{i,j-1} - \left( \frac{\partial \phi}{\partial x} \right)_{i,j-1} \Delta t, \quad (13)$$

$$y_{i,j} \approx y_{i,j-1} - \left( \frac{\partial \phi}{\partial y} \right)_{i,j-1} \Delta t, \quad (14)$$

where

$$\left( \frac{\partial \phi}{\partial x} \right)_{i,j} = \left( \frac{\partial \phi(x, y)}{\partial x} \right)_{x=x_{i,j}, y=y_{i,j}}, \quad (15)$$

$$\left( \frac{\partial \phi}{\partial y} \right)_{i,j} = \left( \frac{\partial \phi(x, y)}{\partial y} \right)_{x=x_{i,j}, y=y_{i,j}}. \quad (16)$$

For sufficiently small  $\Delta t$  (that is, sufficiently large  $J$ ), this converges to the advection front at time  $t_J$ . Using this algorithm, we predict the advection of solutes from any initial location in the flow domain. In particular, we determine the isochrones of solutes advected from the water table immediately below the saline ponds. Choosing an isochrone of  $t_J = 200,000$  dimensionless units, Figure 2 shows the extent of the advection along each streamline. We have then joined these final advection locations to create the advection front, or isochrone.

### 3 Problems

The above methodology is inherently simple. But for the example considered the large aspect ratio and relative small potential results in extremely low flow velocities in part of the domain. As we see from equations (11) and (12), a low velocity results in small spatial increments  $(\Delta x, \Delta y)$ . If this happens, then even for a large isochrone value, as in Figure 2, the final location of the advection front moves little from its initial location (observe the area near

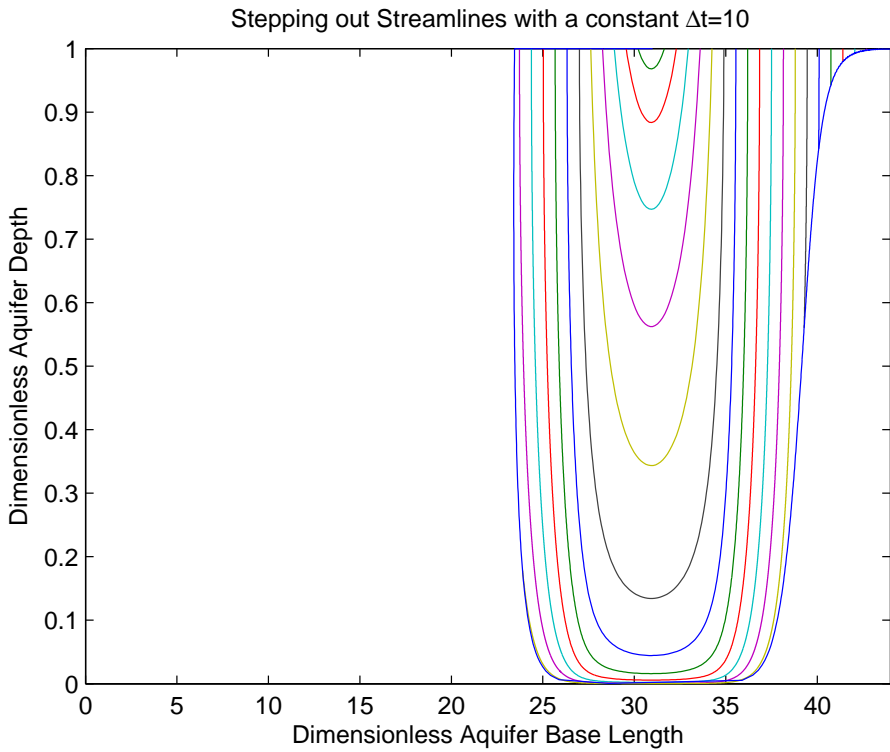


FIGURE 2:  $\Delta t$  solution scheme giving isochrone at  $t_j = 200,000$ .

the right boundary of Figure 2). Although this is not a problem when solving for particular isochrones, solving for the time to reach a particular location can be problematic. If the location is in the low velocity part of the domain, then the solution output time may be extremely high. If we want to solve for the entire domain (that is, the time for water to move everywhere within the domain), then the time to produce a solution may be excessive.

## 4 Modified methodology

For the most part, our modified methodology is similar to that of the original. From Section 2, we follow the procedure described to equations (9) and (10). It is after this point that our new method diverges.

From the original methodology, we observe that in the low velocity part of the domain our solution progresses extremely slowly. If we could get a consistent rate of progress throughout the domain at every iteration, this should result in a faster solution output time. Rather than calculating the distance we move along a streamline for a set  $\Delta t$ , we set  $\Delta s$  as the length we move along the streamline, and find the time it takes to do so, that is,

$$\Delta s = \sqrt{(\Delta x)^2 + (\Delta y)^2} = \text{constant} . \quad (17)$$

Equation (7) defines the velocity field,  $\mathbf{U} = (U, V)$ , everywhere within the domain. We define  $W$  as the speed along  $\Delta s$ , therefore

$$W = \sqrt{U^2 + V^2} . \quad (18)$$

We also know,

$$W \approx \frac{\Delta s}{\Delta t} . \quad (19)$$

So, re-arranging and solving equations (11), (12) and (19),

$$\Delta x \approx \left( \frac{U}{W} \right) \Delta s , \quad (20)$$



$$\Delta \mathbf{y} \approx \left( \frac{\mathbf{V}}{W} \right) \Delta \mathbf{s}, \quad (21)$$

$$\Delta t \approx \frac{\Delta s}{W}. \quad (22)$$

Stepping along each streamline by  $\Delta \mathbf{s}$  means consistent progress at each iteration, even in the low flow part of the domain. However, unlike the original method where the time at any iteration,  $j$ , was found from the equation ( $t_j = t_0 + j\Delta t$ ), now the time varies, not only for each iteration, but for each streamline in an iteration. For the modified methodology, as well as recording  $\Delta \mathbf{x}$  and  $\Delta \mathbf{y}$  as in the original method, from equation (22), we need to record  $\Delta t$ .

Consider at time  $t = t_{i,0}$  a discrete set of  $I > 0$  points  $(x_{i,0}, y_{i,0})$ ,  $i = 1, \dots, I$ . For a given number of iterations  $J$ , we now determine our progress through the domain, both spatially and temporally  $(x_{i,J}, y_{i,J}, t_{i,J})$ , by calculating the intermediate values  $(x_{i,j}, y_{i,j}, t_{i,j})$ ,  $j = 1, \dots, J - 1$  using the difference equations

$$x_{i,j} \approx x_{i,j-1} + \left( \frac{U_{i,j-1}}{W_{i,j-1}} \right) \Delta s, \quad (23)$$

$$y_{i,j} \approx y_{i,j-1} + \left( \frac{V_{i,j-1}}{W_{i,j-1}} \right) \Delta s, \quad (24)$$

$$t_{i,j} \approx t_{i,j-1} + \left( \frac{\Delta s}{W_{i,j-1}} \right), \quad (25)$$

where

$$U_{i,j} = - \left( \frac{\partial \phi(x, y)}{\partial x} \right)_{x=x_{i,j}, y=y_{i,j}}, \quad (26)$$

$$V_{i,j} = - \left( \frac{\partial \phi(x, y)}{\partial y} \right)_{x=x_{i,j}, y=y_{i,j}}, \quad (27)$$

$$W_{i,j} = \sqrt{(U_{i,j})^2 + (V_{i,j})^2}. \quad (28)$$

Given a sufficiently small  $\Delta s$  (that is, sufficiently large  $J$ ), the advection location on any streamline will converge at time  $t = t_{i,j}$  (each value of  $i$  represents a separate streamline). But unlike the original methodology, our values of  $t_{i,j}$  are not the same, and this introduces some complexity into our scheme. Let's take the previous example of solutes advected from the water table immediately below the saline ponds. Choosing an isochrone of 200,000 dimensionless units, we now have to find the values of  $i, j$  where  $t_{i,j} = 200,000$ .

In our case, we found the simplest method was to let the scheme iterate through  $j$ , with a stopping condition that for all  $i$ ,  $t_{i,j} > 200,000$ . The ends of the streamlines in Figure 3 show when this stopping condition is met. We now must find the values of  $t_{i,j} = 200,000$ , but this is complicated by the discrete nature of  $\Delta t$ . We may not find even one set of  $i, j$  values where  $t_{i,j} = 200,000$ , so we must find the closest such values on each streamline and interpolate. Using these values of  $i, j$ , we find the corresponding values of  $x_{i,j}$  and  $y_{i,j}$ , and applying the same interpolation ratios, we produce the isochrone for  $t = 200,000$  seen in Figure 3.

## 5 Comparison

The constant time step approach is an intuitive method, but when dealing with low velocity domains such as Figure 1, we found the time taken to produce a solution was excessive. To produce satisfactorily accurate results for the example considered in this article (Figure 2), we needed to run the scheme overnight using a standard PC (Intel Pentium 4 CPU, 3.40 GHz, 1 GB RAM, Matlab). The constant spatial step approach was developed in response to our need for faster solution output times. Keeping all other parameters equal, but changing to a constant spatial step, the example in this article (Figure 3) achieved a seven fold reduction in solution output time, from 11 hours to 95 minutes.



Of course, faster solutions can be achieved using the constant time step approach, simply by choosing a larger value of  $\Delta t$ , and allowing our accuracy to suffer. To show that the increase in speed of the constant spatial step method is not achieved by a corresponding decrease in accuracy, we compare the two methods to the analytic solution. With both methods, we step through the domain incrementally using the characteristics of the flow field at each point to calculate the location of the next point. If our schemes are accurate, the paths stepped out in this fashion correspond to streamlines derived from the stream function (equation 2) starting at the same points. We judge whether our values of  $\Delta t$  or  $\Delta s$  are sufficiently small by finding the corresponding deviation from the analytic result.

For each of our methods, we take the final point of every streamline which emerges at the water table for an isochrone of 200,000, and compare it with the final point as found from the analytic solution. Figure 4 is a plot of the deviation of each of these streamlines for the examples in this article, that is,  $\Delta t = 10$  and  $\Delta s = 0.005$ . We see that the  $\Delta s$  method not only gives a seven fold improvement in solution output time, but is approximately an order of magnitude more accurate as well.

## 6 Discussion

The constant spatial step method was developed to reduce the time required to solve isochrones in low velocity flow domains. Since we are generally interested in the physical location of a contaminant as it progresses through a domain, the fact that this method is consistent in its progress at any point is a big advantage over the constant time step method. A constant time step will require a certain number of iterations to solve for a particular isochrone, but in the low velocity part of the domain, the physical progress of the contaminant may be infinitesimal compared with the progress elsewhere. However, the constant spatial step method can potentially solve even large isochrones in this low velocity area in a single iteration. For this reason

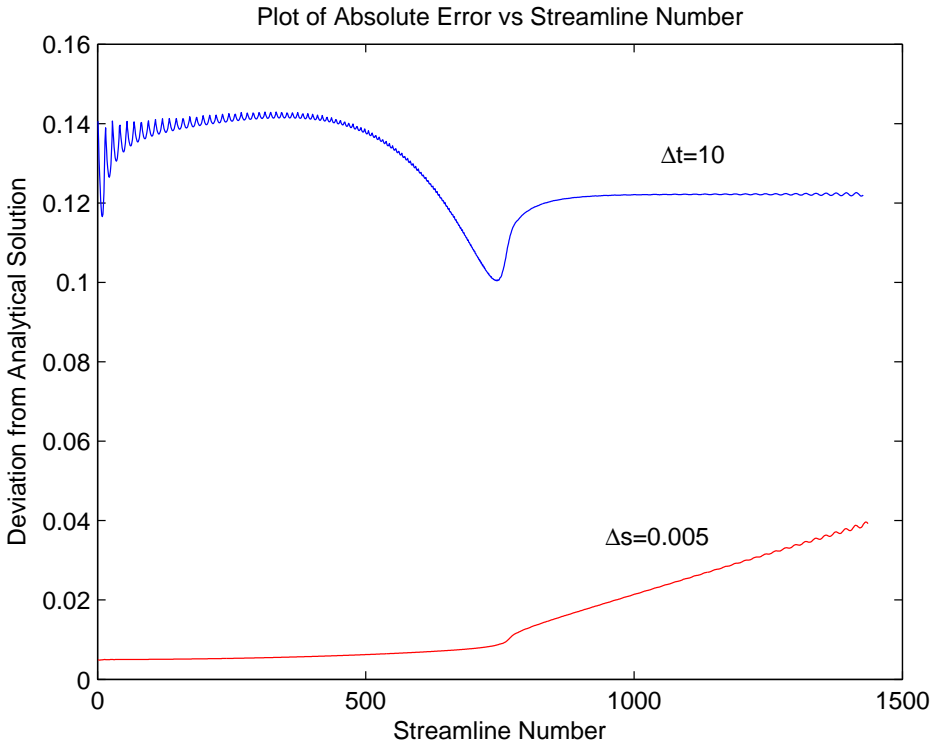


FIGURE 4: Deviation of  $\Delta t = 10$  and  $\Delta s = 0.005$  from the analytical solution.

the constant spatial step method is not constrained to a certain number of iterations for any particular isochrone, but will produce a solution in the number of iterations required to step to the final physical location.

For the examples considered in this article,  $\Delta t = 10$  and  $\Delta s = 0.005$  were chosen as they represent the maximum values at which the accuracy of the solutions started to decay. For  $\Delta t = 10$ , exactly 20,000 iterations were required to solve for the isochrone at  $t = 200,000$ , while just 3173 iterations were required for  $\Delta s = 0.005$  for the same isochrone. We calculated many other isochrones in this same flow domain for varying  $\Delta t$  and  $\Delta s$  values, with the constant spatial step method significantly more efficient in every case (Note that the errors were similar for both methods).

The relative improvements in time and accuracy of the constant spatial step method will vary based on the isochrone being solved, the values of  $\Delta s$  and  $\Delta t$ , and the characteristics of the velocity field throughout the domain. For some situations there may be little advantage in using the constant spatial step method at all. But as we have shown, when dealing with domains where any part of the velocity field is small, the constant spatial step method should perform significantly better.

In this article we have compared two different methods for integrating along streamlines to find the advection front of possible groundwater contaminants. Although different in their application, both techniques are based on simple Euler forward integration. For the purposes of this article, this simple integration technique was sufficient, but future investigation of other numerical integration methods may yet yield more accurate and/or faster results.

## References

- [1] J. R. Philip, Travel Times From Buried and Surface Infiltration Point Sources, *Water Resources Research* **20** (7) 990–994, 1984.  
[doi:10.1029/WR020i007p00990](https://doi.org/10.1029/WR020i007p00990) C1076

- [2] W. L. Powers, D. Kirkham, G. Snowden, Orthonormal Function Tables and the Seepage of Steady Rain through Soil Bedding, *Journal of Geophysical Research* **72** (24) 6225–6237, 1967.  
[doi:10.1029/JZ072i024p06225](https://doi.org/10.1029/JZ072i024p06225) C1076
- [3] W. W. Read, Hillside Seepage and the Steady Water Table I: Theory, *Advances in Water Resources* **19** (2) (1996) 63–73.  
[doi:10.1016/0309-1708\(95\)00034-8](https://doi.org/10.1016/0309-1708(95)00034-8) C1076, C1077, C1078
- [4] W. W. Read, Hillside Seepage and the Steady Water Table II: Applications, *Advances in Water Resources* **19** (2) (1996) 75–81.  
[doi:10.1016/0309-1708\(95\)00035-6](https://doi.org/10.1016/0309-1708(95)00035-6) C1076, C1077, C1078

## Author addresses

1. **D. P. Verrall**, Mathematics and Statistics, James Cook University, Townsville, Qld 4811, AUSTRALIA.  
<mailto:damien.verrall@jcu.edu.au>
2. **W. W. Read**, Mathematics and Statistics, James Cook University, Townsville, Qld 4811, AUSTRALIA.
3. **K. A. Narayan**, CSIRO Land and Water, Davies Laboratory, Townsville PMB, PO Aitkenvale, Qld 4814, AUSTRALIA.