Transient Groundwater Flow

Overview and Learning Goals

This lecture covers the finite different numerical solution for the one-dimensional (1D), transient, groundwater flow problem for a homogeneous confined aquifer. You will learn:

- The diffusion equation that governs transient groundwater flow
- How to discretize a 1D model into finite width cells
- How to use the finite difference method to numerically approximate 1st and 2nd order derivatives in space and time
- How to solve the transient groundwater flow problem in 1D using explicit time stepping and the incorporation of boundary conditions.

Background: The Finite Difference Method

The finite difference method is a numerical technique for approximating the derivatives that are found in differential equations. It is one of the main techniques used to solve partial differential equations in the physical sciences. Other common methods include the finite element method and the finite volume method, however, the finite difference technique is probably the simplest and most straightforward of these.

The finite difference approximation for a first order derivative of a function at the location $x = x_0 + \Delta x$ can be derived from a Taylor series expansion of a function:

$$f(x_0 + \Delta x) = f(x_0) + \frac{f'(x_0)}{1!}\Delta x + \frac{f''(x_0)}{2!}\Delta x^2 + ... \tag{1}$$

where we will ignore the higher order terms on the right side of the ellipsis.

Rearranging this equation so that the first order derivative is on the left hand side and dividing by Δx gives:

$$f'(x_0) = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} - \frac{f''(x_0)}{2!}\Delta x + ... \tag{2}$$

This equation shows that the derivative of the function at x_0 can be approximated from the difference of the function values at $x_0 + \Delta x$ and at x_0. The last term on the right hand side of the equal sign can be made negligible by choosing a very small value for Δx. This allows the derivative to be approximated using only the first term on the right hand side. It can help to visualize this by imagining a plot of a simple smooth function; pick two points on the function's plot curve that are close by. The difference of the function value at these two points, normalized by the spacing between them, will be close the derivative of the function.*(Note: this is exactly what you did for the first part of last week's homework assignment).* As you make the two points closer and closer together, this normalized difference will asymptotically approach the value of the functions's derivative.

This is the basis of the finite difference method. You simply approximate the derivatives in a partial differential equation by differencing nearby values of the function. We will see how this works in more detail later when we consider the groundwater flow equation.

Now consider a grid of n evenly spaced points with $x = [x_1, x_2, x_3, ..., x_n]$ and a grid spacing Δx. We can state the finite difference formula for a function defined on this grid as:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{\Delta x} \tag{3}$$

This is known as a *forward* finite difference formula since the derivative is approximated at the location x_i using its function value $f(x_i)$ and the next location to the right $f(x_{i+1})$.

A similar method is used to find the formula for the second order derivative approximation. We will skip the derivation and just state the formula:

$$f''(x_i) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{\Delta x^2} \tag{4}$$

This is known as a central finite difference method since it uses the function values on the right side at x_{i+1} and left side at x_{i-1}, in addition to the central value at x_i.

Note that while the examples above looked at the finite difference formulas for a function of spatial position x, they also apply to functions of other variables such as time t. In the transient groundwater problem below, you will use finite difference formulas for both position x and time t.

Background: Notation for Arrays Representing Functions

Since we will be working with arrays when programming the finite difference method, it is helpful to introduce new notation that simplifies the mathematical expressions.

1D Functions

We have already seen the array $x = [x_1, x_2, x_3, ..., x_n]$ which has n values. We can denote the i-th element of this array as x_i.

Now consider the function $f(x)$. We will use the notation f_i to represent the value of the function $f(x_i)$ at location x_i (i.e. $f_i \equiv f(x_i)$. In our Julia codes, we will have array f with values $[f_1, f_2, f_3, ..., f_n] = [f(x_1), f(x_2), f(x_3), ..., f(x_n)]$.

2D Functions

Suppose that our function now depends on two variables: spatial position x and time t and we note the function as $f(x, t)$. Defining a 1D array of m values of time $t = [t_1, t_2, t_3, ..., t_m]$, we can then denote the value of the function at the i-th position x_i and the j-th time t_i using the notation f_i^j, which is a more concise way of representing the function than using $f(x_i, t_j)$.

One-Dimensional Groundwater Aquifer Model

Before we get to the modeling problem, we need to introduce some definitions that will be used in our model setup.

aquifer A geological unit consisting of fractured rocks or a bed of grains that can both store and transmit water.

groundwater flow Fluid moves through an aquifer by transmission through the pore spaces between grains. This is also sometimes referred to a porous flow. Fluid can also move through fractures in a rock formation.

elevation head, z The elevation of the groundwater relative to some datum such as sea level. (units: m)

hydraulic head, h This is a combination of both the elevation head z and the pressure head, ψ which is the height the water would obtains above the elevation head due to pressure ($\psi = P/\rho g$, where P is pressure, ρ is density, and g is gravitational acceleration). Thus we have $h = z + \psi$.

Hydraulic head is a useful quantity for determining where water will flow (or not), since water will only flow from areas of high to low hydraulic head. For example, water flows downhill since the head decreases. Water in a spigot connected to the bottom of a water jug flows out since the head outside the water jug is less than that on the inside of the jug. If the pressure head is large water and a transmission path exists, water can flow upward to a height where the new elevation head equals the hydraulic head of the source.

For our 1D aquifer model, we define the following properties:

thickness d [m] The thickness of the aquifer unit in meters.

hydraulic conductivity K [m/s] The ease which which water can move through the pore spaces or fractures in the aquifer. K depends primarily on the permeability of the aquifer and the degree of saturation, but also the viscosity and permeability of the fluid.

specific storage S_s [m^{-1}] This property defines how much water the particular rock type can hold. It depends primarily on the porosity of the rock (i.e., how much empty space is there between grains). It is the mass of water that an aquifer releases per mass of aquifer, per unit decline in hydraulic head.

While K and S_s are properties that primarily are determined by the rock type, there are two related parameters that are formed by integrating these properties over the thickness of the aquifer b:

transmissivity T [m^2/s] $T = Kd$. This is similar to hydraulic conductivity but includes the aquifer thickness. Transmissivity is a measure of how much water can be transmitted horizontally.

storativity S [unitless] $S = S_s d$. This is similar to the specific storage but includes the aquifer thickness. Storativity is the volume of water released from aquifer per unit decline in hydraulic head, per unit area of the aquifer.

The upper panel in Figure 1 shows the initial state of the 1D model. The aquifer is 1000 m long and 100 m thick (here the aquifer is unconfined). Prior to the start of our simulation ($t < 0$), the head in the aquifer is fixed by the head value in lakes on each side of the aquifer. These boundary heads were such that initially the head in the aquifer is constant, $h = 100$ m, so there is no flow in or out through the side boundaries. At $t = 0$, head on

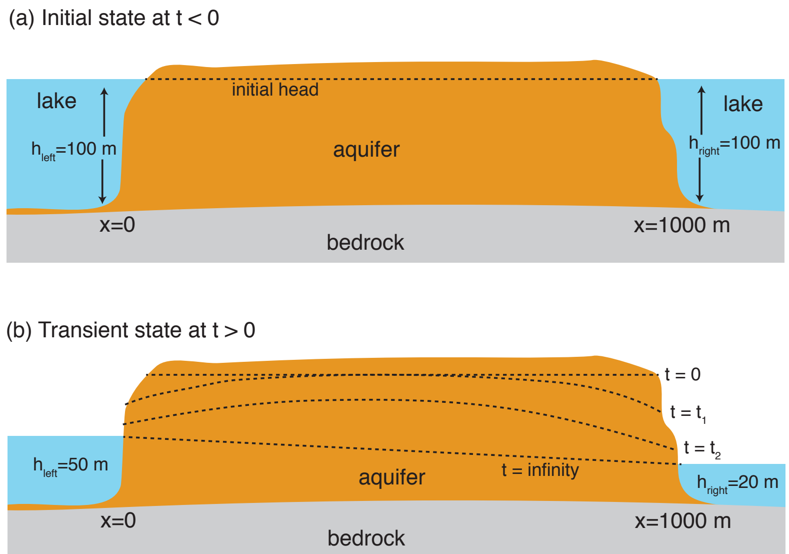

Figure 1: Groundwater aquifer model problem setup at (a) the initial state when $t < 0$ and (b) at $t > 0$ after a rapid change in the lake depths.

both sides of the aquifer drops instantaneously, to 50 m on the left and 20 m on the right (i.e. the water level in the lakes was quickly drawn down, perhaps to provide irrigation water to nearby farms). We want to know how the head changes in the aquifer with time, from t_0 to t_1 to t_2 and so on until the head is no longer changing.

We will assume the aquifer is unconfined, meaning there is no overlying confining layer. Further, we will assume the aquifer is homogeneous, meaning that K and S_s do not vary with position.

The governing equation for flow in this one dimension homogeneous system is:

$$\frac{T}{S}\frac{\partial^2 h}{\partial x^2} = \frac{\partial h}{\partial t}, \tag{5}$$

where ∂ is the partial derivative operator and the hydraulic head $h \equiv h(x,t)$. This equation is known as the **transient groundwater flow equation**. The derivation of this equation starts with Darcy's law and mass conservation equations; however, we will omit the derivation here and instead will concentrate on the finite difference numerical solution of the equation.

Before embarking on the numerical solution, it is usually helpful to consider the various terms in the differential equation. The term on the left hand side of the equation is the second order spatial derivative of the hydraulic head h. In other words, this is the spatial curvature of the hydraulic head. $\frac{\partial h}{\partial t}$ is the first order time derivative of the hydraulic head. Thus we have a first order time derivative equal to some coefficients multiplying a second order spatial derivative. This is known as a diffusion equation (i.e. those with a first-order time derivative and a second-order spatial derivative) and it describes the diffusion of some quantity through space as a function of time.

You may have already encountered a similar differential equation for heat flow, where h is replaced by temperature and S and T are replaced by terms for heat capacity and thermal conductivity.

For the groundwater flow equation considered here, the ratio T/S is known as the hydraulic diffusivity. If this ratio is large we can see that the time derivative of the hydraulic head will be large. This makes sense from intuition: since T is the transmissivity of the aquifer, it makes sense that the head will increase or decrease more quickly with T. We can also see that the time changes in the hydraulic head will be largest where the spatial curvature of the head is largest. *When you are looking at your results for the homework problem and wondering if they are correct, consider this last point carefully, especially for the early time steps in the model.*

This differential equation is called a homogeneous equation since there are no terms for sources or sinks. For example, if we had a well drilled into the middle of the aquifer and water was being extracted (say for drinking water or irrigation purposes), or water was being injected (say for waste disposal or hydraulic fracturing for oil exploration), the equation would need to include additional terms to describe these sources and sinks, and then the equation would be called an inhomogeneous equation.

Finite Difference Numerical Solution

Now we will numerically approximate the solution to the partial differential equation (5) using the finite difference method. Since this is a one-dimension problem, our modeling grid will consist of a 1D grid of cells arranged laterally as shown in Figure **??**.

We will divide our problem up into evenly spaced cells of width Δx, denoting the position of each cell as $x = [x_1, x_2, x_3, ...x_n]$. Within each cell we will solve for the evolution of the hydraulic head over time using evenly spaced values of time $t = [t_1, t_2, t_3, ...t_m]$. We will represent head at position x_i time step t_i using the notation h_i^j.

The explicit, finite-difference approximation for equation (5) is:

$$\frac{T}{S}\frac{h_{i+1}^j - 2h_i^j + h_{i-1}^j}{\Delta x^2} = \frac{h_i^{j+1} - h_i^j}{\Delta t}, \tag{6}$$

where the superscript j refers to the time t_j, and superscript $j+1$ refers to time t_{j+1} and

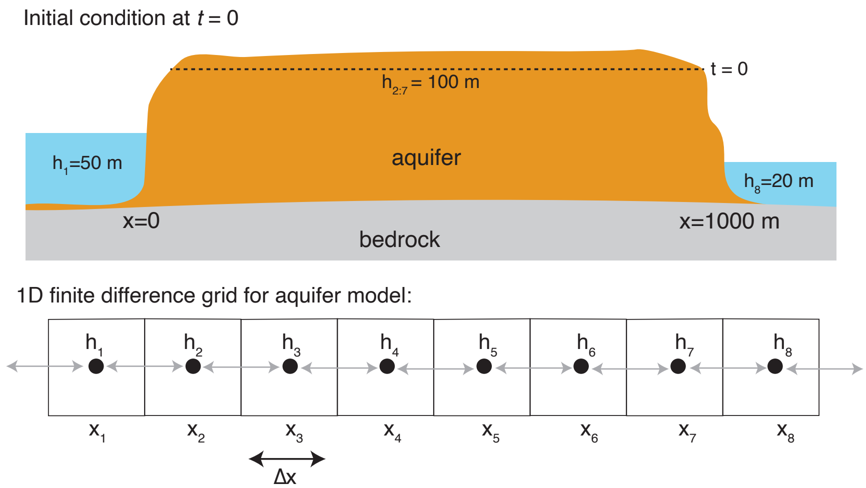

Figure 2: 1D finite difference grid example. Top panel shows the initial condition for the hydraulic head h at $t = 0$. Bottom panel shows a simple 1D finite difference grid approximating the aquifer using 8 cells. The cells have spacing Δx and gray arrows represent flow between cells. In this simple example, each cell has the same values for T and S. The left and right-most cells (cells 1 and 8) represent the lakes and are referred to as boundary cells. For the modeling problem in the homework assignment, the hydraulic head in these boundary cells should remain fixed through time. The interior cells (2 through 7) representing the aquifer will have transient groundwater flow that gradually draws down the water over time, with the water flowing out in the lakes on each side.

subscript i refers to the cell x_i. Subscript $i+1$ indicates the cell to the right, and subscript $i-1$ indicates the cell to the left.

Suppose that we have initial values for all values of h at time t_j; in other words, we know h_i^j for all values of j like in Figure 1 (a). Equation 6 can then be rearranged so that we can solve for the head at the next time step t_{j+1}:

$$h_i^{j+1} = h_i^j \left(1 - \frac{2T\Delta t}{S\Delta x^2}\right) + \frac{T\Delta t}{S\Delta x^2}\left(h_{i+1}^j + h_{i-1}^j\right) \tag{7}$$

Everything on the right-hand side (RHS) of the equation is known, being based entirely on head values from the time step t_j.

We can then solve for each value of h_i^{j+1} on the left-hand side based on the previous values of h at that location (h_i^j), as well as the previous values of h in the cells to either side of that location (h_{i-1}^j and h_{i+1}^j). This equation also includes the aquifer properties T and S, the cell width (Δx), and the length of the time-step (Δt).

Note that the constant terms in equation 7 can be grouped together into single constants since this will allow for a simpler version of the equation. Here we group these terms into new constants named a and b:

$$h_i^{j+1} = a\, h_i^j + b\left(h_{i+1}^j + h_{i-1}^j\right), \tag{8}$$

where

$$a = 1 - \frac{2T\Delta t}{S\Delta x^2}, \tag{9}$$

$$b = \frac{T\Delta t}{S\Delta x^2}. \tag{10}$$

Now we have all the pieces needed to create a finite difference code to model this problem. The key idea is that we will *iterate* through all desired time steps t, finding the values h_i^{j+1} at time step t_{j+1} for all the grid points $i = 1, ...n$, using the values h_{i-1}^j, h_i^j and h_{i+1}^j from the previous time step t_j.

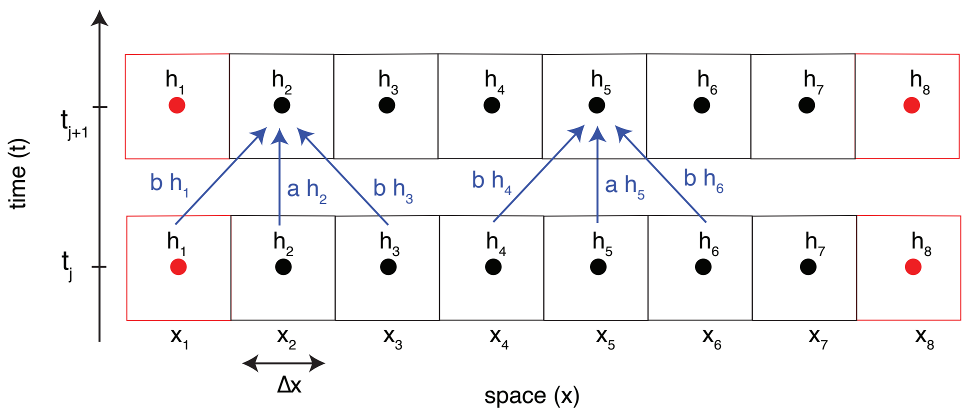

Figure 3: Example of applying the finite difference update in equation 8 to go from time t_j to t_{j+1}. The first and last cells are boundary cells where the hydraulic head should remain the same through time. The finite difference time-step update should be applied to the interior cells (2:8). A single time step update can be accomplish with a loop over the interior cell indices and applying equation 8 inside the loop. The blue arrows and text shows examples of applying equation 8 to compute the update h_2^{j+1} and separately the update h_5^{j+1}. Make sure you insert the updated values into the correct column representing t_{j+1} in your 2D array storing the values of $h(x, t)$.

Figure 3 graphically shows how a grid cell is updated from time step t_j to t_{j+1} by multiplying its value by a and then adding this to b multiplied by the neighboring values of h from time step t_j (and not t_{j+1}).

Using some mathematics to analyze the stability of the finite difference equation as a function of step sizes in time and space, it can be shown that the time-step Δt and grid spacing Δx size should be set so that:

$$\frac{T\Delta t}{S\Delta x^2} \leq 0.5 \tag{11}$$

For a given grid spacing and T/S ratio, this equation can be solved for the maximum time step possible for a stable and accurate solution. In other words, this equation can be use to ensure that the time step is small enough to be accurate using the finite difference approximation. Notice how it depends on both T/S and Δx^2.

In the homework assignment, you will set up a finite difference time domain code to step through equation 8 for a fixed number of iterations. Over time, the head values you calculate will approach the steady state solution (i.e. the solution when $\frac{\partial h}{\partial t} = 0$).

We can define an effective numerical steady-state as the time it takes for the head h to no longer be changing appreciably between time-steps. We can measure the change between time steps using the absolute value of the differences in h between the current and previous time-steps:

$$\delta_{h_i} = |h_i^{j+1} - h_i^j| \tag{12}$$

If the head is changing a lot between time steps, δ_h will be large, whereas once the solution is close to the steady state, δ_h will become negligibly small.

Your homework assignment is to solve the transient groundwater equation using the finite difference method and the initial conditions shown in Figure 2.