Mathematical Geosciences manuscript No.
(will be inserted by the editor)

Teaching transient groundwater flow modeling in a confined aquifer with spreadsheets and large language models

J. Jaime Gómez-Hernández · Daniele Secci

Received: date / Accepted: date

Abstract This paper extends our previous work on spreadsheet-based steady-state groundwater flow modeling to include transient conditions. Using a step-by-step Excel implementation, a numerical solution for transient groundwater flow in a confined aquifer is presented, highlighting the spreadsheet's educational value in understanding complex groundwater flow phenomena. Implementing transient flow in an easy-to-visualize and interactive way in Excel was not trivial and required the use of macros created with Visual Basic for Applications. Although the actions of the macros were simple to describe, their implementation was challenging. For this reason, we used ChatGPT as a coding companion, which provided the code after carefully designed prompts.

1 Introduction

Groundwater modeling is a fundamental tool for understanding aquifer behavior and managing water resources. While specialized numerical modeling software (such as MODFLOW) offers robust capabilities, the use of spreadsheet implementations in teaching and introductory research contexts has a long and compelling history.

One of the earliest references to spreadsheet-based groundwater flow modeling dates back to Olsthoorn's work in the mid-1980s (Olsthoorn 1985; 1999), in which the author used Lotus 1-2-3 to solve practical problems related to aquifer behavior.

PRIMA Programme supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 2222

J. Jaime Gómez-Hernández (✉)
Institute for Water and Environmental Engineering, Universitat Politècnica de València, 46022 Valencia · Spain, E-mail: jgomez@upv.es

Daniele Secci
Departament of Engineering and Architecture, University of Parma, 43124 Parma · Italy, E-mail: daniele.secci@unipr.it

Since then, other contributions have explored both steady-state and transient groundwater models in spreadsheets (Akhter et al. 2006 Bair and Lahm 2006 Fox 1996 Karahan and Ayvaz 2005 Elfeki and Bahrawi 2015 Niazkar and Afzali 2015), highlighting how this approach can streamline the learning process. Spreadsheets allow students to focus on hydrogeologic concepts and numerical fundamentals rather than on software intricacies.

In two earlier contributions (Gómez-Hernández 2022 Gómez-Hernández and Secci 2024), spreadsheet-based numerical solutions for steady-state groundwater flow were presented. Specifically, these earlier works demonstrated the solution for vertically-integrated flow in single-layer confined aquifers and then extended the approach to a broader range of cases, including unconfined conditions, anisotropic media, and multilayer systems in vertical cross-sections. In this manner, spreadsheets were shown to be a transparent and interactive means of studying groundwater flow problems, offering students and practitioners a direct glimpse into finite-difference approximations, iterative solution techniques, and boundary-condition handling.

The present work builds upon these ideas by introducing transient groundwater flow conditions into the spreadsheet modeling framework. Transient models capture an aquifer's dynamic response over time, a critical aspect for many real-world water management scenarios in which pumping schedules, recharge rates, or boundary heads vary in time. As before, our approach takes advantage of Excel's built-in iterative calculation features, allowing the user to solve implicit finite-difference formulations of the transient flow equation; however, as will be explained later, some coding is required to create a series of macros that will compute and store the head fields in time as they are being computed. A task for which we resorted to ChatGPT (OpenAI 2025) with carefully written prompts describing what had to be done. After all, the goal of this paper is to understand how groundwater flows, not to turn students and practitioners into full-fledged coders. That is where ChatGPT came in handy, generating the necessary Visual Basic macros in a flash, allowing us to focus on hydrogeology rather than on code debugging.

The paper is organized as follows: Section 2 briefly summarizes the essential elements from previous works, outlining the equations and spreadsheet implementation for steady-state conditions. Section 3 introduces the governing equations for transient groundwater flow and their discretization scheme. Section 4 describes the methodology for implementing these equations in Excel spreadsheets, including the novel approach of using ChatGPT to generate the necessary Visual Basic for Applications (VBA) code. Section 5 illustrates the setup and validation of the model through representative numerical examples. Finally, Section 6 discusses possible extensions, and Section 7 presents the conclusions, highlighting the educational implications of the transient spreadsheet model.

2 Recap of Previous Works

In this work, we continue working with the same confined aquifer as in Gómez-Hernández (2022), which is depicted in Fig. 1 and discretized into 19 rows by 33 columns of 100 m by 100 m as shown in Fig. 2. At this point, the only difference with that paper is that we have decided not to provide explicitly the transmissivities T of the three zones, but rather, the conductivities K, plus the elevation of the top and bottom of the layer, from which the layer thickness is computed and, subsequently, the transmissivity is derived. The reason for this modification is that, later, we will need the layer thickness to compute the storage coefficient as the product of the specific storage times the layer thickness.

In a previous study (Gómez-Hernández 2022), we considered steady-state groundwater flow in a confined aquifer governed by the continuity equation under assumptions of constant fluid density and vertically-integrated flow. The basic numerical solution was implemented using Excel spreadsheets, leveraging iterative calculations to handle circular references inherent in discretized aquifer systems.

Specifically, the general steady-state flow equation, discretized via finite differences for a generic cell, is:

$$h = \frac{T'_N h_N + T'_S h_S + T'_W h_W + T'_E h_E - W + N\Delta^2 + Q_{Riv}}{T'_N + T'_S + T'_W + T'_E}, \tag{1}$$

where h represents the hydraulic head at a generic cell, and T'_N, T'_S, T'_W, T'_E are harmonic averages of transmissivities at cell interfaces (the subscripts refer to orientation with respect to the current cell: north, south, east and west). Inputs include pumping or injection wells (W), infiltration ($N\Delta^2$, N infiltration rate, Δ^2 the square-cell area), and river inflow (Q_{Riv}). Boundary conditions included prescribed head (Dirichlet), no-flow (Neumann), and river interactions (Cauchy).

The method was subsequently extended (Gómez-Hernández and Secci 2024) to more complex aquifers, such as unconfined aquifers with variable transmissivities computed dynamically, anisotropic aquifers, and multilayer vertical cross sections. Each implementation showcased Excel's capacity for rapid calculation and visualization, ease of experimentation, and pedagogical clarity.

We refer readers to these earlier publications for detailed implementation procedures, intermediate calculation descriptions, and the complete spreadsheet architecture, since only the modifications introduced in the confined flow spreadsheet in Gómez-Hernández and Secci (2024) will be fully described here. The sheet naming convention changed between the 2022 and the 2024 papers to accommodate the peculiarities of Excel's iterative calculation and variable naming conventions; therefore, readers are urged to start from the 2024 paper if they would like to follow along with the implementation of the transient solution as described next.

In the following section, we discuss how the extension to transient flow conditions was implemented by incorporating storage coefficients and temporal discretization.

3 Transient Groundwater Flow Equations

Transient groundwater flow in two dimensions in a confined aquifer is governed by the following partial differential equation, which now includes a temporal derivative:

$$\nabla \cdot (T \nabla h) + Q_{ext} = S \frac{\partial h}{\partial t}, \tag{2}$$

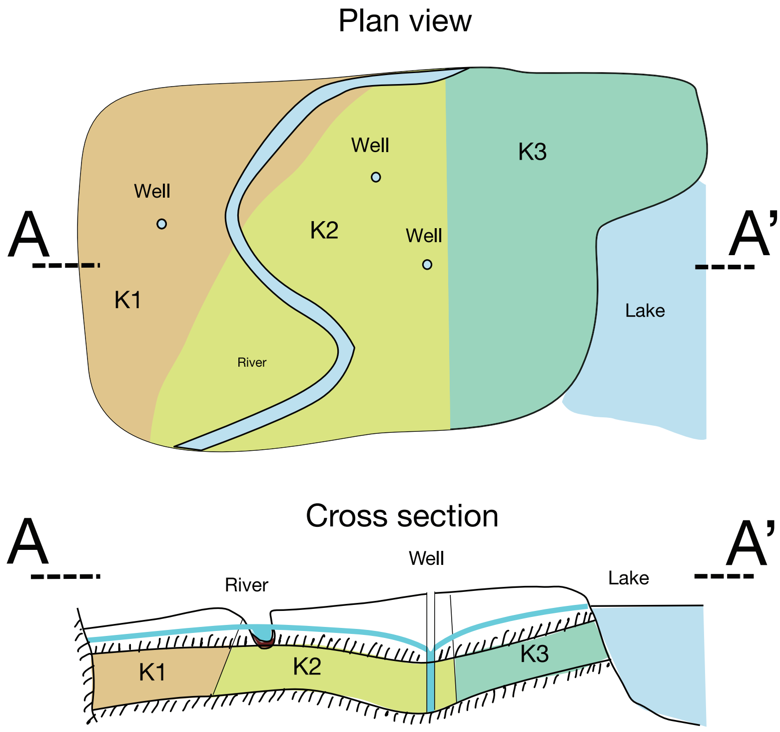


Fig. 1 Aquifer sketch in plan view and cross section.

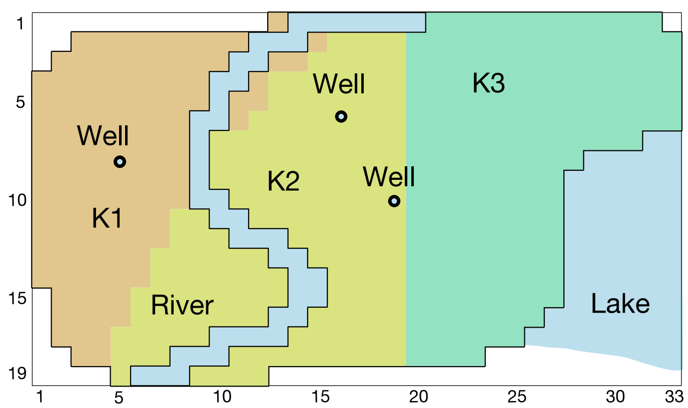


Fig. 2 Aquifer discretization

where:

- S is the storage coefficient [dimensionless], which equals the specific storage S_s [L^{-1}] times the layer thickness [L],
- T is the transmissivity tensor, which can be anisotropic [L^{2}T^{-1}],
- h is the hydraulic head [L],
- t is time [T], and
- Q_{ext} represents external stresses [LT^{-1}].

To solve Eq. (2) numerically, a finite-difference discretization in space and time is applied. For increased stability, an implicit-in-time discretization scheme will be implemented.

3.1 Implicit Discretization Scheme

An implicit discretization of Eq. (2) for each grid cell at time t can be expressed as:

$$\frac{T_N'(h_N^t - h^t) + T_S'(h_S^t - h^t) + T_W'(h_W^t - h^t) + T_E'(h_E^t - h^t)}{\Delta^2} + Q_{\text{ext}}^t = S\frac{h^t - h^{t-1}}{\Delta t},$$

where superscript $t-1$ refers to the heads computed at step $t-1$, and superscript t to the heads (and external stresses) at the current time step, with Δt equal to the time step size. The aquifer is discretized into equal-sized square cells of size Δ.

Rearranging terms in this equation, we arrive at an equation similar to Eq. (1)

$$h^t = \frac{T_N'\, h_N^t + T_S'\, h_S^t + T_W'\, h_W^t + T_E'\, h_E^t - W^t + N^t\Delta^2 + Q_{Riv}^t + \frac{\text{HCOEF}}{\Delta t} h^{t-1}}{T_N' + T_S' + T_W' + T_E' + \frac{\text{HCOEF}}{\Delta t}}, \quad (3)$$

where we have expanded the external inflows in the same three terms as before (wells, recharge, and river infiltration), adding a superscript t to explicitly state that they could be time variant. The term HCOEF [L^2] uses the MODFLOW nomenclature to refer to the quantity

$$\text{HCOEF} = S\Delta^2 = S_s b \Delta^2.$$

With the mathematical foundation established, we now turn to the practical implementation of these equations in an Excel spreadsheet environment.

4 Implementation details

4.1 From stationary to transient: moving ahead by Δt

By comparing Eq. (1) and (3), the modification to the Excel formula to compute the heads at any time is simple. There is only a need to add $(\text{HCOEF}/\Delta t)h^{t-1}$ to the numerator and $(\text{HCOEF}/\Delta t)$ to the denominator, provided that the HCOEF has been calculated and that the heads at the end of the last time step and Δt are known. For the first time step, there would be a need to provide the initial heads.

To achieve this, we must add a sheet with the specific storage coefficient (`A_SS`), a sheet with the initial heads (`A_h0`) and another sheet to store the heads from the last time step (`A_h_t-1`) to the Excel worksheet that solves the steady-state problem. After that, there is a need to compute HCOEF and store it in a new sheet (`B_HCOEF`).

The first step, before implementing the entire transient simulation, is to implement the calculation of h^t given h^{t-1}, knowing that the time step is Δt. For this purpose, we will introduce two named variables in spreadsheet `B_h`: `'iteration'` and `'Dt'`. When iteration equals zero, it indicates that we are restarting the simulation, and the heads should equal the initial heads (stored in the `A_h0` sheet). Variable `'Dt'` is the value of the Δt for the current step. The formula used to compute the heads (i.e., in cell `I5`) is modified, including a conditional query for iteration equal to zero, and with the inclusion of the additional terms in Eq. (3) with respect to Eq. (1). Then, this modification is extended to all cells within the aquifer.

The equation at cell `B_h!I5` is reproduced next

```
=IF(A_i!I5<>1,
    hundef,
    IF(A_hfix!I5<>"",
       A_hfix!I5,
       IF(Iteration=0,
          A_h0!I5,
          (B_h!I4*B_TN!I5+B_h!I6*B_TS!I5+B_h!H5*B_TW!I5
          +B_h!J5*B_TE!I5-A_W!I5+A_QInf!I5
          +B_QRiv!I5+B_HCOEF!I5/deltaT*'A_h_t-1'!I5)
          /(B_TT!I5+B_HCOEF!I5/deltaT))))
```

This expression reads as follows:

- If the current cell is inactive, set the piezometric head to value `'hundef'` else
  - If the cell corresponds to a prescribed head cell, set the piezometric head equal to the prescribed value in `A_hfix` else
    - If `'iteration'` is zero, set the head equal to the initial head in sheet `A_h0` else
    - Compute the head using Eq. (3)

Computing h^t in sheet `B_h` takes advantage of the iterative calculation capabilities of Excel when there are circular references. This requires the introduction of some iteration-stopping criteria. When the iterative calculation is activated in the Excel options, one must set the maximum number of iterations and a tolerance for the iterations to stop (in our example, we have used a maximum of 5000 iterations and a tolerance of 0.00001). Sometimes, these numbers are insufficient to achieve convergence, and the 'Calculate' label in the sheet footer must be clicked several times until convergence is reached. A good way to check for convergence is the global balance, which is shown to the lower left of sheet `B_h`. This global balance was already implemented for the steady-state solution, but now there is a need to add the variation in storage due to the change in heads between the beginning and the end of the iteration step. For this purpose, there is a need to create a new sheet `C_Storage` where the inputs into a cell due to the variation in storage are computed as

$$\text{C\_Storage} = \frac{(h^{t-1} - h^t) \cdot \text{HCOEF}}{\Delta t}.$$

In summary, to implement the transient solution for a single time step, the steady-state spreadsheets used in Gómez-Hernández and Secci (2024) must be modified as follows:

1. Add a spreadsheet with the spatially variable specific storage.
2. Add a spreadsheet with the spatially variable initial heads.
3. Add a spreadsheet with the heads at the beginning of the time step, h^{t-1}.
4. Add a spreadsheet to compute the HCOEF coefficient.
5. Add a named variable `'Dt'` in sheet `B_h`.
6. Modify the equation that provides the value of h to include the terms involving HCOEF in Eq. (3).
7. Add a spreadsheet to compute the input flows into each cell due to the variation in storage.

A list and recall of all Excel sheets needed is included in Table 1

4.2 Solving for multiple time steps: automation and data storage

The critical part of the transient implementation and the main contribution of this Teaching Aid is to perform the simulation over a certain time in multiple steps while storing the head solution for each time step and displaying the head time evolution at selected cells.

From a conceptual point of view, the solution is simple. First, the simulation period and the number of time steps to be used must be established. Following the same approach as in MODFLOW, a time-step multiplier is also defined so that the length of Δt increases as time passes. This approach allows a better approximation of the larger head changes expected at the beginning of the simulation. For this purpose, three named variables are defined in sheet `B_h`: `'SimTime'`, `'alpha'` and `'nsteps'`. These three values define a time sequence $\{t_0, t_1, \ldots, t_i, \ldots, t_{\text{nsteps}}\}$, where $\text{SimTime} = t_{\text{nsteps}} - t_0$, $\Delta_i = t_i - t_{i-1} = \text{alpha} \cdot \Delta_{i-1}$ and $\Delta_1 = t_1 - t_0 = \text{SimTime} \cdot (\text{alpha} - 1)/(\text{alpha}^{\text{nsteps}} - 1)$.

Second, we need to specify the cells for which we wish to display the head time evolution. We reserve an area in sheet `B_h` to define an array with the row, column, minimum head, and maximum head to be displayed for the observation locations. This array should have as many rows as observation points, and once it has been defined, it should be named as the named range `'Observation_indices'`. In the example, the `'Observation_indices'` named-range is `B_h!B16:E19` and contains four points being monitored. In addition, the range containing the aquifer heads should also be named as the named-range `'Aquifer_heads'`, which, in the example, is `B_h!I5:AO23`.

Table 1 Summary of Excel sheets used in the transient groundwater flow model

Sheet Name	Purpose
`A_h0`	Initial heads
`A_h_t-1`	Heads from previous time step
`A_i`	Active cells
`A_hfix`	Prescribed heads
`A_bottom`	Aquifer bottom elevation
`A_top`	Aquifer top elevation
`A_HK`	Hydraulic conductivity
`A_SS`	Specific storage coefficient
`A_W`	Pumping wells
`A_Qinf`	Infiltration rates
`A_hR`	River stage
`A_hB`	River bottom
`A_R`	River conductance
`B_Qriv`	Infiltrated river flow calculation
`B_h`	Current heads calculation
`B_HCOEF`	Storage coefficient calculation
`B_T`	Transmissivity calculation
`B_TN`	North intercell transmissivity calculation
`B_TS`	South intercell transmissivity calculation
`B_TW`	West intercell transmissivity calculation
`B_TE`	East intercell transmissivity calculation
`B_TT`	Sum of TN, TS, TW and TE
`C_QNorth`	Input flow through the north side calculation
`C_QSouth`	Input flow through the sourth side calculation
`C_QWest`	Input flow through the west side calculation
`C_QEast`	Input flow through the east side calculation
`C_Storage`	Storage variation calculation
`C_CellBal`	Flow balance calculation
`D_Intermediate`	Time evolution of heads at observation points
`D_heads`	Storage of head fields for all time steps

And third, there is a need to create two macros (also referred to as subroutines in VBA), one to reset the simulation to time zero and another one to advance one time step while saving the heads at the observation locations in the intermediate sheet `D_intermediate` and the spatial distribution of the heads at the end of the simulation step in sheet `D_heads`. These two macros, which will be described in detail in the next section, require using the VBA capabilities of Excel. Using VBA transforms Excel into a powerhouse capable of performing not only iterative calculations but also all kinds of complex calculations and data manipulations.

4.3 Visual Basic for Application coding using ChatGPT

The two macros needed close to 700 lines of code. To pretend that this is a Teaching Aid that requires one to write such an amount of code would be foolish. However, understanding our needs and describing what the macros should do to a large language model like ChatGPT does the job of writing the code with a few lines of natural language. The key is to provide the prompt with enough detail for ChatGPT to produce the needed code.

Nota Bene 1. The code generated and included in the provided spreadsheet was generated with version 4.o of ChatGPT as of February 2025. Given the rapid evolution of large language models and the fact that running twice the same prompt never gives the exact same answer, the VBA code resulting from running the code below may not be exactly the same as the ones in the provided spreadsheet, but, in principle, the resulting code should be functional, similar, and yield the same results.

Nota Bene 2. The handling of the VBA code, macros, and the linking to the buttons in the spreadsheet was done with Excel for MacOS v.16.97.2, so the details on how it is implemented in Windows or for other versions may be different. However, it was verified that the provided spreadsheet works well on both MacOS and Windows computers.

Nota Bene 3. While the macros are generated by a large language model like ChatGPT, users are still expected to have a basic understanding of coding principles to adapt or troubleshoot the scripts when needed. However, one of the strengths of this approach is the possibility of continuing the interaction with the large language model by describing what is not working or what needs to change, using clear, step-by-step instructions rather than professional programming terminology or perfect grammar. Minor typos or imprecise language usually do not compromise the outcome, as the model is capable of interpreting user intent. What truly matters is not linguistic accuracy, but a procedural description of the task at hand.

Prompt 1: Restart

The ChatGPT prompt used to create the macro that will reset the simulation is the following one.

> You are an expert in Excel. You can provide the simplest solutions to the most intricate problems in spreadsheet calculations, using formulas and Visual Basic for Applications macros. Please be concise and direct to the point. At the end, explain what you have done. Ask for additional information if you feel that the problem is not well described.
>
> 1. I need a VBA macro.
> 2. I want it to be very modular with multiple small subroutines to make small tasks. This would facilitate maintenance and modifications.
> 3. There will be a button that, when clicked, should trigger a 'Restart' subroutine that will do the following tasks:
>   (a) Make sure that all sheets, named variables, and named ranges mentioned below exist. If not, send a message and stop. Set calculation to manual.
>   (b) Fully clean worksheets 'D_Intermediate' and 'D_heads'. In 'D_heads' set column width = 5 and column height = 30.
>   (c) Remove all graphs in worksheet 'B_h'
>   (d) Set named-ranges 'time' and 'Dt' equal to zero. Force calculation.
>   (e) Retrieve the range of definition of named-variable 'Aquifer_head' and copy such range from worksheet 'A_h0' into 'A_h_t-1'
>   (f) Build a static array 't' with named-variable 'nsteps' components, where the first value is equal to named-variable 'SimTime' * (named-variable 'alpha' - 1)/('alpha'^ 'nsteps' - 1), this initial value represents the first deltat, this deltat will increase for each iteration in a geometric progression so that the following times are given by t(i+1)=t(i)+ (t(i)-t(i-1))*alpha, with t(0)=0

(g) Read named-range 'Observation_indices'. It contains as many rows as locations where to monitor the time evolution of piezometric heads and four columns: the row and column where to read h in named-range 'Aquifer_head' for time 'time', plus the minimum and maximum values for the heads in the scattergrams that will be built as the simulation progresses.

(h) Initialize sheet 'D_intermediate': in the first row, literal 't', followed by a literal in the form (row, column) for each observation in 'Observation_indices'; in the second row, '0.0', followed by the head retrieved for each observation location from range 'Aquifer_head'

(i) Create as many scattergrams as observation locations in sheet 'B_h' starting five columns to the right of the first empty column with x_range taken from the first column in 'D_intermediate' and rows 2 to 1+nsteps, and y_range taken from the corresponding column in 'D_intermediate' to the observation location and rows 2 to 1+nsteps. The scattergrams' vertical axis minimum and maximum values for each observation are given in the third and fourth columns of 'Observation_indices.'

(j) Call a subroutine that would write in the first empty row in sheet 'D_heads' a text 'Time =' time variable, and in the next empty row, it would copy the values from 'Aquifer_head.' Make sure that only the values are copied, not the formulas. Keep the same conditional formatting as the source 'Aquifer_head' range. Far to the right, create a mirrored copy along the horizontal axis of the 'Aquifer_head' values and then create a contour plot of these mirrored values next to the cells occupied by the initially copied range with a small vertical gap and with the same vertical span, for the contour plot use xlSurfaceTopView as the ChartType.

The prompt starts with an introduction establishing the large language model to act as an expert in Excel, and then it requests the creation of a VBA macro that should be written in a very modular way. There will be a button in the spreadsheet `B_h` that will reset the calculations. When this button is clicked, the actions described in the prompt are quite evident: (a) Check that all the sheets and named-ranges that will be used in the rest of the prompt exist; set calculation to manual so that there are no automatic updates while the macros are running and updating certain cells; (b) `D_intermediate` will contain the time evolution of the heads at the observation locations; this sheet should be cleared; and D_heads will contain the heads for each time step organized from the top of the page down, this should be cleared, too; it also sets the column width and the row height of `D_heads` sheet so that cells look approximately square; (c) the graphs that will be created with the head evolution at the observation location should be removed from sheet `B_h`; (d) initialize the values of the variables time and Dt to zero, since these are named-ranges, these values will be updated in head `B_h` where they were defined, then, force calculation so that the values of h are initialized to the initial heads h_0; (e) for $t = 0$ the values of h^{t-1} should be equal to the initial heads h_0; this task first retrieves the range where the aquifer is defined in sheet `B_h` (in our example, it is `I5:AO23`), and then copies this exact range from sheet `A_h0` into `A_h_t-1`; (f) build an array with the series $\{t_1, t_2, \ldots, t_i, \ldots, t_{\text{nsteps}}\}$ at which the simulation will be computed, it assumes that $t_0 = 0$ and explains how t_1 should be calculated, and how the rest of t_i are obtained; (g) it describes the content of the named-range `'Observation_indices'`, indicating that the row and column in the range are the indices of the row and column from where to retrieve the head in the named-range `'Aquifer_heads'` (they are not the absolute row and column in the sheet itself); it also indicates that the maximum and minimum values to be used for the head scattergrams are given in the `'Observation_indices'` range; (h) it initializes the header for sheet `D_Intermediate` with the literal 't' followed by literals with the row and column for which h^t will be displayed, and initializes the first data row with t equal to zero, and the heads retrieved from the `'Aquifer_heads'` range, which had already been initialized with the initial heads in task (b); (i) create the scattergrams to display the head evolution at the observation locations, these scattergrams should be to the right of the aquifer display in `B_h`, and the values to be displayed are in the columns of `D_Intermediate`, and (j) using a subroutine, detect the first empty row in `D_heads`, write the simulation time (zero in this case, since we are restarting), and copy the values of `'Aquifer_heads'` onto here; it is important to clarify that only the values should be copied, not the formulas, and by keeping the same conditional formatting, the shaded cells will be displayed accordingly; finally, to build a contour plot, it is necessary to create an intermediate array with the head array flipped upside down, with the first row at the bottom and the bottom row at the top, so that the displayed plot has the same orientation as the plan view with the computed values.

Running this prompt through ChatGPT provides a VBA code, which starts with the `Restart()` subroutine. Next, open the Visual Basic editor, which should be in the Macro submenu of the Tools Menu, and Insert Module. Copy and paste the generated code into the new Module.

Finally, we need to associate the Restart() subroutine with a button in `B_h`. For this, first activate the Developer tab in the Excel preferences under the View category. Go to the Developer tab, select the button, draw the button in sheet `B_h`, give it the text ‘Restart’, and finally, open the contextual menu by right-clicking on the button and choosing ‘Assign macro’, navigate to the module created and choose the routine `Restart`.

Prompt 2: Calculate next time step

The ChatGPT prompt used to compute the heads for the next time step is the following one.

> Next, I want a macro to associate with a button to perform a one-time step calculation. The macro should:
> 1. Make sure that all sheets and named-ranges mentioned below exist.
> 2. Increase named-variable ’iteration’ by 1. Set named variable ’time’ to t(flow_iteration) and named variable ’Dt’ to t(flow_iteration)-t(flow_iteration-1).
> 3. Copy the values in the range ’Aquifer_head’ from sheet ’B_h’ onto the same range in ’A_h_t-1’
> 4. Calculate sheet B_h.
> 5. Retrieve ’Observation_indices’ range. Fill the first empty row in ’D_intermediate’ with the time(flow_iteration), followed by the heads in the ’Aquifer_head’ range that correspond to the (row, column) pairs in the the first two columns of the ’Observation_indices’ range.
> 6. Use the same routine created in the previous prompt to draw the contour plot for the calculated heads.

The prompt starts by checking that all named variables and sheets used later exist, then it increases the iteration named variable by one, retrieves the time that corresponds to that iteration from the time array that was created when calling `Restart()`, assigns it to named variable `'time'`, computes the value of Δt and assigns it to named variable `'Dt'`. Next, before computing the heads for the current time, the heads from the previous time, which are still in `B_h` must be copied onto `A_h_t-1`, and then the calculation is forced (because of the way Excel works, it is necessary to use `Application.CalculateFullRebuild` for the calculation to ensure that all sheets are updated). After that, the prompt requests the retrieval of the heads at the locations indicated in the observation locations named range and the copying of the heads onto `D_heads` for storage purposes and to build a contour plot in a similar way as it was done in the previous prompt.

The code generated can be copied into a new Module in the VBA editor, and the routine `OneTimeStepCalculation()` assigned to a button labeled NextTimeStep in the same way as before.

Prompt 3: Restart and run the full simulation

With the previous two buttons, the transient simulation can be performed, and the evolution of the heads in the aquifer, as well as in the observation cells, can be followed in sheet `B_h`. For completeness, we added one more routine that restarts the simulation and performs all the simulation steps. This routine is generated by ChatGPT with the following prompt.

> Next, I need a macro that takes advantage of the previous ones to restart the simulation and then run all simulation steps.

This prompt produces the code that can be found in the VBA editor with the name `RunSimulation()`. Similarly, as before, a button is created with the label Restart+Run, which is associated with the `RunSimulation()` routine.

Finally, because contouring of the head fields in sheet `D_heads` requires some computing resources, we included a toggle button named ‘Build Contours’ to decide whether the contours should be built or not. Such a toggle button is created through the Developer tab in Excel. Once it is created and a text is associated with it, we have to look for its name, which appears in the box to the left of the formula bar when the toggle box is selected. In our example, the name is `‘Check Box 8’`. To prevent the contouring, we had to add a snippet that checks the status of the check box to the subroutine `WriteHeadsAndContour()` right after the heads have been copied and before the contouring has started.

In summary, to fully implement the transient simulations, we have to make use of Visual Basic for Applications for two purposes:

1. To be able to reset the calculations at any moment, which basically implies cleaning all intermediary sheets and objects that are created during the calculation.
2. Stepping in time, which requires keeping track of the current time step, extracting head values at observation locations after calculation, storing head values in an auxiliary sheet, and moving the calculated heads to sheet `A_h_t-1` before proceeding to the next step.

5 Example

The same confined aquifer used by Gómez-Hernández (2022) is used here, with the addition of two spreadsheets, one with the specific storage as shown in Fig. 3 and one with the initial heads as shown in Fig. 4.

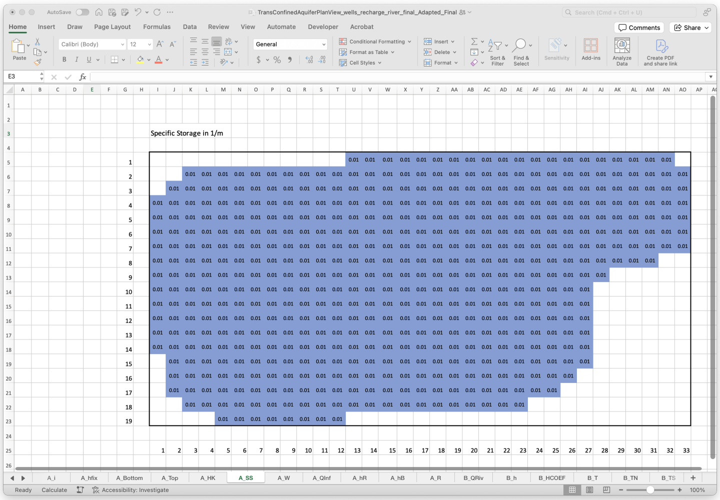


Fig. 3 Specific storage

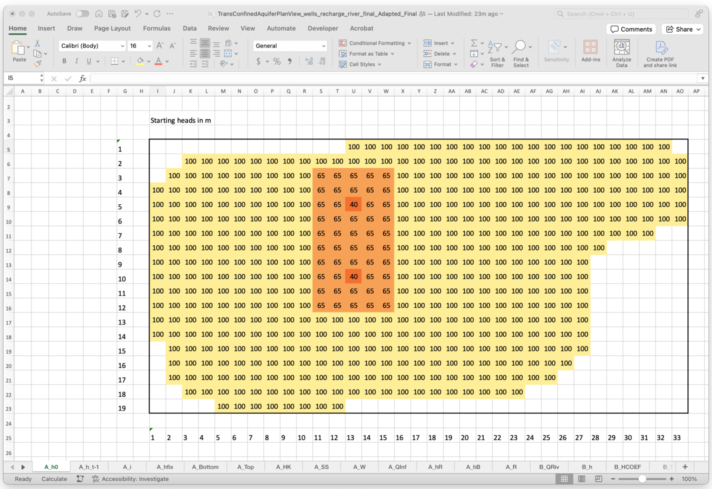


Fig. 4 Initial heads

5.1 Model Setup

Fig. 5 shows the sheet where heads are displayed as they are being calculated after resetting the simulation by clicking the Restart button. In the columns to the left of the aquifer, we have:

1. The values for named-variables `'Iteration'`, `'time'`, and `'Dt'`, which are updated by the VBA macros.

2. The value for named-variable `'delta'`, which is used to compute the cell area when needed, which must be set by the user.
3. The value for named-variable `'hundef'`, which is the numerical value assigned to the cells in the `'Aquifer_head'` range that are inactive and must be set by the user.
4. The value for named-variable `'minexchundef'`, which is the minimum value of the computed heads in the aquifer, excluding `'hundef'`. This value is used in the conditional formatting of the aquifer heads. It is calculated.
5. The named-variables `'SimTime'`, `'alpha'`, and `'nsteps'`, which must be set by the user and serve to identify the times at which heads must be computed and displayed.
6. The buttons for the NextTimeStep, Restart, and Restart+Run actions.
7. In light red, the range containing the information about the cells at which the piezometric head evolution will be displayed. There could be as many rows as desired. The user can add additional rows if needed. For each observation row, the values to be provided are the row and column of the cell in local coordinates (row 1 is the top row, column 1 is the leftmost column). In the example, row and column numbers are displayed in column G and row 25, respectively, for easy referencing, and the minimum and maximum values that will be used for the piezometry in the display. After the observation cells have been defined, the range should be selected and named `'Observation_indices'` for the VBA routines to work properly.
8. Finally, in light green, the different components of the global balance in terms of outflows. If the balance is correct, their sum, given in the Error row, should be close to zero.

Finally, on the right side of the sheet, there are the graphs for the piezometers, which at time zero only display one point corresponding to $t = 0$.

5.2 Simulation Process

A simulation period of 500 days, divided into 30 time steps of increasing size following a geometric progression of ratio 1.5, is considered. Clicking now 25 times on the button NextTimeStep advances the simulation 25 time steps until time 65.8 days.

5.3 Results and Analysis

Fig. 6 shows the piezometric distribution at this time. We can observe how the large initial depression of the initial heads in the center of the aquifer has almost disappeared. Piezometers close to this depression have gone up, and those who are outside have gone down. The global balance for this time step is perfect, with most of the water extracted through the wells coming from the storage, then from the lake, then from the infiltration, and a small portion from the river.

If we let the simulation run for a sufficiently long time, we can appreciate how the aquifer approaches steady-state, as can be seen in Fig. 7. At $t = 3000$ days, there

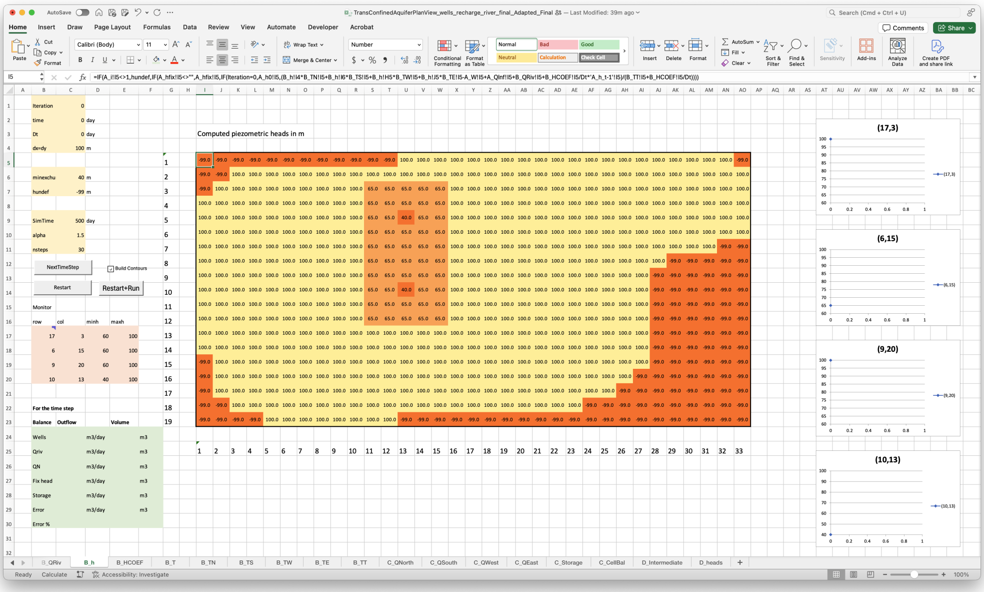


Fig. 5 Sheet B_h after simulation reset. This sheet shows the named variables, the calculated heads for the current iteration and the piezometric head evolution for the observation cells. The buttons that trigger the reset of the simulation, the advance to the next step, and the simulation run along the toggle box to compute the contour plots are also shown.

is still a small amount of water coming from the storage for the last time step of 1000 days, but it is less than 0.5% of the total withdrawn.

At any moment, sheet `D_intermediate` contains a table with the times and the piezometric head levels at each of the observation wells, and `D_heads` contains the head maps for all the simulation steps. Fig. 8 shows the piezometric head maps for times 0.27, 0.40, 0.60, 0.90, 1.35, 2.03, and 3.00 days.

6 Possible extensions

The first natural extension would be to include multiple stress periods, including the possibility of running a steady solution to set up the initial heads of the transient simulation. Although possible, it would complicate the already complex architecture of the workbook too much. There will be a need to create as many additional sheets with the items that could change between stress periods as there are stress periods (boundary conditions and external stresses, i.e., sheets `A_hfix`, `A_W`, `A_Qinf`, `A_hR`, `A_hB`, `A_R`), and then, programmatically, copy them into the sheets `A_hfix`, `A_W`, `A_Qinf`, `A_hR`, `A_hB`, `A_R` as the stress periods progressed. In a typical simulation in which we wish to obtain first the steady-state solution and then run two additional stress periods, it would imply the addition of 18 new sheets! The educational purposes of this Teaching Aid would disappear.

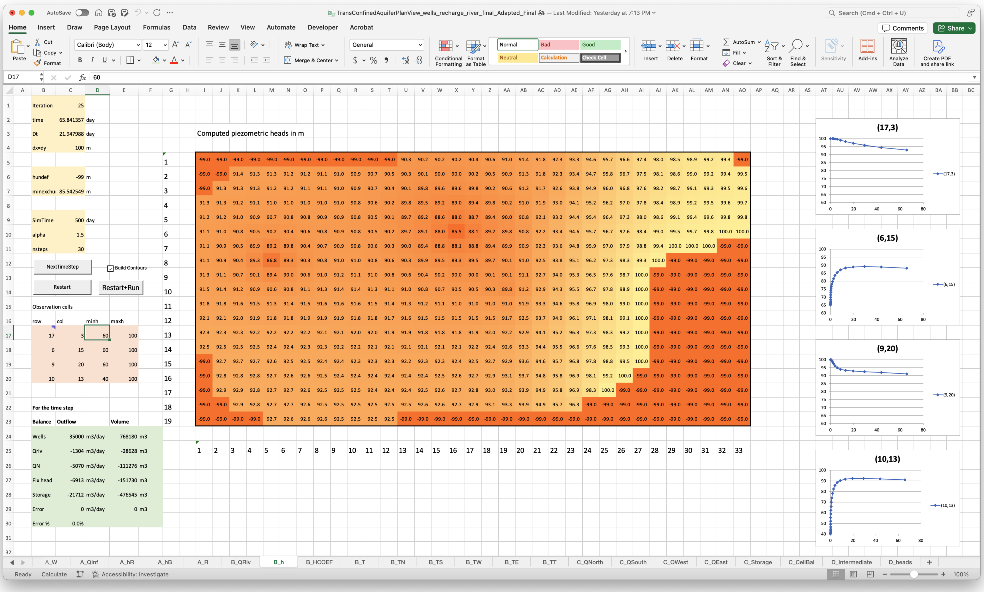


Fig. 6 Heads after 25 iterations, at $t = 65.8$ days

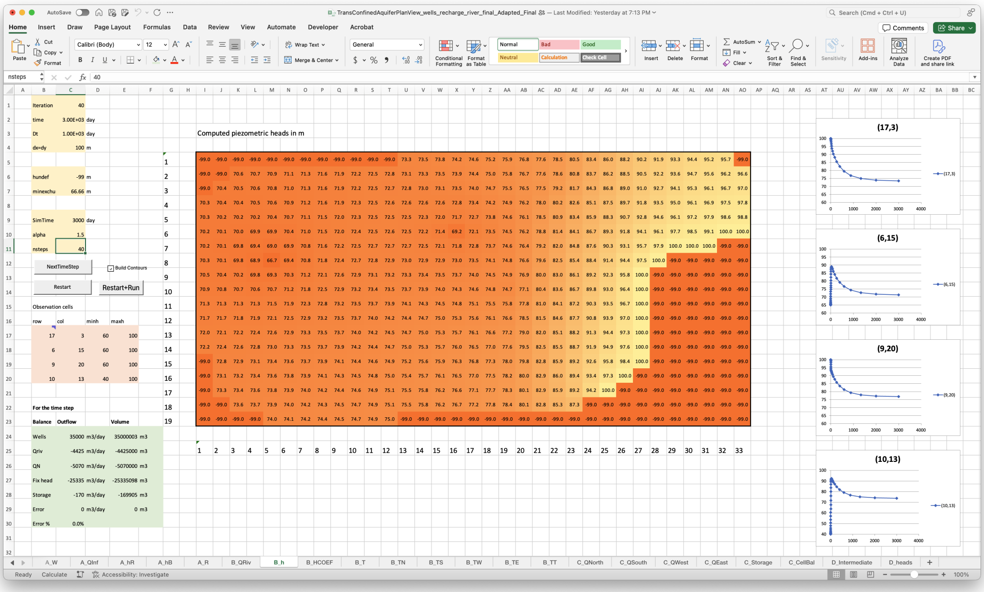


Fig. 7 Heads at $t = 3000$ days, approaching steady state

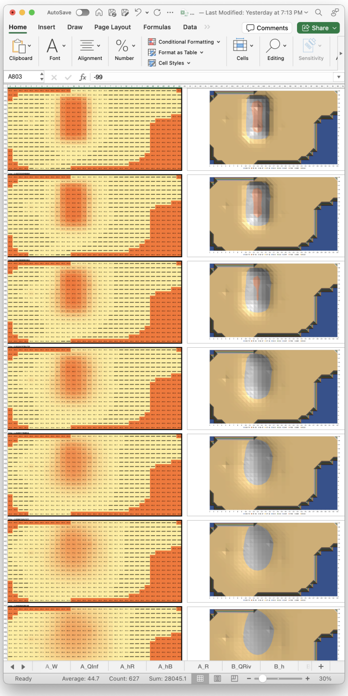


Fig. 8 A glimpse at sheet `D_heads` where the piezometric head maps are gathered and contoured. Maps shown correspond to times 0.27, 0.40, 0.60, 0.90, 1.35, 2.03, and 3.00 days

The modification of the provided spreadsheet for its application to an aquifer of a different size and shape is straightforward with the following considerations: (i) make sure that all user-defined input sheets start at the same cell (`I5` in the example) and cover the same area, (ii) for all computed sheets, make sure that the equation in cell `I5` is copied over the same range (in the example, this range is `I5:AO23`), and (iii) make the `'Aquifer_heads'` range match the aquifer (in the example `'B_h!I5:AO23'`).

Additional extensions could be to model unconfined aquifers and vertically-layered cross-sections as in Gómez-Hernández and Secci (2024).

7 Conclusions

In this Teaching Aid, we have demonstrated how to take advantage of the new actors in the coding arena: the large language models. Contrary to our previous two Teaching Aids, we have not focused so much on which sheets should form the workbook or which formulas should be used to compute the intermediate variables and the final results. Instead, we have distilled the steps that are needed to transform the steady-state spreadsheet into one that can simulate a transient problem. To perform this task, there is a need to first modify the formula to compute the heads accounting for the aquifer storage term, and to copy the just-calculated piezometric heads into the sheet that will hold the piezometric heads from the previous step, extract from this solution the values that should be later displayed, and make a copy of the heads elsewhere for a potential perusal later on. This shuffling of data across spreadsheets and the dynamic extraction of specific values could not be done with the standard Excel formulas, and we had to resort to using Visual Basic for Applications (VBA). VBA has a steep learning curve, but the use of ChatGPT with a good descriptive prompt outlining all the steps that are needed to update the spreadsheets—either to restart the simulation or to advance one time step—has resulted in a fast and easy solution that a non-expert in VBA could use. The spreadsheet provided is ready to demonstrate how the implicit solution of the transient groundwater flow equation could be implemented in Excel and to show the impact that initial heads, stresses, boundary conditions, or time discretization has in the transient modeling of groundwater flow.

Spreadsheet availability

The GitHub repository https://github.com/jaumegomez/GroundwaterFlowByExcel contains all the spreadsheets discussed in this paper and in the previous papers, along with the Python Flopy scripts that mimic the examples. The spreadsheets and scripts are released under GNU General Public License v3.0.

Conflict of interest

The authors declare no conflict of interest with regard to the content of this paper.

Acknowledgements The authors acknowledge project OurMED, which is part of the PRIMA Programme supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 2222.

References

Akhter M G, Ahmad Z, Khan K A (2006) Excel based finite difference modeling of ground water flow. Journal of Himalayan Earth Sciences 39:49–53

Bair E S, Lahm T D (2006) Practical problems in groundwater hydrology. Prentice Hall

Elfeki A M, Bahrawi J (2015) A fully distributed spreadsheet modeling as a tool for analyzing groundwater level rise problem in jeddah city. Arabian Journal of Geosciences 8(4):2313–2325

Fox P J (1996) Spreadsheet solution method for groundwater flow problems. In Subsurface Fluid-Flow (Ground-Water and Vadose Zone) Modeling, ASTM International

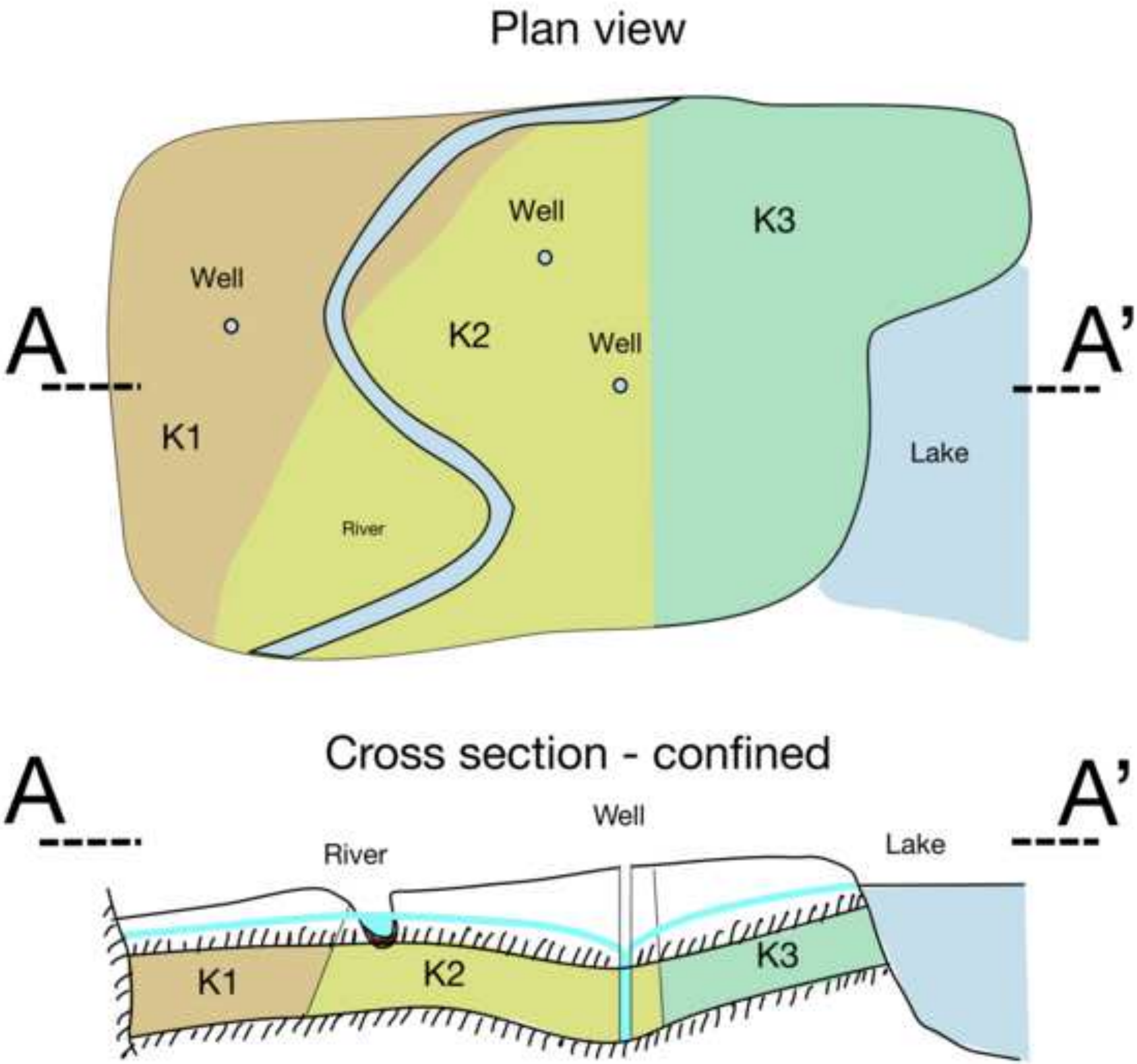
Gómez-Hernández J J (2022) Teaching numerical groundwater flow modeling with spreadsheets. Mathematical Geosciences 54(6):1121–1138

Gómez-Hernández J J, Secci D (2024) Teaching numerical groundwater flow modeling with spreadsheets: Unconfined aquifers and multilayered vertical cross-sections. Mathematical Geosciences 56(6):1355–1378

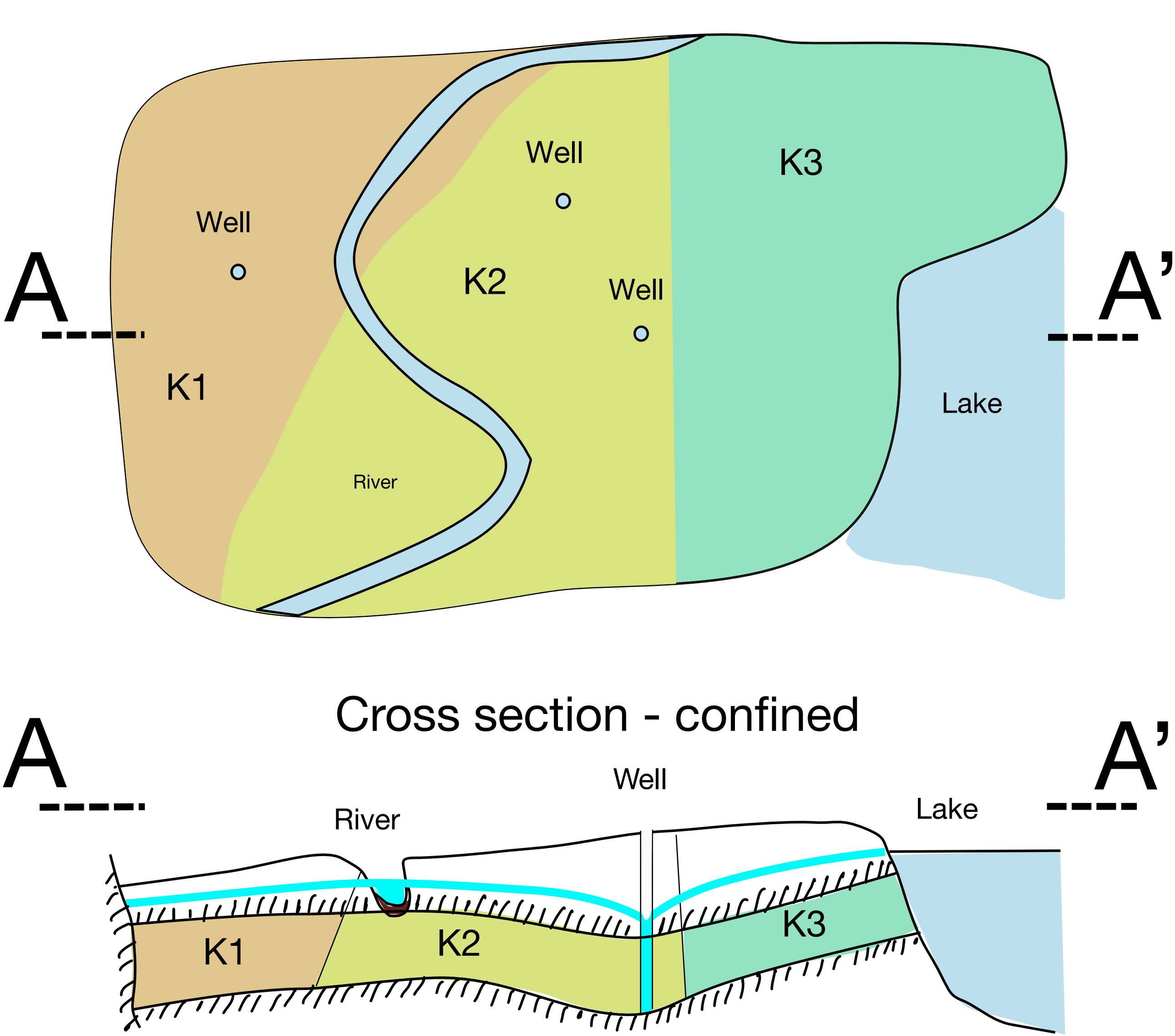
Karahan H, Ayvaz M T (2005) Transient groundwater modeling using spreadsheets. Advances in Engineering Software 36(6):374–384

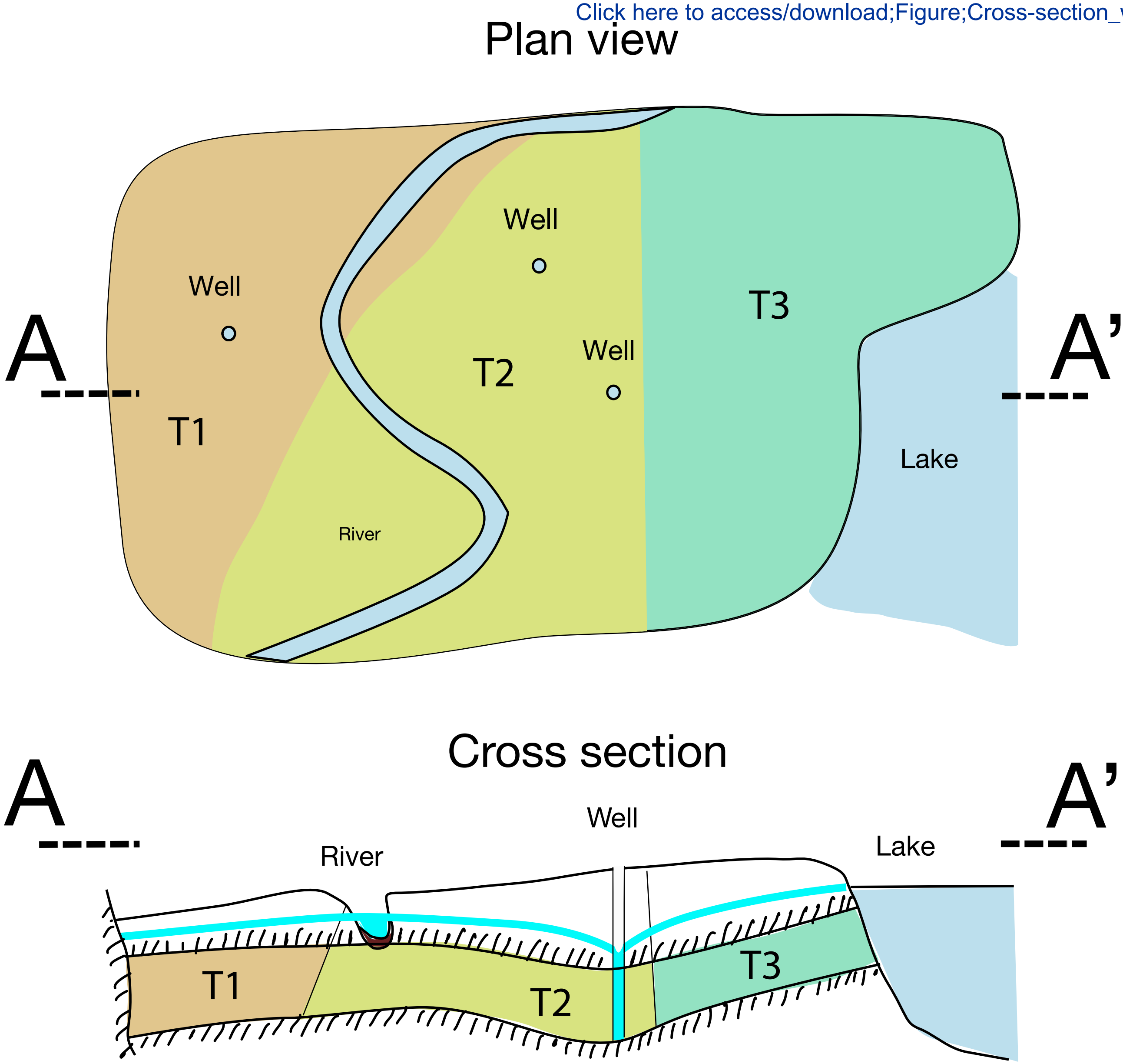
Niazkar M, Afzali S H (2015) Application of excel spreadsheet in engineering education. In First International and Fourth National Conference on Engineering Education, Shiraz University, 10–12

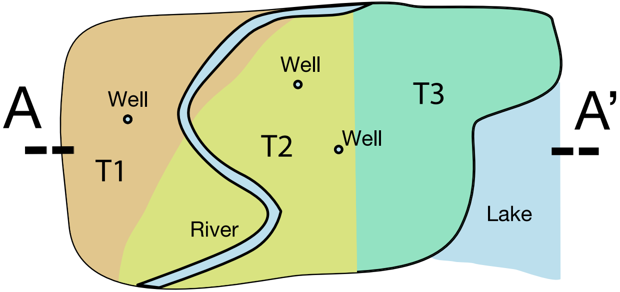
Olsthoorn T (1985) The power of the electronic worksheet: modeling without special programs. Groundwater 23(3):381–390
Olsthoorn T N (1999) Groundwater modelling: Calibration and the use of spreadsheets. Ph.D. thesis
OpenAI (2025) Chatgpt (january 2025)[large language model]


Plan view
A
A’
Well
Well
Well
K1
K2
K3
River
Lake
Cross section - confined
A
A’
Well
River
Lake
K1
K2
K3

Plan view


A
A’
Well
Well
Well
K1
K2
K3
River
Lake
Cross section - confined
A
A’
River
Well
Lake
K1
K2
K3


Plan view
A
A’
Well
Well
Well
T1
T2
T3
River
Lake
Cross section
A
A’
Well
River
Lake
T1
T2
T3


A
A'
Well
Well
Well
T1
T2
T3
River
Lake