https://doi.org/10.5194/egusphere-2024-3962
Preprint. Discussion started: 8 January 2025

HydroModPy: A Python toolbox for deploying catchment-scale shallow groundwater models

Alexandre Gauvain[1,2], Ronan Abhervé[1,3], Alexandre Coche[1], Martin Le Mesnil[1], Clément Roques[3], Camille Bouchez[1], Jean Marçais[4], Sarah Leray[5], Etienne Marti[5], Ronny Figueroa[3], Etienne Bresciani[8], Camille Vautier[1], Bastien Boivin[1], June Sallou[6], Johan Bourcier[7], Benoit Combemale[7], Philip Brunner[3], Laurent Longuevergne[1], Luc Aquilina[1], Jean-Raynald de Dreuzy[1]

[1] Univ Rennes, CNRS, Geosciences Rennes — UMR 6118, Rennes, France
[2] Laboratoire de Météorologie Dynamique (LMD), CNRS, Sorbonne Université, Paris, France
[3] Centre for Hydrogeology and Geothermics (CHYN), Université de Neuchâtel, Neuchâtel, Switzerland
[4] INRAE, UR RiverLy, Villeurbanne, France
[5] Pontificia Universidad Católica de Chile, Santiago, Chile
[6] INF, Wageningen University & Research, Wageningen, Netherlands
[7] Univ Rennes, Inria, CNRS, IRISA, Rennes, France
[8] Instituto de Ciencias de la Ingeniería, Universidad de O'Higgins, Rancagua, Chile

Correspondence to: Alexandre Gauvain (alexandre.gauvain.ag@gmail.com), Ronan Abhervé (ronan.abherve@gmail.com)

Highlights.

- HydroModPy is an innovative toolbox leveraging geospatial/geomorphological processing for advanced subsurface flow and transport modeling.
- HydroModPy facilitates the automatic collection of data and deployment of shallow groundwater models at catchment scale.
- HydroModPy is well-suited for both research and education, enabling the analysis of critical zone processes to address key water resource management challenges.

https://doi.org/10.5194/egusphere-2024-3962
Preprint. Discussion started: 8 January 2025

Abstract. In response to the growing demand for groundwater flow models, we present HydroModPy, an open-source toolbox designed to automate their deployment at the catchment scale. Built on top of the MODFLOW-enabling FloPy library, HydroModPy combines the robust WhiteboxTools toolbox for geospatial analysis and the well-validated MODFLOW code for groundwater modeling. This Python-based toolbox streamlines the construction, calibration, and analysis of unconfined aquifer models while adhering to FAIR (Findable, Accessible, Interoperable, and Reusable) principles. It enhances model reproducibility through editable Python code, supports multi-site deployment, and provides compatibility with alternative groundwater flow solvers. Furthermore, it integrates pre- and post-processing functionalities to simplify workflows. The toolbox enables catchment delineation and hydrological feature extraction from DEMs, followed by semi-automatic model construction and advanced visualization of hydraulic head and flow results. Users can choose from predefined aquifer structures and hydraulic properties such as exponential decay of hydraulic conductivity and porosity with depth or import complex 3D geological models. HydroModPy outputs can be exported in standard formats (e.g., raster, shapefile, netCDF), including water table elevation, water table depth, groundwater storage, groundwater-dependent hydrographic network and streamflow rates, and subsurface residence times. HydroModPy is tailored for the deployment in diverse geomorphological and hydrological settings, enabling the testing and exploration of aquifer models under varying recharge conditions. Its deployment capabilities are demonstrated in complex shallow basement and crystalline aquifers, where topography and geology primarily govern groundwater flow dynamics from hillslope to catchment scales. As an open-source toolbox, HydroModPy is designed for the community and actively encourages contributions from its users. It supports research in hydro(geo)logy and land and water management, while also providing valuable opportunities for teaching and education.

Keywords.

Watershed delineation; numerical modeling; groundwater flow; residence times; shallow aquifers; subsurface-surface interactions; catchment hydro(geo)logy; automatic python deployment

1. Introduction

Setting-up groundwater catchment-scale models is essential to study both quality and quantity of water resources. Nevertheless, it often requires the use of a significant number of software packages and libraries, not to mention the programming skills needed to link the various tools available. Even when it does not pose conceptual difficulties, model development continues to raise practical issues of time mobilization, simulation replicability and maintenance. Especially, the extension of a model developed in one site to other comparable sites is confronted, among others, with limiting development practices of local parameterization, manual transfers of information between successive software packages, lack of documentation, difficulty in identifying simulation stages and specific features. While using graphical user interfaces (GUIs) (Trefry & Muffels, 2007; Winston, 2009), first simplifies the development of models thanks to their high level of simplicity and intuitiveness, they limit the model replicability and transfer to other sites, the exploration of parameters, sensibility, and uncertainty analysis across various model areas. Shifting from the GUI to more systematic approaches is frequently a challenge by itself.

Several tools have been developed to address these technical difficulties (Bakker et al., 2016; Bakker & Kelson, 2009; Velásquez et al., 2023). They can be classified into three categories: those that facilitate execution, those that provide coupling and those that automate deployment. They use interpreted programming languages (Larsen et al., 2022; Stacke & Hagemann, 2021; Velásquez et al., 2023) such as R, Matlab or Python, which have gained popularity in the fields of science and engineering (Pérez et al., 2011). In the first category, Python tools provide interfaces to access existing advanced software. An example of this is FloPy, a set of Python scripts designed to run MODFLOW-related groundwater programs. In the second category, additional software offers additional functionalities. For example, GSFLOW and GroMoPo (Gardner et al., 2018; Zipper et al., 2023), propose an external surface-subsurface coupling (Crystal Ng et al., 2018; Guillaumot et al., 2022; Jing et al., 1989; Markstrom et al., 2008; Naz et al., 2023). There have been so far fewer developments in the third category for automatizing the deployment of models (Staudinger et al., 2019) to integrative development, replicability, and exportability of hydrogeological models.

While modeling tools provide a framework for developing hydrogeological models, the process of effectively applying it to a particular study site, remains complex and requires careful consideration of the assumptions, data, and methods used. Furthermore, the complexity increases when attempting to apply or deploy a model across multiple sites. To address these technical challenges in hydrology, we developed a new Python toolbox called HydroModPy, designed to automatically build, run, explore, and visualize the results of shallow catchment-scale hydrogeological models. We propose various approaches for implementing hydraulic properties, ranging from homogeneous to complex and heterogeneous fields (Orth et al., 2015). Thus, HydroModPy aims at enhancing our capacities to quantify geomorphologic, climatic and hydrogeological controls on the subsurface water cycle within a broad range of catchments. The toolbox relies on codes and open-source packages to perform three main tasks. The first step delineates catchment domains from a Digital Elevation Model (DEM) and provides extraction and discretization tools for groundwater flow modeling. Furthermore, in this initial step, functions are provided to extract data needed to constrain the hydraulic parameters (such as streamflow rates, piezometric measurements, geological maps, etc.) and

https://doi.org/10.5194/egusphere-2024-3962
Preprint. Discussion started: 8 January 2025

define climatic forcing, i.e. groundwater recharge from existing national and global databases. The toolbox implements standard procedures to set up and run batches of simulations across different catchments using standardized inputs. In the second step, it calls the groundwater software to encapsulate the setup and execution of hydrogeological simulations. The third step generates outputs in standard geospatial formats (such as GeoTIFF, netCDF, shapefile, etc.) and provides 2D/3D visualization tools to facilitate the exploration and comparison of model results.

This paper describes the framework of HydroModPy. First, we present the general workflow, its architecture and its five main components with their specific functions. Second, we demonstrate the deployment of HydroModPy at a regional scale across multiple catchment site models. Finally, we discuss the key features of HydroModPy, highlighting potential improvements for its future development for applications, as well as its potential for educational purposes.

2. Workflow and code description

HydroModPy is structured into five main components of (1) watershed extraction defining the model domain area, (2) model conceptualization and retrieving public/private data, (3) parametrization of the hydrogeological model, (4) computation of groundwater flows and particle tracking and (5) standardized outputs and visualization capacities (Figure 1). It relies as much as possible on existing, well-validated and broadly used python packages. General data structures are provided by *numyy* (Harris et al., 2020), *pandas* (McKinney, 2010) and *xarray* (Hoyer and Hamman, 2017). A Python-based API (Application Programming Interface) allows users to interact with the model through a well-defined interface. The API revolves around the *Watershed* object, which acts as a central orchestrator. This object is created by calling a static method: *watershed_root* (Code 1). Once the *Watershed* object is instantiated, users can call various methods on it to modify its state. The API is designed to be expressive, enabling users to define, manipulate, and solve hydrogeological models through simple method calls, making it accessible for both experts and non-experts in the field. For example, by calling methods on the *Watershed* object, users can configure forcing inputs, aquifer parameters, set boundary conditions, and run simulations, allowing for flexible and efficient modeling of different hydrological scenarios.

2.1. Watershed extraction defining the model domain area

HydroModPy allow to delineate the geographical boundaries of the model domain (geographic.py) extracting the catchment area using classical Geographic Information System (GIS) functions relying on WhiteBoxTools (Lindsay, 2016) complemented by other packages such as *GDAL* (GDAL/OGR contributors, 2024), *rasterio* (Gillies & others, 2013) to manage raster, and *geopandas* (Jordahl et al., 2020) to manage shapefile. HydroModPy defines the model domain area using one of the following sources: (1) a Digital Elevation Model (DEM) directly, (2) a shapefile to clip a specific area from a larger DEM, or (3) the outlet coordinates (XY) to extract catchment is from a sufficiently large DEM (Figure 2a). By default, HydroModPy adopts the projected coordinate reference system (CRS) of the input DEM. From this initial DEM, geospatial processing is performed using WhiteBoxTools (Lindsay, 2016), labelled WBT in the following. To extract the boundary of the watershed, the DEM is corrected by filling all local depressions and removing flat areas (WBT.FillDepressions) to ensure that the surface flow of water is continuous across each cell of the DEM. When the modelled domain is defined from the outlet coordinates, a flow direction raster (WBT.D8Pointer) and flow accumulation raster (WBT.D8FlowAccumulation) derived from the corrected DEM. To allow groundwater divides to extend beyond the catchment boundaries (Staudinger et al., 2019), a buffer (*geopandas*) zone can be added to the extracted catchment, enlarging the model domain. This buffer should increase with aquifer depth to accommodate longer groundwater flow paths. The catchment and model domain are finally generated and stored as raster and shapefile files (WBT.Watershed).

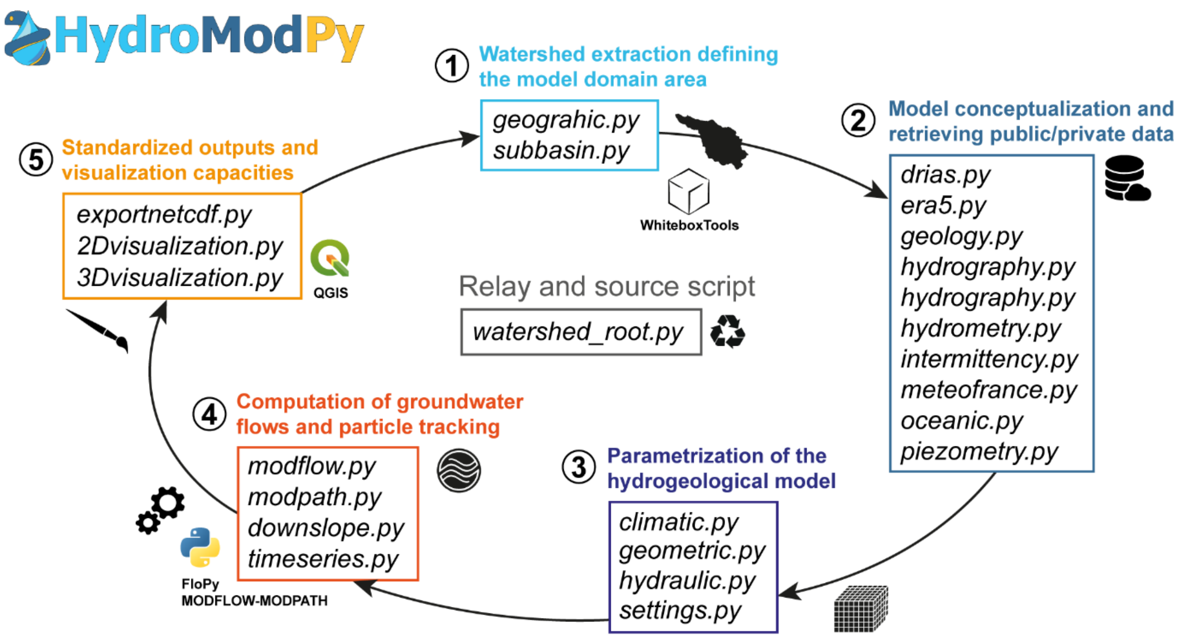


Figure 1. Global workflow of HydroModPy displaying the organization and the name of the Python files for handling 5 major stages of the toolbox.

2.2. Model conceptualization and retrieving public/private data

The groundwater model is built and constrained from generally accessible sources and databases or user's own data (BRGM, 2006; Dequesne & Portela, 2024; Le Moigne et al., 2020; Nowak & Durozoi, 2012). It includes the geological structures (*geology.py*), the hydrographic network (*hydrography.py*), streamflow rates (*hydrometry.py*) and stream intermittency (*intermittency.py*). These data can be used to constrain the model focusing on subsurface-surface interactions. The aquifer recharge (*climatic.py*) is included as source terms, sea level (*oceanic.py*) as a boundary condition for coastal aquifers, and piezometric levels are used to compare modelled to observed aquifer levels (*piezometry.py*). The study site data are retrieved as much as possible from national and global databases or added manually. They are expressed as timeseries or georeferenced maps within the modelled area to ensure accurate localization (Figure 2b). To sum up, HydroModPy facilitates the retrieval and the clipping of hydrology, piezometry, hydrometry, oceanic, climatic, and geological data belonging to the catchment of interest. By default, this is how a model is conceptualized and built in HydroModPy:

Spatial discretization is defined by the resolution of the DEM as a structured mesh or can be adjusted by the user. The depth discretization is defined by the number of layers set by the user.

Boundary conditions. The DEM defines the upper boundary of the model. The water table height is limited by the surface potentially triggering seepage. By default, seepage is not re-infiltrated into the aquifer and considered as either runoff or direct contributions to streams. Without any other information, no-flow boundary conditions are applied to the sides of the modelled domain. Constant hydraulic head can be imposed at prescribed domain limits to represent the boundary condition imposed by an ocean/sea/lake. Modelled cells where the elevation of the DEM is lower than the imposed hydraulic head are considered as fixed head boundary conditions.

Initial conditions. The initial state is taken as the steady state of the system, constrained by the mean of the recharge chronicle or an imposed value.

Temporal discretization. The recharge timeseries determines the temporal discretization of the model. An option is available to downscale (e.g. daily to monthly) the recharge timestep to reduce the computation time.

Recharge (Figure 2c). Recharge is assumed uniform over the watershed and operates on each cell of the model at the top of the water table. Distributed recharge can also be applied thorough a raster or a NetCDF file. The model is run in steady state when a single value is provided for the recharge.

2.3. Aquifer parametrization of the hydrogeological model

Input format and structures enable parameter explorations and sensitivity analysis to facilitate model calibration. Main parameters are the hydraulic properties of the aquifer and its geometry. By default, this is how a model is parameterized in HydroModPy, with all parameters remaining fully customizable by users:

Hydraulic properties. The hydraulic conductivity K and, in case of transient simulations, the storage coefficients S of the aquifer (specific yield Sy and the specific storage Ss) are set by default uniform and isotropic over the modelled domain.

Aquifer geometry. Model thickness can be defined as constant, with an aquifer bottom parallel to the topography, or variable with an aquifer bottom at defined altitude (flat bottom or bottom elevation fixed for each cell informed by a raster given as input). The model thickness is discretized according to the number of layers set by the user. The thickness of the layers can be either constant or variable (increasing exponentially with depth).

Heterogeneity structure. Lateral heterogeneity of the hydraulic properties can be specified through a mask representing the geological model (De La Varga et al., 2019; Los Alamos National Laboratory, 2016) or other structures. Depth-dependent hydraulic properties can be incorporated by assigning different values to each model layer, such as implementing an exponential decrease, which is easily parameterized by the user.

2.4. Computation of groundwater flows and particle tracking

Within its modular structure, HydroModPy can include other computational methods and software (Figure 2d). It is currently based on MODFLOW-*NWT*, a Newton-Raphson formulation for MODFLOW-2005 (Harbaugh, 2005; Niswonger, 2011) through the library FloPy (Bakker et al., 2016; Hughes et al., 2023) (*modflow.py* and *modpath.py*). HydroModPy is especially suited for shallow unconfined aquifers. It particularly contributes to quantifying groundwater contributions to streams through the analysis of baseflow dynamics and spatio-temporal distribution of the hydrographic network. Currently, based on MODFLOW, recharge is directly transmitted to the watertable, and the unsaturated zone is not simulated. In HydroModPy the fully convertible layer type is applied, i.e. a cell is confined if the overlying cell contains groundwater and, otherwise, unconfined. For a confined (resp. unconfined) layer, the storage coefficient is the vertically integrated specific storage (*Ss*) (resp. specific yield *Sy*). When recharge input of the model in negative, the evapotranspiration package (*EVT*) is activated at the first layer with its default settings, assuming direct evapotranspiration from the water table.

Computations provide watertable levels, seepage flows, groundwater flows and groundwater storage. Using the *DRAIN* package of MODFLOW, seepage areas result from the water table fluctuation and interception with the surface (Anderson et al., 2015). Surface flows resulting in a continuous hydrographic network from seepage pixels are computed using surface accumulation fluxes following the steepest topographic gradient (downslope.py). Furthermore, particle tracking with MODPATH suite (Pollock, 2012, 2016) provides information on the organization of subsurface flow paths and associated residence times.

2.5. Standardized outputs and visualization capacities

Output results. HydroModPy stores input data, model domain characteristics and simulation results in standard formats. The results are automatically stored in a designated directory by the user. Two folders are generated: “results_stable” for the model area data and “results_simulations” for hydrogeological simulation data. Input data and model output are structured in classical file formats: raster (*.tif*), netCDF (*.nc*), text file (*.csv*), and shapefile (*.shp*) to facilitate the use of visualization software like QGIS (QGIS Development Team, 2024) or PARAVIEW (Ahrens et al., 2005). This ensures seamless analysis and interpretation of hydrological model outputs across different platforms and applications. The outputs of hydrogeological models include watertable elevation, watertable depth, seepage areas, seepage outflow, groundwater flux, groundwater storage,

https://doi.org/10.5194/egusphere-2024-3962
Preprint. Discussion started: 8 January 2025

persistency index (spatio-temporal water occurrence) and accumulation flux (more detailed in the documentation). Additionally, a *Watershed* object is created as a "*python_object*" to store all geographic and input data providing a track of the whole model parametrization and therefore facilitating reuse for further analyses. This object avoids to re-extract watershed information during step 1 of HydroModPy, which helps to run efficiently several hydrogeological models.

Visualization of model data and model results can be performed in 2D using *matplotlib* (Hunter, 2007) (*visualisation_watershed.py* and *visualisation_results.py*) and in 3D visualization using *vedo* (Musy et al., 2022) standard *.vtk* files (*export_vtuvtk.py*). In 2D, users can map the location of the watershed in the regional DEM with *visualization_watershed.watershed_local*, the topography of the watershed with *visualization_watershed.watershed_dem*, and the geology of the watershed with *visualization_watershed.watershed_geology*. Additionally, model characteristics and results can be mapped over the catchment (Figure 2e-i, *visualization_results.visual2D* function). It includes the topography and model grid, the water table levels and the water table depth. Subsurface pathlines can also be plotted over the catchment as the trajectory of the injected particles. Furthermore, HydroModPy provides the ability to map groundwater seepage flows and the accumulated surface flows as well as the residence times of the groundwater discharging to the stream. These figures are created using the Python packages *matplotlib* for the organization of the figures and the visual aspect, *geopandas* to manage the shapefiles, *rasterio* to manage the rasters and *xarray* to manage the NetCDF. An interactive exploration of piezometric levels at each point of the model is also possible thanks to the development of the interactive tool *visualization_results.interactive_cross_section*. Similarly, using *.vtk* files, the *visualization_results.visual3D* function leverages the *vedo* package to provide interactive 3D representations of features such as topography, water table levels, water table depth, surface flow, groundwater seepage flow, pathlines, and residence times.

https://doi.org/10.5194/egusphere-2024-3962
Preprint. Discussion started: 8 January 2025

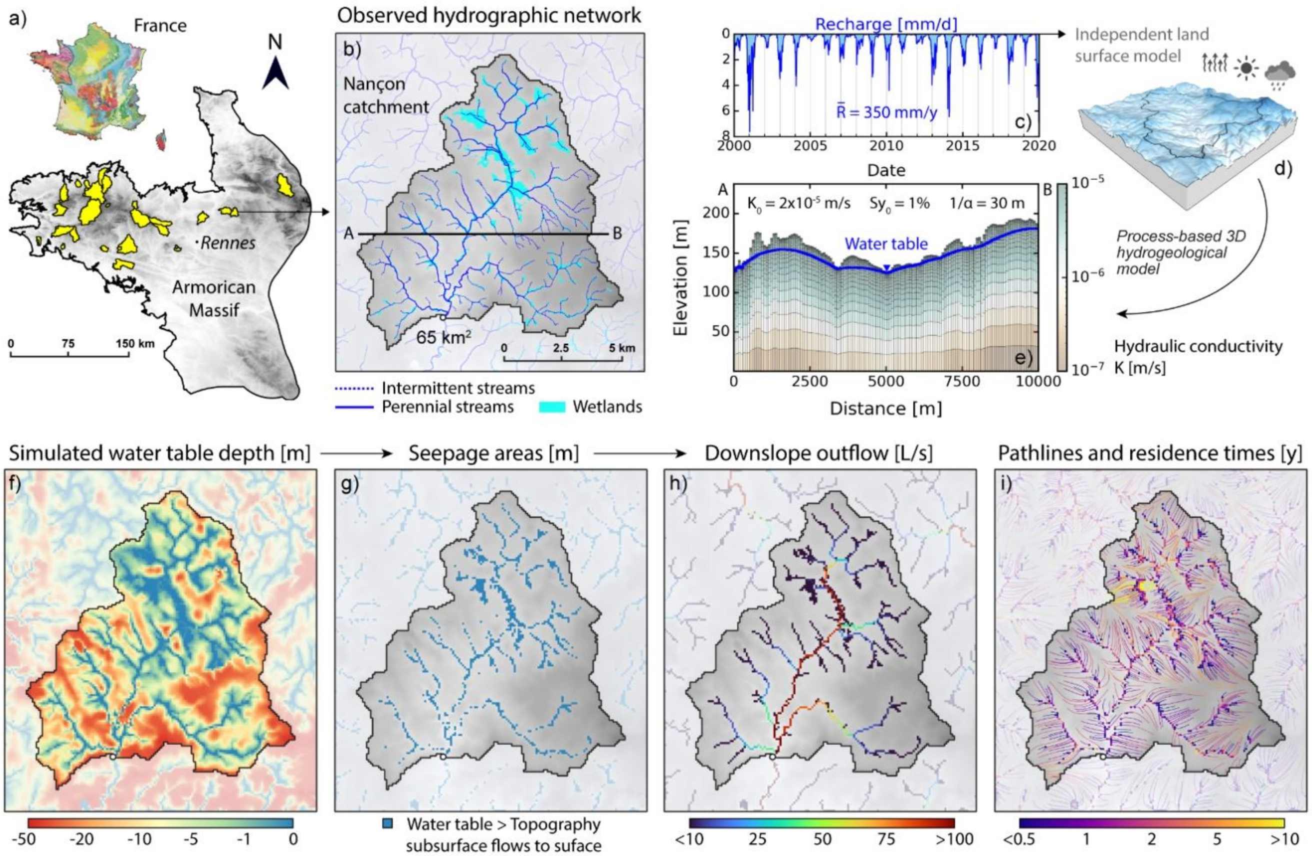


Figure 2. HydroModPy modeling steps illustrated on the site of Nançon. (a) Extraction of the watershed from a regional DEM. (b) Clip data based on the watershed extent. (c) Recharge time series provide from an independent land surface model. d) 3D diagram illustrating the model conceptualization and parameterization based on data and assumptions. (e) The cross-section (A-B) illustrates the vertical grid discretization and the resulting water table. The parameters include an exponential decay of *1/α* (m) with depth from the maximum hydraulic conductivity K_0 and specific yield Sy_0 (%) in the first layer. (f-i) 2D map view visualization displaying spatial data and model results in steady state across the study area (left to right): watertable depth, seepage areas, accumulated outflow, pathlines and residence times.

3. Deployment on multiple sites

3.1. Methodology and calibration

Groundwater flow models were built with HydroModPy on 32 unconfined aquifers located in the Normandy and Brittany regions of France (Figure 3a), two of them being published (Abhervé et al., 2023; Le Mesnil, et al., 2024). For each catchment, the model domain was determined by the catchment outlet coordinates (Table 1) using the 75 meters Digital Elevation Model

https://doi.org/10.5194/egusphere-2024-3962
Preprint. Discussion started: 8 January 2025

(DEM, Figure 3a) sourced from BD ALTI® (IGN, 2011). The aquifer top was set equal to the topography and the aquifer bottom was set at 30 meters below the land surface, representing the typical depth of the interface between the weathered, fissured, or fractured zone and the fresh bedrock in Brittany (Cornette et al., 2022; Dewandel et al., 2012; Mougin et al., 2015; Roques et al., 2016). All models have a single layer. The recharge R was extracted from the SAFRAN-ISBA model (Le Moigne et al., 2020) and was assumed homogeneous over the domain. Transient simulations were carried out over 3 years at monthly time steps. The hydraulic conductivity K and specific yield Sy were assumed homogeneous. Each model was calibrated using the observation of the perennial stream network (Abhervé et al., 2023), the stream intermittency (Nowak & Durozoi, 2012) and streamflow rates following the methodology of Abhervé et al., 2024. The hydraulic conductivity K was first determined by calibrating the steady-state simulated stream network on the perennial stream network given by BD TOPAGE (IGN, 2020), using a dichotomy method performed on K initially range between 10^{-9} and 10^{-2} $m.s^{-1}$ (Figure 3a) (Domenico & Schwartz, 1998; Freeze & Cherry, 1979). The relevant simulation was obtained by calibrating the model to minimize the D_{optim} criteria given by Abhervé et al. (2023). The distance D_{optim} is defined as the average of D_{SO} and D_{OS}, where D_{SO} represents the average distance from the simulated stream network pixels to the nearest downslope observed stream network (Figure 3b), and D_{OS} represents the average distance from the observed stream network pixels to the nearest simulated stream network (Figure 3c). The smaller the value of D_{optim}, the better the match of the simulated seepage pattern and the observed stream network. The specific yield Sy was then calibrated comparing the simulated total streamflow at the catchment outlet with the measured data, using computing the Nash and Sutcliffe Efficiency criteria $\mathrm{NSE}_{\log}$ (Nash & Sutcliffe, 1970; Oudin et al., 2006) (Figure 3e). The best model was selected from a set of 10 values, regularly spaced, explored within the range of 0.1% to 10% (Figure 3d). The consistency of the saturation (pattern of seepage areas) and the stream intermittency (Figure 3f) are used for validation to confirm the optimal Sy value obtained through the $\mathrm{NSE}_{\log}$ on streamflow.

Within HydroModPy, this methodology can be applied with only a few lines of code as shown below (Code 1). A *for* loop has been implemented to explore the outlet coordinates of the 32 catchments, build groundwater model, set parameters, run simulation and generate model outputs.

Code 1. Example of a conceptual script for running a model with HydroModPy on two different catchments (Nançon and Canut).

```python
import hydromodpy.src as hmp

outlet_coord_list = [['Nançon', 389358, 6816630], ['Canut', 327811, 6777901]]
regional_dem_path = 'C:/User/Europe_SRTM30m.tif'

for catchment_site in outlet_coord_list:
        #1 - Watershed extraction defining the model domain area
        W = hmp.watershed_root(regional_dem_path, outlet_coord_XY=catchment_site)

        #2 - Model conceptualization and retrieving public/private data
        W.climatic.update_recharge(R=10, time_series=False)

        #3 - Parametrization of the groundwater flow and transport models
        W.hydraulic.update_parameters(lay=1, thick=30, K=1e-5, Sy=1)

        #4 – Hydraulic conductivity calibration (Abhervé et al., 2023)
        K = W.calib.hydraulic_conductivity(first=1e-5, last=1e-3, method='dichotomy', obs='streams')

        #5 – Specific yield exploration (Abhervé et al., 2024)
        Sy = W.calib.specific_yield(first=0.001, last=0.1, method='exploration', obs='streamflow')

        #6 - Parametrization of the groundwater flow and transport models
        W.hydraulic.update_parameters(lay=1, thick=30, K, Sy)
        W.settings.update_particles(loc= 'seepage ', track_dir= 'backward ')
        W.model_preprocessing(build_model=True, sim_state='steady')

        #7 - Computation of groundwater flows (MODFLOW) and particle tracking (MODPATH)
        W.model_processing(gw_flow=True, particle_tracking=True)
        W.model_postprocessing(watertable_elevation=True,
                                 watertable_depth=True,
                                 seepage_areas=True,
                                 accumulated_outflow=True,
                                 particles_pathlines=True,
                                 residence_times=True,
                                 stream_intermittency=False,
                                 groundwater_flux=False,
                                 groundwater_storage=False)

        #8 - Standardized outputs and visualization
        W.visualization_2D(maps_view=True, cross_section=True)
        W.visualization_3D(interactive=True, export_vtk=True)
```

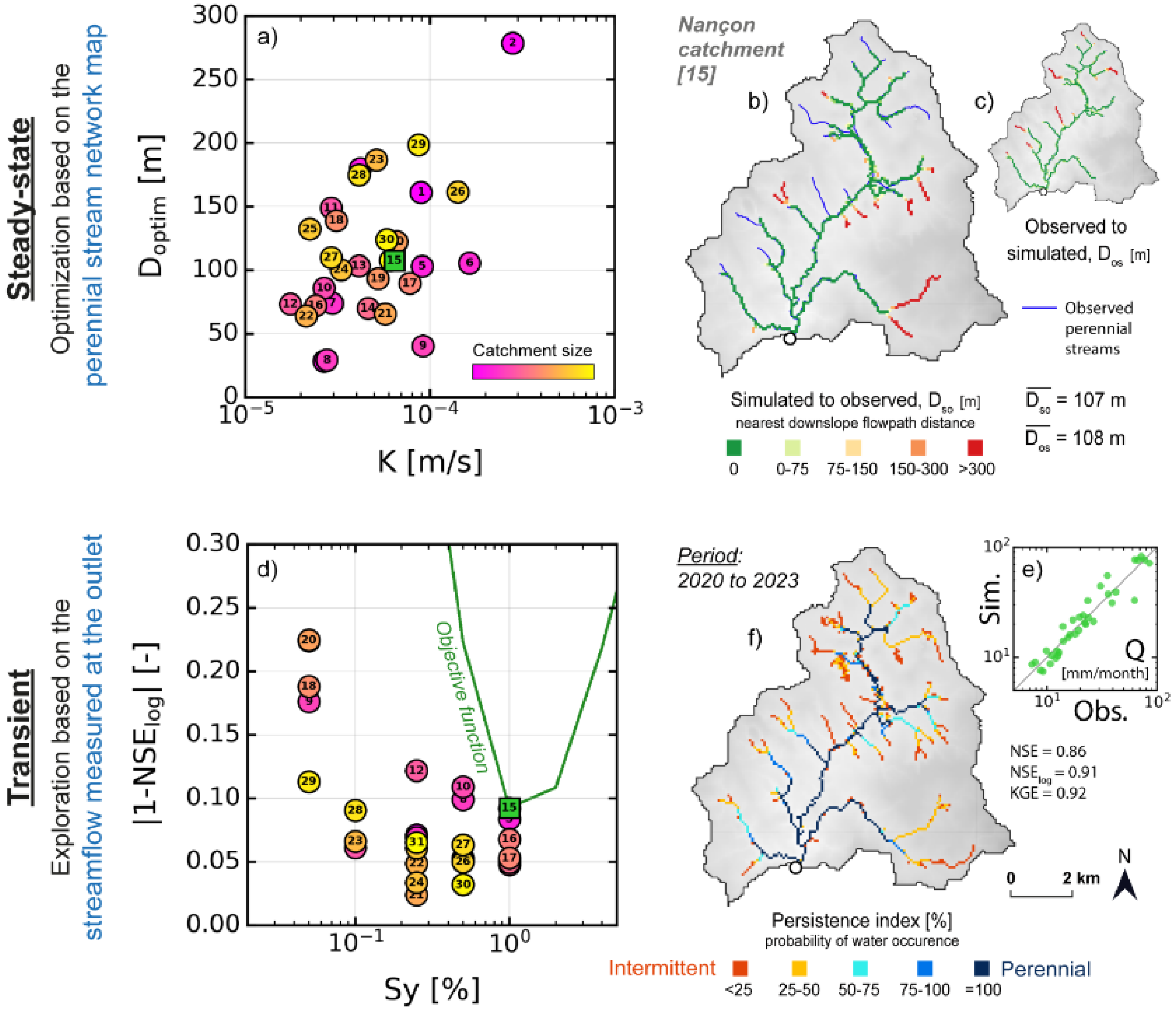
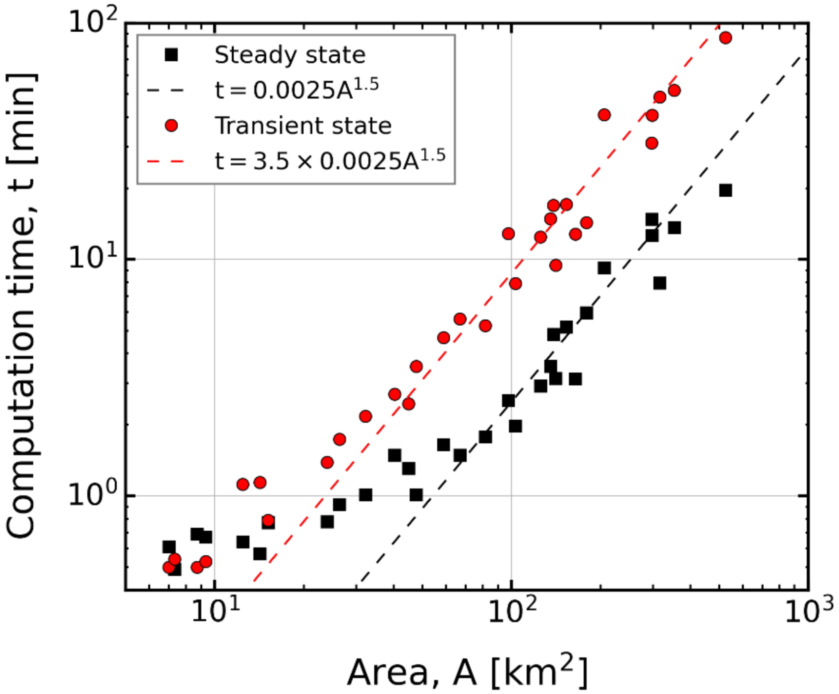


Figure 3. Workflow and results for the estimation of hydraulic conductivity *K* and specific yield *Sy* for 32 catchments (a) The best value of hydraulic conductivity *K* versus D_{optim}. The color bar represents the relative catchment size. The green square is the Nançon catchment. (b) Simulated hydrographic network represented with the distance D_{SO} from simulated seepage pixels to nearest downslope observed stream network. (c) Similar representation for D_{OS}. (d) The best value of specific yield *Sy* obtained for each catchment versus the associated $|1\text{-}NSE_{log}|$ criterion. The green line represents the objective function of the Nançon catchment. (e) Comparison of the observed and simulated specific streamflow at the catchment outlet *Q*. The black line is the 1:1 line. (f) . Representation of the persistence index of the simulated results indicating the maximum (orange lines, highly intermittent) and minimum (dark blue lines, perennial) extents of the simulated stream network.

3.2. Results and computation time

Across all the studied sites with an identical DEM resolution, the computation time depends on the catchment area (Figure 4). 300 It ranges from 0.54 min for the smallest catchment (the Guic catchment; 2 800 cells for an area of 7.3 km^2) to 86.95 min for the largest one (the Hyeres catchment; 207 152 cells for an area of 526 km^2) (Table 1). The computation time is insensitive to the area for areas smaller than 20 km^2, corresponding to cases for which the operations necessary to prepare the catchments, manage the results and draw the figures take more time than groundwater flow computation with MODFLOW. For larger areas, the groundwater flow computational time becomes dominant and scales as a function of the area to the power of 1.5 both in 305 steady-state and in transient-state, a characteristic of the numerical schemes used in MODFLOW (Harbaugh, 2005; Niswonger, 2011). For areas larger than 100 km^2, the computations in transient state take on average 3.5 more time than in steady state. This shows that most of the computing time is spent calculating flows when the number of cells exceeds around 10 000, which corresponds to a one-layer model of about 20-30 km^2 at a resolution of 75 meters.



310 **Figure 4. Evolution of the computation time in steady- and transient-state (3 years with a monthly time-step) with the catchment area for the 32 catchments of Normandy and Brittany (France). Computation times in minutes have been obtained on an Intel® Xeon® CPU E5-1620 v3 @3.50GHz (4 cores, 8 threads).**

This deployment demonstrates the ability of HydroModPy to address a wide range of hydrological conditions across varying catchment scales and topographies. Calibration is good for all catchments with low D_{optim} values and NSE$_{log}$ values greater than 315 0.75 (Figure 3), indicating a good fit between the simulated and observed stream networks and streamflow, respectively. By automating the modeling process through Python scripting, the workflow is significantly streamlined, enabling the systematic calibration of hydraulic conductivities *K* and specific yields *Sy*. Furthermore, the quality and coherence of the results can be easily assessed thanks to the generation of graphics of the calibration performances (Figure 3b), streamflow (Figure 3c) and water table fluctuations (here stream network dynamics, Figure 3d). For example, for the Nançon catchment (67 km^2, 27 004 cells, Figure 3a), the hydraulic conductivity is calibrated on the observed extension of the perennial stream network (Figure 3d) at a value of 6.40x10^{-5} m.s^{-1} , which lies in the range of the values previously determined by Dewandel et al. (2021) with other local methods. A specific yield of 1% is calibrated on the observed stream flows with a NSE$_{log}$ equal to 0.86. The overall close agreement between observations and simulations confirms the consistency of the model and its parameters in representing groundwater dynamics and its interaction with the surface in the studied catchments.

Table 1. Calibration performance criteria for the 32 calibrated models including the watershed name, watershed outlet coordinates (x-y) in the Lambert93 reference system (EPSG:2154), watershed area (km²), and number of model cells. The number of simulations (in dichotomy method for *K* and exploration method for *Sy*), computation time (minutes), calibrated values and calibration criteria (D$_{optim}$ and NSE$_{log}$) are shown for hydraulic conductivity *K* (m.s^{-1}) and specific yield *Sy* (%). Performances of calculations are given in computation times (minutes) on an Intel® Xeon® CPU E5-1620 v3 @3.50GHz (4 cores, 8 threads) processor, and 64 GB of memory for the steady-state models run for *K* and the transient-state models run for *Sy*.

					Hydraulic conductivity (K)				**Specific yield (Sy)**			
ID	Catchments	Outlet coordinates X, Y [m] RGF93 - EPSG:2154	Area (km²)	Number of cells	Number of simulations	Computation time (min)	K (m.s^{-1}) best fit	D$_{optim}$ (m)	Number of simulations	Computation time (min)	Sy (%) best fit	NSE$_{log}$ (-)
1	Langelin	180600, 6801050	7.0	2968	11	0.61	8.94×10^{-5}	161.32	9	0.50	0.25	0.93
2	Guic	213828, 6842804	7.3	2800	9	0.49	2.79×10^{-4}	278.81	9	0.54	1.0	0.91
3	Mougau-Bihan	182977, 6833659	8.7	3577	12	0.69	2.66×10^{-5}	28.35	9	0.50	0.5	0.95
4	Chèze	328853, 6784875	9.3	4757	12	0.67	4.10×10^{-5}	179.80	9	0.53	0.05	0.78
5	Troyon	159125, 6781221	12.4	6156	11	0.64	9.03×10^{-5}	103.01	9	1.12	1.0	0.92
6	Lestolet	238179, 6827960	14.2	5395	10	0.57	1.63×10^{-4}	105.50	9	1.14	1.0	0.95
7	Fremeur	255903, 6776413	15.1	5226	12	0.77	2.95×10^{-5}	74.05	9	0.79	0.25	0.93
8	Styval	186625, 6776584	23.9	9545	12	0.78	2.76×10^{-5}	29.20	9	1.39	0.5	0.90
9	Canut	327811, 6777901	26.3	9344	11	0.92	9.13×10^{-5}	40.29	9	1.74	0.05	0.82
10	Pont-Abbé	159764, 6781187	32.1	14742	12	1.01	2.66×10^{-5}	85.93	9	2.18	0.5	0.89
11	Urne	275188, 6833965	40.4	16731	12	1.49	2.90×10^{-5}	148.88	9	2.69	0.1	0.94
12	Dourduff	201590, 6855584	45.0	18688	13	1.31	1.75×10^{-5}	73.39	9	2.45	0.25	0.88
13	Coët-Organ	237193, 6774264	47.7	19096	12	1.01	4.11×10^{-5}	103.41	9	3.53	1.0	0.95
14	Yar	216004, 6858690	59.0	25669	12	1.65	4.61×10^{-5}	70.02	9	4.68	1.0	0.95
15	Nançon	389358, 6816630	67.0	27004	11	1.49	6.40×10^{-5}	107.50	9	5.60	1.0	0.91
16	Loysance	372020, 6823398	81.5	33258	13	1.78	2.40×10^{-5}	72.20	9	5.25	1.0	0.93
17	Isole	202959, 6786302	97.3	53865	11	2.54	7.77×10^{-5}	89.53	9	12.88	1.0	0.95
18	Ille	353670, 6809810	103.0	52245	12	1.98	3.10×10^{-5}	139.10	9	7.92	0.05	0.81
19	Guindy	240725, 6871783	125.0	69715	11	2.92	5.22×10^{-5}	93.42	9	12.46	0.5	0.95
20	Meu	312118, 6793547	135.0	72352	11	3.53	6.60×10^{-5}	122.35	9	14.89	0.05	0.78
21	Douffine	176707, 6818946	138.0	62139	11	4.84	5.71×10^{-5}	65.65	9	16.98	0.25	0.98

22	Penze	189998, 6854106	141.0	53380	13	3.15	2.13×10^{-5}	63.88	9	9.47	0.25	0.95
23	Rance	316728, 6807388	153.0	78144	11	5.18	5.12×10^{-5}	186.77	9	17.09	0.1	0.93
24	Jaudy	239153, 6864255	164.0	66096	12	3.13	3.29×10^{-5}	100.22	9	12.81	0.25	0.97
25	Loch	251408, 6752806	179.0	77520	13	5.96	2.24×10^{-5}	132.33	9	14.33	0.25	0.94
26	Odet	173957, 6790824	205.0	91310	10	9.23	1.41×10^{-4}	161.63	9	40.99	0.5	0.95
27	Lie	282660, 6803938	296.0	128390	12	14.78	2.90×10^{-5}	109.93	9	31.08	0.5	0.94
28	Rouvre	450637, 6862316	297.2	136640	12	12.64	4.12×10^{-5}	174.81	9	40.83	0.1	0.91
29	Evel	254000, 6772493	316.0	132854	11	7.96	8.64×10^{-5}	198.64	9	48.72	0.05	0.89
30	Leguer	227883, 6856475	353.0	172221	11	13.67	5.82×10^{-5}	123.86	9	52.07	0.5	0.97
31	Hyeres	206347, 6812298	526.0	207152	11	19.73	6.10×10^{-5}	107.75	9	86.95	0.25	0.93

4. HydroModPy key features

HydroModPy has been developed to address the need for an efficient process-based modeling approach for shallow subsurface groundwater flows, with emphasis on accessibility and ease of deployment. As an open-source toolbox, HydroModPy provides a user-friendly, flexible and adaptable platform for modeling hydrogeological systems at the catchment scale, across a wide range of spatial scales (typically 1 to 10^3 km²). It enables efficient deployment of catchment scale groundwater flow models on multiple sites using a simple for-loop in Python. This flexibility opens new perspectives to explore the role of groundwater on catchment scale flow and transport processes. Its modular and extensible architecture also allows users to expand its functionalities to meet specific research and application needs. Here, we successively discuss the software applications based on previous studies, its relevance for teaching, and the ongoing improvements and perspectives.

4.1. Applications to hydrogeological challenges

HydroModPy has been used in several catchments, mainly to constrain hydrodynamic properties and simulate groundwater contributions to streamflow (Abhervé et al., 2023, 2024; Floriancic et al., 2024; Le Mesnil et al., 2024). These early applications have paved the way for multiple research questions, including the modeling of ungauged basins. By targeting surface-subsurface interactions, HydroModPy allows surface data to be used to constrain groundwater flows, which has been a major innovation (Abhervé et al., 2023, 2024). Stream information has been used to calibrate *K/R* at steady state based on its network (Abhervé et al., 2023) following a calibration of porosity based on streamflows (Abhervé et al., 2024). These developments will leverage databases on streamflow like low-water observatories ONDE in France (Nowak & Durozoi, 2012). They will also benefit from the growing availability of high-resolution DEMs and stream network data acquired from remote sensing (Abhervé et al., 2024).

Furthermore, HydroModPy toolbox enables automatic retrieval of piezometric heads from national databases like ADES (Winckel et al., 2022) to be used in the calibration process. Additionally, HydroModPy supports the analysis of coastal aquifer response to tidal and seasonal fluctuations to calibrate the hydrodynamic properties of the aquifer, considering sea-level data as time-variant specified head boundaries (Harbaugh, 2005). This feature addresses challenges such as groundwater-induced flooding in coastal areas, particularly under sea-level rise (Gauvain, 2022; Le Mesnil et al., 2024). HydroModPy also integrates particle tracking methods to simulate residence times and calibrate aquifer storage properties (Gauvain et al., 2021). These features are essential for assessing water resource availability and designing management strategies, making HydroModPy a powerful tool for water managers and stakeholders.

HydroModPy has been initially developed for applications where streamflow is mostly supported by shallow aquifers, and assuming homogeneous effective hydraulic properties. It considers catchments where groundwater is topography controlled, focusing on near-surface flows that mostly follow the topography. The deployment remains possible in other contexts, with more complex groundwater flow patterns, e.g. regional-scale flows or inter-basin fluxes, and with geological heterogeneity. In this case, a well-defined hydrogeological catchment and/or geological model is required (De La Varga et al., 2019). By incorporating such information, HydroModPy can address of modeling complex flow patterns and accounting for geological heterogeneity, although careful calibration and validation remain essential for reliable application.

As a collaborative tool, HydroModPy is currently being used by several research teams around the world (**Erreur ! Source du renvoi introuvable.**). Several research knowledge gaps are being addressed, including the role of groundwater in coastal flood (Gauvain, 2022; Le Mesnil et al., 2023; Le Mesnil et al., 2024), the influence of topographical features on river flow (Floriancic et al., 2024) and the surface-subsurface interactions linked to geomorphology (Marti et al., 2024). These research projects follow a critical zone approach, emphasizing the need for modeling methods that are accessible to both subsurface specialists and non-specialists (Gaillardet et al., 2018). This does not mean the methods themselves are simplistic or that the processes are oversimplified. Instead, HydroModPy ensures that their implementation is transparent and user-friendly.

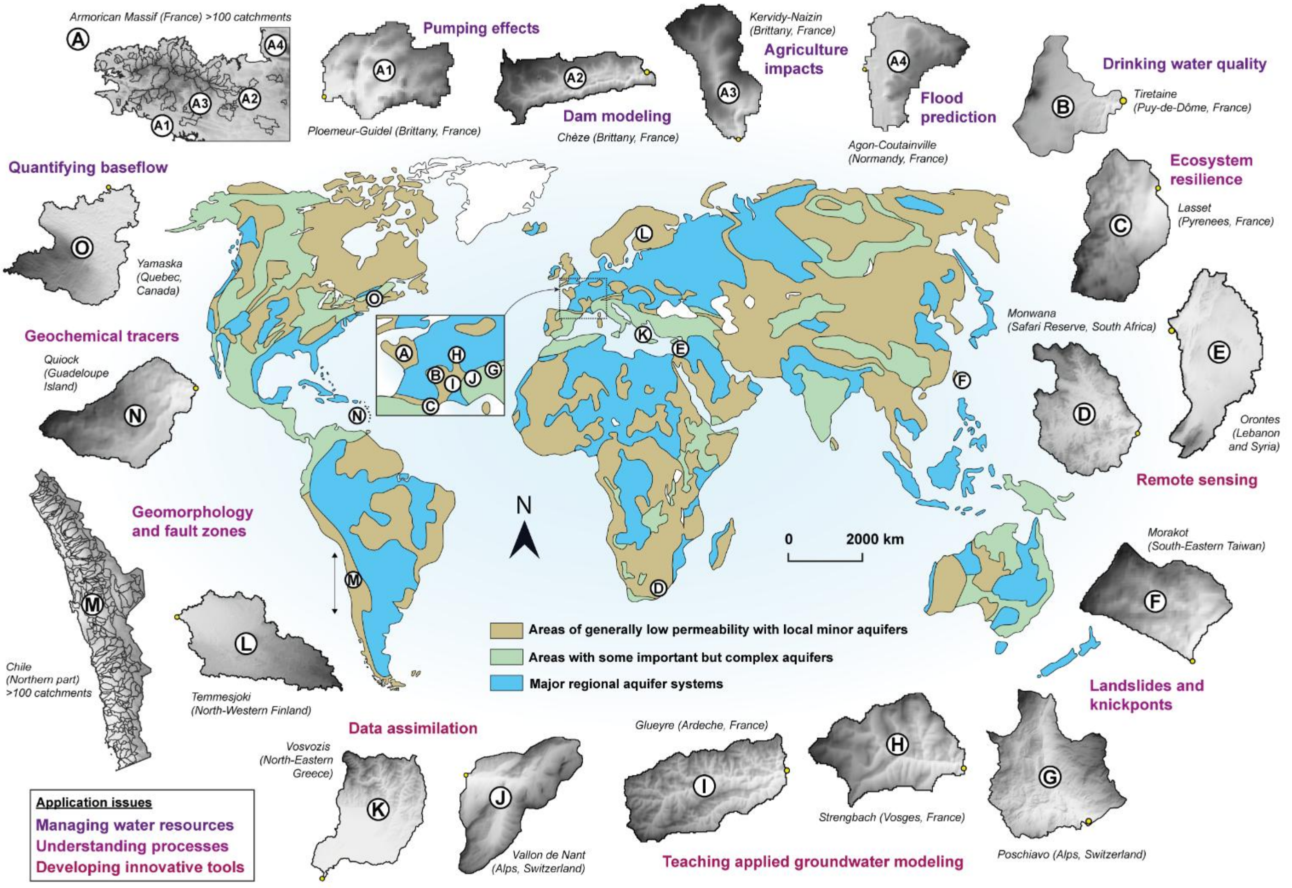


Figure 5. Worldwide application sites of HydroModPy. Simplified Global groundwater resources map modified from Taylor et al., 2013, initially obtained from Struckmeier et al., 2008. Catchments are grouped into three main application fields: water resources management, understanding processes and tool development. HydroModPy being primarily focused on subsurface/surface interactions, all catchments are in area of generally low permeability with generally shallow and local minor aquifers (brown areas on world map). Extensive use of HydroModPy has been carried out in France (A) and Chile (M) across multiple catchments.

4.2. Suitability for teaching groundwater modeling

HydroModPy offers new opportunities for training and teaching in hydrogeological modeling but also for lectures, and especially at Master level on the role of groundwater in the critical zone. HydroModPy courses have been given at Master level in several universities in France and Switzerland: University of Grenoble (led by J. Marçais), University of Rennes (led by C. Bouchez), University of Neuchâtel (led by R. Abhervé). In line with modern teaching methods, we have implemented Jupyter notebooks to run HydroModPy on example watersheds. These interactive notebooks empower students by allowing them to engage with the modeling process in a structured, step-by-step manner, running complex models with only a few lines of code.

Another advantage is that it provides a continuously evolving platform for educational purposes, as the notebooks can be easily adapted.

HydroModPy-based scripts for teaching have been used by environmental science students, proving their accessibility and ease of use even for beginners in programming and hydrogeological modeling. Students can develop their own catchment-scale MODFLOW model, which allows them to spend time directly on fundamental and applied problems, such as studying the effects of hydrogeological parametrization and boundary conditions on groundwater dynamics or evaluating the controls of groundwater on hydrographic networks. By integrating HydroModPy into the educational curriculum, students can gain hands-on experience in hydrogeology-specific modeling techniques, data analysis, and result interpretation. In addition to teaching hydrogeological modeling skills, HydroModPy-notebooks can be used as a tool to support lectures on the role of climate and geology in hydrological landscapes. Indeed, thanks to visualization tools, HydroModPy-based lectures can illustrate how the hydrological landscape changes with parameters available such as changes in recharge, aquifer thickness, hydraulic conductivity or porosity. The step-by-step notebooks developed for teaching purposes are also of interest to graduate students or faculties starting with HydroModPy or even with hydrogeological modeling in general.

4.3. Improvements and perspectives

HydroModPy provides long-term opportunities to advance hydrological modeling by integrating different critical zone processes, and new tools and datasets. Future developments aim to improve its ability to model surface and subsurface hydrodynamics with greater complexity and precision. Its flexible design allows integration with various hydrological models, such as HS1D (Marçais et al., 2017), MODFLOW 6 (Hughes et al., 2017; Langevin et al., 2017) or MT3D (Bedekar et al., 2016). These models will expand the range of options available within HydroModPy, providing users with additional tools for simulating complex hydrological processes and conducting detailed hydrological analyses. Additionally, the inclusion of more specialized aquifer models or coupled surface-subsurface models will allow for model comparison and benchmarking within HydroModPy, facilitating a more comprehensive analysis and selection of the most suitable model for their specific hydrological study or application.

A major improvement will be the use of land surface models to predict recharge input. Conceptual models, such as GR4J (Perrin et al., 2003), and distributed land surface models, such as PyHeLP (Croteau et al., 2010), are currently being integrated. Distributed land surface models will enable users to incorporate spatially variable climate data, soil types and land use data into the modeling process, providing the capacity to calculate groundwater recharge rates at spatio-temporal scales relevant from hillslope to catchment scale applications. Sequential non-iterative coupling between FloPy and other tools will also be introduced, allowing for the inclusion of intermediary processes and feedback mechanisms, notably improving the representation of groundwater-evapotranspiration interactions. To extend the use of HydroModPy in a wider range of hydrological contexts (e.g., confined aquifers, alluvial plains, or scenarios where the unsaturated zone plays a critical role), we plan to include packages/models with different levels of fidelity and coupling to represent key hydrological processes. This approach will allow users to select the most appropriate packages/models based on the dominant processes at their site of interest.

HydroModPy's calibration strategy primarily relies on systematic parameter exploration across multiple datasets and goodness-of-fit indicators. Calibration uses piezometric data evaluated by RMSE, river discharge data assessed with the Nash & Sutcliffe (1970) efficiency criteria, and stream network mapping by D_{optim} as described by Abhervé et al. (2023). Ongoing developments aim to incorporate optimized calibration techniques, including resolution methods such as Simplex (Nelder & Mead, 1965) and Metropolis-Hastings (Metropolis & Ulam, 1949). In addition to developing tailored calibration strategies to meet specific needs, we plan to integrate established open-source tools for parameter and uncertainty estimation, such as pyEMU (White et al., 2016) based on PEST (Doherty, 2015) into HydroModPy. These advances will allow multi-criteria, multi-observable, and multi-method calibrations, providing a more robust and comprehensive approach to hydrological model optimization.

HydroModPy is also undergoing developments to offer a user interface, including the creation of a graphical interface through Jupyter Notebook widgets (Kluyver et al., 2016). This work aims at providing users with a more intuitive and interactive platform for setting up, running and vizualizing models, streamlining the modeling process and improving the user experience. Tools such as Voilà (de Marchi, 2021) also offer the possibility to easily create GUIs. This will allow the tool to be rolled out to a wider audience, such as water stakeholders.

5. Overview and conclusions

The python HydroModPy toolbox allows to build and calibrate hydrogeological models at the catchment scale. It has been developed to facilitate the deployment of catchment models over large areas. The version presented here allows i) to extract a watershed from a DEM and an outlet and the available hydrological and geological data, ii) to build and run a groundwater flow model with MODFLOW, iii) to calibrate the aquifer hydrodynamic properties and iv) to visualize the results in several ways (2D or 3D). By focusing models at the catchment scale, HydroModPy facilitates a comprehensive assessment of the role of groundwater in the critical zone. Its design enables deployment ranging from localized studies to broader regional evaluations through the development of models at multiple catchment scales. We believe that HydroModPy is relevant to investigate hydrological dynamics under varying forcing at different scales, such as changing climatic conditions or land use.

We propose seven examples (available on the Git repository) to test HydroModPy across various hydrogeological conditions in different regions of France, demonstrating both its versatility and practical application:

- **Example 1:** A simplified case study centered on the catchment mainly studied in this article.
- **Example 2:** A simple case detailing the inventory of the functionalities available within the toolbox.
- **Example 3:** A demonstration of the effect of the hydraulic conductivity on the simulated hydrographic network.
- **Example 4:** A calibration of groundwater flow model in transient using streamflow and stream intermittency data.

- **Example 5:** An exploration of how the piezometric signal is affected by aquifer recharge and tidal fluctuations.
- **Example 6:** An illustration of simulated subsurface flow pathlines in 3D along with their corresponding residence times.
- **Example 7:** A 2D hillslope model that compares output results with a streamflow recession analytical solution.

HydroModPy leverages Python's capacity to automate workflows and to facilitate a shift towards deployable and replicable catchment models, which can be used for both research and educational purposes.

Code and data availability

HydroModPy is available in a public GitLab hub: https://gitlab.com/Alex-Gauvain/HydroModPy. In this paper, we present the first quite stable version 0.1 of HydroModPy. For comprehensive information on the available functionalities and options, an automatic Read the Docs documentation is available on this link: https://hydromod.readthedocs.io/

The installation of HydroModPy, as outlined in the documentation, involves checking system requirements, installing Git if necessary, cloning the repository, switching to the stable branch, navigating to the installation folder, and executing the appropriate script for setting up the environment on either Linux or Windows.

The data that support the findings of this study are available from the corresponding author upon reasonable request.

The HydroModPy Users Group, accessible via the link https://groups.google.com/g/hydromodpy, serves as a valuable platform for members of the hydrological modeling community to engage in discussions, share insights, troubleshoot issues, and collaborate on various aspects related to the utilization and development of the HydroModPy features.

Author contributions

Alexandre Gauvain and Ronan Abhervé: Conceptualization, Methodology, Software, Validation, Visualization, Writing – original draft preparation, Writing – review & editing. **Alexandre Coche:** Methodology, Software, Writing – review & editing. **Martin Le Mesnil:** Methodology, Software, Writing – review & editing. **Clément Roques:** Conceptualization, Methodology, Project administration, Supervision, Validation, Writing – review & editing. **Camille Bouchez:** Supervision, Validation, Software, Writing – review & editing. **Jean Marçais:** Software, Writing – review & editing. **Sarah Leray:** Supervision, Writing – review & editing. **Etienne Marti:** Methodology, Software. **Ronny Figueroa:** Software. **Etienne Bresciani:** Supervision, Writing – review & editing. **Camille Vautier:** Writing – review & editing. **Bastien Boivin:** Software. **June Sallou:** Software, Writing – review & editing. **Johan Bourcier:** Supervision, Writing – review & editing. **Benoit Combemale:** Supervision, Writing – review & editing. **Philip Brunner:** Project administration, Supervision. **Laurent Longuevergne:** Project administration, Supervision. **Luc Aquilina:** Funding acquisition, Project administration, Supervision, Validation, Writing – review & editing. **Jean-Raynald de Dreuzy:** Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Validation, Visualization, Writing – original draft preparation.

Acknowledgements

Alexandre Gauvain, Martin Le Mesnil, Luc Aquilina and Jean-Raynald de Dreuzy acknowledge funding from the RIVAGES Normands 2100 project. Ronan Abhervé, Alexandre Coche, Luc Aquilina and Jean-Raynald de Dreuzy acknowledge financial support from Eau du Bassin Rennais and Rennes Métropole, through the "Eaux et Territoires" ("Water and Territories") research chairs of the Foundation of the University of Rennes (Fondation Rennes 1). Clément Roques acknowledges financial support from the Rennes Métropole research chair "Ressource en Eau du Futur". Ronan Abhervé, Philip Brunner and Clément Roques acknowledge European project WATERLINE, project id CHIST-ERA-19-CES-006. The investigations also benefited from the support of the Network of hydrogeological research sites (H+) observatory and the French research observatory network of Critical Zone Observatories: Research and Applications (OZCAR-RI).

Conflict of interest

None of the authors has any competing interests.

References

Abhervé, R., Roques, C., de Dreuzy, J. R., Datry, T., Brunner, P., Longuevergne, L., & Aquilina, L. (2024). Improving calibration of groundwater flow models using headwater streamflow intermittence. *Hydrological Processes*, *38*(6), e15167. https://doi.org/10.1002/HYP.15167

Abhervé, R., Roques, C., Gauvain, A., Longuevergne, L., Louaisil, S., Aquilina, L., & De Dreuzy, J. R. (2023). Calibration of groundwater seepage against the spatial distribution of the stream network to assess catchment-scale hydraulic properties. *Hydrology and Earth System Sciences*, *27*(17), 3221–3239. https://doi.org/10.5194/hess-27-3221-2023

Ahrens, J., Geveci, B., & Law, C. (2005). ParaView: An End-User Tool for Large Data Visualization. *Visualization Handbook*.

Anderson, M. P., Woessner, W. W., & Hunt, R. J. (2015). *Applied groundwater modeling: Simulation of Flow and Advective Transport Second Edition*. http://www.sciencedirect.com:5070/book/9780120581030/applied-groundwater-modeling

Bakker, M., & Kelson, V. A. (2009). Writing analytic element programs in python. *Ground Water*, *47*(6), 828–834. https://doi.org/10.1111/j.1745-6584.2009.00583.x

Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Starn, J. J., & Fienen, M. N. (2016). Scripting MODFLOW Model Development Using Python and FloPy. *Groundwater*, *54*(5), 733–739. https://doi.org/10.1111/gwat.12413

Bedekar, V., Morway, E. D., Langevin, C. D., Tonkin, M. J., & Survey, U. S. G. (2016). MT3D-USGS version 1: A U.S. Geological Survey release of MT3DMS updated with new and expanded transport capabilities for use with MODFLOW. In *Techniques and Methods*. https://doi.org/10.3133/tm6A53

BRGM. (2006). *BSS - Ouvrages de la Banque du Sous-Sol*. https://www.geocatalogue.fr/geonetwork/srv/fre/catalog.search#/metadata/BR_BSS_BAA

Cornette, N., Roques, C., Boisson, A., Courtois, Q., Marçais, J., Launay, J., Pajot, G., Habets, F., & de Dreuzy, J. R. (2022). Hillslope-scale exploration of the relative contribution of base flow, seepage flow and overland flow to streamflow dynamics. *Journal of Hydrology*, *610*, 127992. https://doi.org/10.1016/J.JHYDROL.2022.127992

Croteau, A., Nastev, M., & Lefebvre, R. (2010). Groundwater Recharge Assessment in the Chateauguay River Watershed. *Canadian Water Resources Journal*, *35*(4), 451–468. https://doi.org/10.4296/CWRJ3504451

Crystal Ng, G. H., Wickert, A. D., Somers, L. D., Saberi, L., Cronkite-Ratcliff, C., Niswonger, R. G., & McKenzie, J. M. (2018). GSFLOW-GRASS v1.0.0: GIS-enabled hydrologic modeling of coupled groundwater-surface-water systems. *Geoscientific Model Development*, *11*(12), 4755–4777. https://doi.org/10.5194/gmd-11-4755-2018

De La Varga, M., Schaaf, A., & Wellmann, F. (2019). GemPy 1.0: Open-source stochastic geological modeling and inversion. *Geoscientific Model Development*, *12*(1), 1–32. https://doi.org/10.5194/GMD-12-1-2019

de Marchi, D. (2021). *Voilà dashboards for policy support*. Zenodo. https://doi.org/10.5281/zenodo.5082992

Dequesne, J., & Portela, S. (2024). *Panorama des services et de leur performance (rapport - données 2022) | Eaufrance*. https://www.eaufrance.fr/publications/panorama-des-services-et-de-leur-performance-rapport-donnees-2022

Dewandel, B., Boisson, A., Amraoui, N., Caballero, Y., Mougin, B., Baltassat, J. M., & Maréchal, J. C. (2021). Improving our ability to model crystalline aquifers using field data combined with a regionalized approach for estimating the hydraulic conductivity field. *Journal of Hydrology*, *601*, 126652. https://doi.org/10.1016/J.JHYDROL.2021.126652

Dewandel, B., Maréchal, J. C., Bour, O., Ladouche, B., Ahmed, S., Chandra, S., & Pauwels, H. (2012). Upscaling and regionalizing hydraulic conductivity and effective porosity at watershed scale in deeply weathered crystalline aquifers. *Journal of Hydrology*, *416–417*, 83–97. https://doi.org/10.1016/J.JHYDROL.2011.11.038

Doherty, J. (2015). Calibration and Uncertainty Analysis for Complex Environmental Models. *Groundwater*, 227.

Domenico, P. A., & Schwartz, F. W. (1998). Physical and Chemical Hydrogeology, 2nd edn John Wiley & Sons. *New York*, 528. https://www.wiley.com/en-us/Physical+and+Chemical+Hydrogeology%2C+2nd+Edition-p-9780471597629

Floriancic, M. G., Abhervé, R., Bouchez, C., Jimenez-Martinez, J., & Roques, C. (2024). Evidence of Groundwater Seepage and Mixing at the Vicinity of a Knickpoint in a Mountain Stream. *Geophysical Research Letters*, *51*(17), e2024GL111325. https://doi.org/10.1029/2024GL111325

Freeze, R. A., & Cherry, J. A. (1979). Groundwater. *Groundwater*. https://books.google.com/books/about/Groundwater.html?hl=fr&id=8P7kFowKnGUC

Gaillardet, J., Braud, I., Hankard, F., Anquetin, S., Bour, O., Dorfliger, N., Dreuzy, J. R. de, Galle, S., Galy, C., Gogo, S., Gourcy, L., Habets, F., Laggoun, F., Longuevergne, L., Borgne, T. Le, Naaim-Bouvet, F., Nord, G., Simonneaux, V., Six, D., … Zitouna, R. (2018). OZCAR: The French Network of Critical Zone Observatories. *Vadose Zone Journal*, *17*(1), 1–24. https://doi.org/10.2136/VZJ2018.04.0067

Gardner, M. A., Morton, C. G., Huntington, J. L., Niswonger, R. G., & Henson, W. R. (2018). Input data processing tools for the integrated hydrologic model GSFLOW. *Environmental Modelling & Software*, *109*, 41–53. https://doi.org/10.1016/J.ENVSOFT.2018.07.020

Gauvain, A. (2022). *Intérêts de la modélisation des résurgences d'eaux souterraines pour la caractérisation des aquifères et la définition des zones inondables : application aux bassins versants côtiers sous l'effet du changement climatique | Theses.fr*.

Gauvain, A., Leray, S., Marçais, J., Roques, C., Vautier, C., Gresselin, F., Aquilina, L., & de Dreuzy, J. R. (2021). Geomorphological Controls on Groundwater Transit Times: A Synthetic Analysis at the Hillslope Scale. *Water Resources Research*, *57*(7), e2020WR029463. https://doi.org/10.1029/2020WR029463

GDAL/OGR contributors. (2024). *GDAL/OGR Geospatial Data Abstraction software Library*. https://doi.org/10.5281/zenodo.5884351

Gillies, S., & others. (2013). *Rasterio: geospatial raster I/O for Python programmers*. https://github.com/rasterio/rasterio

Guillaumot, L., Marçais, J., Vautier, C., Guillou, A., Vergnaud, V., Bouchez, C., Dupas, R., Durand, P., de Dreuzy, J. R., & Aquilina, L. (2021). A hillslope-scale aquifer-model to determine past agricultural legacy and future nitrate

https://doi.org/10.5194/egusphere-2024-3962
Preprint. Discussion started: 8 January 2025

concentrations in rivers. *Science of The Total Environment*, *800*, 149216. https://doi.org/10.1016/J.SCITOTENV.2021.149216

Guillaumot, L., Smilovic, M., Burek, P., De Bruijn, J., Greve, P., Kahil, T., & Wada, Y. (2022). Coupling a large-scale hydrological model (CWatM v1.1) with a high-resolution groundwater flow model (MODFLOW 6) to assess the impact of irrigation at regional scale. *Geosci. Model Dev*, *15*, 7099–7120. https://doi.org/10.5194/gmd-15-7099-2022

Harbaugh, A. W. (2005). MODFLOW-2005 : the U.S. Geological Survey modular ground-water model--the ground-water flow process. *Techniques and Methods*. https://doi.org/10.3133/TM6A16

Harris, C. R., Millman, K. J., der Walt, S. J. van, Gommers, R., Virtanen, P., David Cournapeau, Wieser, E., Taylor, J., Sebastian Berg, Smith, N. J., Kern, R., Hoyer, M. P. and S., van Kerkwijk, M. H., Matthew Brett, Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Pierre Gérard-Marchant, … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hoyer, S., & Hamman, J. (2017). xarray: N-D labeled Arrays and Datasets in Python. *Journal of Open Research Software*, *5*(1), 10. https://doi.org/10.5334/JORS.148

Hughes, J. D., Langevin, C. D., Paulinski, S. R., Larsen, J. D., & Brakenhoff, D. (2023). FloPy Workflows for Creating Structured and Unstructured MODFLOW Models. *Groundwater*, 1–16. https://doi.org/10.1111/gwat.13327

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

IGN. (2011). *BD ALTI® | Descriptif de contenu*. https://geoservices.ign.fr/documentation/donnees/alti/bdalti

IGN. (2020). *Référentiel hydrographique (BD TOPAGE®) version 1 | Sandre*. https://www.sandre.eaufrance.fr/notice-doc/document-de-pr%C3%A9sentation-description-du-r%C3%A9f%C3%A9rentiel-hydrographique-bd-topage%C2%AE-version-1

Jing, M., Heße, F., Kumar, R., Wang, W., Fischer, T., Walther, M., Zink, M., Zech, A., Samaniego, L., Kolditz, O., & Attinger, S. (1989). Improved regional-scale groundwater representation by the coupling of the mesoscale Hydrologic Model (mHM v5.7) to the groundwater model OpenGeoSys (OGS). *Geosci. Model Dev*, *11*. https://doi.org/10.5194/gmd-11-1989-2018

Jordahl, K., den Bossche, J. Van, Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., maxalbert, Bilogur, A., … Leblanc, F. (2020). *geopandas/geopandas: v0.8.1*. Zenodo. https://doi.org/10.5281/zenodo.3946761

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. (2016). Jupyter Notebooks—a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016*, 87–90. https://doi.org/10.3233/978-1-61499-649-1-87

Kolbe, T., Marçais, J., Thomas, Z., Abbott, B. W., de Dreuzy, J. R., Rousseau-Gueutin, P., Aquilina, L., Labasque, T., & Pinay, G. (2016). Coupling 3D groundwater modeling with CFC-based age dating to classify local groundwater circulation in an unconfined crystalline aquifer. *Journal of Hydrology*, *543*, 31–46. https://doi.org/10.1016/J.JHYDROL.2016.05.020

Langevin, C. D., Hughes, J. D., Banta, E. R., Niswonger, R. G., Panday, S., & Provost, A. M. (2017). Documentation For The MODFLOW 6 Groundwater Flow Model. *U.S. Geological Survey, Techniques and Methods 6-A55*, 197. https://doi.org/10.3133/TM6A57

Larsen, J. D., Alzraiee, A. H., Martin, D., & Niswonger, R. G. (2022). Rapid Model Development for GSFLOW With Python and pyGSFLOW. *Frontiers in Earth Science*, *10*, 907533. https://doi.org/10.3389/FEART.2022.907533/BIBTEX

Le Mesnil, M., Gauvain, A., De Foville, S., Gresselin, F., Poirier, F., De Dreuzy, J.-R., & Aquilina, L. (2023). Rivages Normands 2100 : fédérer les acteurs académiques, institutionnels et territoriaux pour comprendre et s’adapter aux risques hydrogéologiques littoraux. *Géologues*, *219*, 90–96.

Le Mesnil, M., Gauvain, A., Gresselin, F., Aquilina, L., & de Dreuzy, J. R. (2024). Characterizing coastal aquifer heterogeneity from a single piezometer head chronicle. *Journal of Hydrology*, *642*, 131859. https://doi.org/10.1016/J.JHYDROL.2024.131859

Le Moigne, P., Besson, F., Martin, E., Boé, J., Boone, A., Decharme, B., Etchevers, P., Faroux, S., Habets, F., Lafaysse, M., Leroux, D., & Rousset-Regimbeau, F. (2020). The latest improvements with SURFEX v8.0 of the Safran-Isba-Modcou hydrometeorological model for France. *Geoscientific Model Development*, *13*(9), 3925–3946. https://doi.org/10.5194/GMD-13-3925-2020

Leray, S., Gauvain, A., & de Dreuzy, J. R. (2019). Residence time distributions in non-uniform aquifer recharge and thickness conditions – An analytical approach based on the assumption of Dupuit-Forchheimer. *Journal of Hydrology*, *574*, 110–128. https://doi.org/10.1016/J.JHYDROL.2019.04.032

Lindsay, J. B. (2016). Whitebox GAT: A case study in geomorphometric analysis. *Computers and Geosciences*, *95*, 75–84. https://doi.org/10.1016/j.cageo.2016.07.003

Los Alamos National Laboratory. (2016). *Los Alamos Grid Toolbox, LaGriT*.

Marçais, J., de Dreuzy, J. R., & Erhel, J. (2017). Dynamic coupling of subsurface and seepage flows solved within a regularized partition formulation. *Advances in Water Resources*, *109*, 94–105. https://doi.org/10.1016/j.advwatres.2017.09.008

Markstrom, S. L., Niswonger, R. G., Regan, R. S., Prudic, D. E., & Barlow, P. M. (2008). GSFLOW - Coupled Ground-Water and Surface-Water Flow Model Based on the Integration of the Precipitation-Runoff Modeling System (PRMS) and the Modular Ground-Water Flow Model (MODFLOW-2005). *Techniques and Methods*. https://doi.org/10.3133/TM6D1

Marti, E., Leray, S., & Roques, C. (2024). *Catchment landforms predict groundwater-dependent wetland sensitivity to recharge changes*. https://doi.org/10.5194/HESS-2024-381

McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

Metropolis, N., & Ulam, S. (1949). The Monte Carlo Method. *Journal of the American Statistical Association*, *44*(247), 335–341. https://doi.org/10.1080/01621459.1949.10483310

Mougin, B., Dheilly, A., Thomas, E., Blanchin, R., Courtois, N., Lachassagne, P., Wyns, R., Allier, D., & Putot, E. (2015). *Cartographie régionale au 1/250 000 de l'épaisseur des altérites et de l'horizon fissuré utile (projet SILURES Bretagne)*. https://brgm.hal.science/hal-01180206

Musy, M., Jacquenot, G., Dalmasso, G., de Bruin, R., neoglez, Müller, J., Pollack, A., Claudi, F., Badger, C., Sol, A., Zhou, Z.-Q., Sullivan, B., Lerner, B., Hrisca, D., Volpatto, D., Evan, mkerrinrapid, Schlömer, N., RichardScottOZ, … Schneider, O. (2022). *Vedo, a python module for scientific analysis and visualization of 3D objects and point clouds*. Zenodo. https://doi.org/10.5281/zenodo.7019968

Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I - A discussion of principles. *Journal of Hydrology*, *10*(3), 282–290. https://doi.org/10.1016/0022-1694(70)90255-6

Naz, B. S., Sharples, W., Ma, Y., Goergen, K., & Kollet, S. (2023). Continental-scale evaluation of a fully distributed coupled land surface and groundwater model, ParFlow-CLM (v3.6.0), over Europe. *Geosci. Model Dev*, *16*, 1617–1639. https://doi.org/10.5194/gmd-16-1617-2023

Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, *7*(4), 308–313. https://doi.org/10.1093/COMJNL/7.4.308

Niswonger, R. G. (2011). *MODFLOW-NWT, A Newton Formulation for MODFLOW-2005 Section A, Groundwater Book 6, Modeling Techniques Groundwater Resources Program*. http://www.usgs.gov/pubprod

Nowak, C., & Durozoi, B. (2012). Observatoire national des Etiages. Technical report. https://onde.eaufrance.fr/. *OFB*.

Orth, R., Staudinger, M., Seneviratne, S. I., Seibert, J., & Zappa, M. (2015). Does model performance improve with complexity? A case study with three hydrological models. *Journal of Hydrology*, *523*, 147–159. https://doi.org/10.1016/J.JHYDROL.2015.01.044

Oudin, L., Andréassian, V., Mathevet, T., Perrin, C., & Michel, C. (2006). Dynamic averaging of rainfall-runoff model simulations from complementary model parameterizations. *Water Resources Research*, *42*(7). https://doi.org/10.1029/2005WR004636

Pérez, F., Granger, B. E., & Hunter, J. D. (2011). Python: An ecosystem for scientific computing. *Computing in Science and Engineering*, *13*(2), 13–21. https://doi.org/10.1109/MCSE.2010.119

Perrin, C., Michel, C., & Andréassian, V. (2003). Improvement of a parsimonious model for streamflow simulation. *Journal of Hydrology*, *279*(1–4), 275–289. https://doi.org/10.1016/S0022-1694(03)00225-7

Pollock, D. W. (2012). User guide for MODPATH version 6 - A particle-tracking model for MODFLOW. *Techniques and Methods*. https://doi.org/10.3133/TM6A41

Pollock, D. W. (2016). User guide for MODPATH Version 7 -- A particle-tracking model for MODFLOW: U.S. Geological Survey Open-File Report. In *U.S. Geological Survey* (Issue Open File Report 2016-1086). http://pubs.er.usgs.gov/publication/ofr20161086

QGIS Development Team. (2024). *QGIS Geographic Information System*. https://www.qgis.org

Roques, C., Bour, O., Aquilina, L., & Dewandel, B. (2016). *High-yielding aquifers in crystalline basement: insights about the role of fault zones, exemplified by Armorican Massif, France*. https://doi.org/10.1007/s10040-016-1451-6

Stacke, T., & Hagemann, S. (2021). HydroPy (v1.0): A new global hydrology model written in Python. *Geoscientific Model Development*, *14*(12), 7795–7816. https://doi.org/10.5194/gmd-14-7795-2021

Staudinger, M., Stoelzle, M., Cochand, F., Seibert, J., Weiler, M., & Hunkeler, D. (2019). Your work is my boundary condition!: Challenges and approaches for a closer collaboration between hydrologists and hydrogeologists. *Journal of Hydrology*, *571*, 235–243. https://doi.org/10.1016/J.JHYDROL.2019.01.058

Struckmeier, W., Richts, A., Acworth, I., Arduino, G., Bocanegra, E., Commander, P., Cunningham, W., Döll, P., Droubi, A., da Franca, N., Gilbrich, W., Girman, J., van der Gun, J., Margat Jean, Puri, S., Rivera, A., Safar-Zitoun, M., Vrba, J., Winter, P., … Zektser, I. (2008). *Groundwater Resources of the World (1:25,000,000) (BGR & UNESCO World-wide Hydrogeological Mapping and Assessment Programme)*.

Taylor, R. G., Scanlon, B., Döll, P., Rodell, M., Van Beek, R., Wada, Y., Longuevergne, L., Leblanc, M., Famiglietti, J. S., Edmunds, M., Konikow, L., Green, T. R., Chen, J., Taniguchi, M., Bierkens, M. F. P., Macdonald, A., Fan, Y., Maxwell, R. M., Yechieli, Y., … Treidel, H. (2013). Ground water and climate change. *Nature Climate Change 2012 3:4*, *3*(4), 322–329. https://doi.org/10.1038/nclimate1744

Trefry, M. G., & Muffels, C. (2007). FEFLOW: A finite-element ground water flow and transport modeling tool. In *Ground Water* (Vol. 45, Issue 5, pp. 525–528). John Wiley & Sons, Ltd. https://doi.org/10.1111/j.1745-6584.2007.00358.x

Velásquez, N., Vélez, J. I., Álvarez-Villa, O. D., & Salamanca, S. P. (2023). Comprehensive Analysis of Hydrological Processes in a Programmable Environment: The Watershed Modeling Framework. *Hydrology 2023, Vol. 10, Page 76*, *10*(4), 76. https://doi.org/10.3390/HYDROLOGY10040076

White, J. T., Fienen, M. N., & Doherty, J. E. (2016). A python framework for environmental model uncertainty analysis. *Environmental Modelling & Software*, *85*, 217–228. https://doi.org/10.1016/J.ENVSOFT.2016.08.017

Winckel, A., Ollagnier, S., & Gabillard, S. (2022). Managing groundwater resources using a national reference database: the French ADES concept. *SN Applied Sciences*, *4*(8), 1–12. https://doi.org/10.1007/S42452-022-05082-0/FIGURES/4

Winston, R. (2009). ModelMuse—A Graphical User Interface for MODFLOW–2005 and PHAST. *U.S. Geological Survey Techniques and Methods*, *6*(A29), 1–52. https://pubs.usgs.gov/tm/tm6A29/

Zipper, S., Befus, K. M., Reinecke, R., Zamrsky, D., Gleeson, T., Ruzzante, S., Jordan, K., Compare, K., Kretschmer, D., Cuthbert, M., Castronova, A. M., Wagener, T., & Bierkens, M. F. P. (2023). GroMoPo: A Groundwater Model Portal for Findable, Accessible, Interoperable, and Reusable (FAIR) Modeling. *Groundwater*, *61*(6), 764–767. https://doi.org/10.1111/GWAT.13343