Methods Note

DISOLV: A Python package for the interpretation of borehole dilution tests

Sarah Collins[1]*, Marco Bianchi[2]

1 British Geological Survey, The Lyell Centre, Edinburgh, EH14 4AP, United Kingdom Tel.: +44 131 650 0224 Email: sarcol@bgs.ac.uk

2 British Geological Survey, Environmental Science Centre, Keyworth, NG12 5GG, United Kingdom.

*corresponding author

Conflict of interest: None.

Key words: Single borehole dilution test, transport modelling, fluid flow conductivity logging, Python

Article impact statement: Analyse, model and plot single borehole test data with the Python package DISOLV.

Abstract

Single borehole dilution tests (SBDTs) are an inexpensive but effective technique for hydrogeological characterization of hard-rock aquifers. We present a freely available, easy-to-use, open-source Python package, DISOLV, for plotting, analyzing, and modelling SBDT data. DISOLV can significantly reduce the time spent interpreting field data by helping to identify flowing fractures intersecting the borehole and estimate the corresponding flow rates. DISOLV is successfully benchmarked against two analytical solutions. We also present an example application to real data collected in a borehole in a crystalline basement aquifer in southern India.

Introduction

Hard-rock aquifers constitute the only source of water in many regions in Africa and India (Gustafson and Krásný 1994, MacDonald et al. 2012). The hydrogeological characterization of these aquifers is particularly important for locating water supplies, given that the productivity of boreholes strictly depends on intersecting water-bearing fractures. Estimating the depth and flow rate of discrete fractures can therefore provide key information with regard to the sustainability of groundwater abstractions.

Borehole flow logging techniques (e.g., Tsang et al. 1990; Brainerd and Robbins 2004; Doughty and Tsang 2005; Paillet 1998; Williams and Paillet 2002; Coleman et al. 2015) can be used to locate productive fractures and their flow properties. Among these techniques, the single borehole dilution test (SBDT) has been successfully applied to characterize the vertical distribution of aquifer hydraulic properties (e.g. Ward et al. 1998; Tsang and Doughty 2003; Williams et al. 2006; Pitrak et al. 2007; West and Odling 2007; Maurice et al. 2012; Doughty et al. 2017; Parker et al. 2018), to help find fracture depths for packer tests (e.g. West and Odling 2007, Sorensen et al. 2013), and to establish the origin of sampled groundwater for chemical analysis (e.g. Sorensen et al. 2015). The SBTD is less expensive and time consuming than other often-used traditional methods of aquifer characterization such as pumping tests, slug tests and packer tests. Moreover, it can resolve smaller flow velocities than other flow-logging techniques such as impeller and heat-pulse flowmeter logging (Mathias et al. 2007).

Practical considerations in SBTD tests including some of the potential limitations, such as the inability to resolve fractures above a highly transmissive section in a borehole and the necessity of using packers, are discussed by Tsang and Doughty (2005).

The SBDT technique involves injecting a tracer (generally a saline solution) in the open or screened section of a borehole and then measuring the dilution of the tracer with time at different depths. A variant of the SBDT is the fluid flow electrical conductivity (FFEC) logging method (Tsang et al. 1990; Doughty and Tsang 2005; Doughty et al. 2008, 2017; Tsang et al. 2016). With this technique, the water in the borehole is initially replaced with deionized water, and then the electrical conductivity profile of the fluid in the borehole is monitored while the borehole is pumped at a constant rate.

The interpretation of the experimental data collected during SBDT or FFEC tests requires the implementation of a numerical model of solute transport. In particular, the identification of flowing fractures intersected by the borehole and estimation of the associated flow rates are obtained with an inverse modelling approach aimed at fitting a 1D advection−dispersion model to the dilution versus depth data (e.g. Tsang et al. 1990, Evans et al. 1995, Mathias et al. 2007). Two codes, BORE (Hale and Tsang 1988) and its update BORE II (Doughty and Tsang 2000), were developed to perform the forward 1D solute transport required to simulate the wellbore salinity profile. However, these two codes rely on a potentially lengthy manual trial and error approach to find an optimal set of locations of inflow points and flow rates. This procedure has also the disadvantage of resulting in a subjective best match between simulated and observed concentration profiles. Alternatively, the advection−dispersion equation can be solved directly to obtain flow rates from FFEC logs, but this method cannot incorporate outflows and crossflows (Moir et al. 2014).

In this work, we present a new Python package (DISOLV) for the interpretation of both SBDT and FFEC logging data. DISOLV presents several features that make it appealing to the hydrogeological community:

- *Freely available and open source*: DISOLV is written in Python, an object-oriented, open-source programming language that has become very popular in science and engineering. DISOLV can be run on different operating systems and does not need to be compiled. Despite not having a user interface, DISOLV can be used without any knowledge of programming, as all parameters are contained in the input files. However, given that DISOLV is open source, further development and ad-hoc modification of the code can also be made.
- *Plotting capabilities*: DISOLV makes use of the *matplotlib* plotting library (Hunter, 2007) to produce publication-quality graphics.
- *Ease of use*: DISOLV input and output files are comma-separated values (CSV) files and are, therefore, easy to produce and edit in Python or in a spreadsheet software such as Microsoft Excel.

- *Automatic estimation of fracture locations and flows*: DISOLV can be used in combination with any package from Python's extensive library (e.g. Oliphant 2006). One of the main advantages of DISOLV is that it can be used to perform automatic calibration of the transport model, and therefore the aquifer parameterization, with the optimization algorithms included in the scientific computing package SciPy (Jones et al. 2004). Compared with the trial and error approach, which is also an option with DISOLV, automatic calibration eliminates user intervention, reduces the time required for finding best fit parameter values, and allows the uncertainty of the calibration process to be quantified (Poeter and Hill, 1995).

There are four features coded into BORE II that are not included in DISOLV: (1) time-varying inflow/outflow rates, (2) time-varying feed-point concentrations, (3) a delay before concentration response occurs and (4) long logging times (i.e. time taken to move the probe up and down the borehole) relative to the time between loggings. However, DISOLV's simple Python code, provides the required flexibility to add in extra features such as these.

This paper provides an overview of DISOLV's capabilities and structure. Benchmark tests and a real-world example of its application are also presented.

Mathematical background

For the interpretation of SBDT or FFEC data, DISOLV solves the 1D advection–dispersion equation (Tsang et al. 1990, Mathias et al. 2007):

$$\frac{\partial}{\partial z}\left(D(z)\frac{\partial C}{\partial z}\right) - \frac{\partial}{\partial z}\,(q(z)\;C) + \frac{Q_s C_s}{\Delta z_s A}\; = \frac{\partial C}{\partial t} \tag{1}$$

where z [L] is depth, A [L^2] is the cross-sectional area of the borehole, C [ML^{-3}] is the solute concentration, $q(z)$ [LT^{-1}] is the longitudinal flow in the borehole, Q_s are the inflows or the outflows [$L^3\ T^{-1}$] associated with the fractures intersecting the borehole and Δz_s is the height of the fracture. The dispersion coefficient D(z) [$L^2\ T^{-1}$] can be expressed in terms of dispersivity α [L], flow velocity longitudinal to the borehole and the diffusion coefficient D_d [$L^2\ T^{-1}$] : $D(z) = \alpha\ q(z) + D_d$. For inflowing fractures, the terms Q_s are positive and C_s is the solute concentration of the formation water, whereas, for outflowing fractures, the Q_s terms are taken as negative and C_s is equal to the solute concentration in the borehole. The height of the fracture, Δz_s, is set to the chosen spatial discretization of the grid in DISOLV, but larger fractures can be simulated with multiple inflows/outflows. A typical grid discretization is ~0.1–0.2 m, given that the dispersivity is typically ~0.1–0.5 m and the Peclet number ($\Delta z/\alpha$) must be lower than 2. Equation (1) assumes steady-state flow and that there are no lateral changes in concentration within the borehole. To solve (1), we consider the following boundary conditions:

$$\left.\frac{\partial C}{\partial z}\right|_{z=0} = \left.\frac{\partial C}{\partial z}\right|_{z=z_{max}} = 0 \tag{2}$$

where z = 0 and z_{max} are the elevations of the water table and the base of the borehole, respectively. For a SBDT with tracer injection, the initial solute concentration $C(z,0)$ is the first measured concentration profile. Generally SBDT tests are conducted using a saline solution as the tracer and by monitoring variations in electrical conductivity of the fluid in the borehole. In DISOLV these data are converted into concentration values using the empirical formula (Doughty and Tsang 2005):

$$C = \frac{1870 - \sqrt{1870^2 - 160\ \mathrm{FEC}_{20}}}{80} \qquad (3)$$

where FEC_{20} is the electrical conductivity at 20˚C in μS cm^{-1} and the resulting concentration is expessed in kg m^{-3}. The relationship between electrical conductivity and temperature is generally non-linear, but can be approximated as linear for 0–30˚C with the following equation (Sorenson and Glass, 1987):

$$\mathrm{FEC}_{T2} = \frac{\mathrm{FEC}_{T1}}{\left(1+c\ (T1-T2°C)\right)} \qquad (4)$$

where FEC_{T1} and FEC_{T2} are electrical conductivity at temperatures *T1* and *T2* and c [˚C^{-1}] is a compensation factor. Hayashi (2003) notes that c varies depending on *T2* – despite the near linear relationship – and that for any given *T2* there are a range of values of c in the literature. For example, values of c are reported to vary between 0.0191 and 0.025 for a *T2* of 25˚C

(Hayashi 2003). DISOLV uses a value of 0.024˚C^{-1} – which is used by Doughty and Tsang (2005) for *T2* = 20˚C – for both 20˚C and 23˚C. Equation (3) breaks down above 10 000 μS cm^{-1}, at which point DISOLV uses the equation of Vinogradov et al. (2010):

$$C = 5.9738\ \times 10^{-7}\ \mathrm{FEC}_{23}^{6}\ - 3.5136\ \times\ 10^{-5}\ \mathrm{FEC}_{23}^{5} + 7.823\ \times\ 10^{-4}\ \mathrm{FEC}_{23}^{4} - 8.0334\ \times\ 10^{-3}\ \mathrm{FEC}_{23}^{3} + 4.0791\ \times\ 10^{-2}\ \mathrm{FEC}_{23}^{2} + 3.4996\ \times\ 10^{-2}\ \mathrm{FEC}_{23} + 3.6104\ \times 10^{-2} \qquad (5)$$

where FEC$_{23}$ is the electrical conductivity at 23˚C and *C* is solute concentration in mol l^{-1}. The solute concentration is converted to kg m^{-3} by multiplying by the molar mass of NaCl (58.44 g mol^{-1}).

DISOLV solves Equation (1) using a finite difference approach. The central-in-space weighting scheme is used to derive a set of ordinary differential equations with respect to time. These are solved at exact times with the numerical integrator *odeint,* available from the SciPy package, using an algorithm adapted from the FORTRAN library *odepack* (Hindmarsh 1983). Time stepping and the integration method are both adaptive, with o*deint* automatically switching between the Adams method for non-stiff problems and the backward differentiation formula for stiff problems. The solution is a 2D array containing calculated concentration values at each requested output time at each node of the 1D grid given a set of input parameter values, including the depth *z* of inflowing and outflowing fractures, their flow rates (Q_s in Equation (1)) and the dispersivity α. When DISOLV is used in inverse modelling mode, these parameters are automatically adjusted until a best fit is found between simulated and measured concentration data.

Package structure and use

DISOLV has a simple file structure (Figure 1) such that it can be run with just two lines of Python code, as all input parameters are contained in the input files. The package has two modes, forward and inverse modelling (i.e. automatic calibration).

DISOLV has four input files:

- *in.csv* contains the input parameters as well as user-defined constraints on dispersivity and total flow for the automatic calibration. An example of an *in.csv* file is shown in Figure 2.
- *flows.csv* contains the fracture depths and flow rates. In inverse mode, this file also contains user-defined constraints on fracture depth. An example is shown in Figure 3.
- *initialcondition.csv* contains the first set of depth versus concentration data (Figure 3). Output times in *in.csv* are defined relative to the time at which the initial condition was measured, and therefore it is irrelevant whether the initial condition was measured directly after the tracer injection or sometime later.
- *measuredprofiles.csv* contains depth versus concentration data for each measured profile (Figure 3). This input file is only required when DISOLV is used in the inverse modelling mode.

In DISOLV, the data in *initialcondition.csv* and *measuredprofiles.csv* files are assigned to the blocks of the 1D grid using linear interpolation. Any units can be used for values of the parameters in the input files, as long as they are consistent, the only exception being water temperature, which is in degrees Celsius.

Forward modelling

The package for simulating concentration profiles in the borehole is imported and ran in a Python script as follows:

```
Import disolv
disolv.run(’Input’,’Output’,calibrate=False,convertFEC=False)
```

The first two arguments are the input and output directories. The third argument is a switch to turn automatic calibration on (‘True’) or off (‘False’) and the final argument indicates whether the initial condition has been given in fluid electrical conductivity (μS cm^{-1}) and must be converted to concentration (in kg m^{-3}) (‘True’) or whether it has been given as a concentration (‘False’). Two files will be generated in the output directory: *profiles.csv*, comprising the simulated depth versus concentration data at different times, and *profiles.png*, a plot of the simulated and measured (if given) depth versus concentration data.

The procedure for modelling FFEC logging is the very similar. The initial condition *C*(z,0) is the concentration of the solution with which the borehole water has been replaced, and the output times (provided in the *in.csv* file) are relative to the onset of pumping rather than the time at which the first profile was measured. Borehole pumping is simulated as an additional outflow point q_s by adding an extra line to *flows.csv* specifying the depth of the pump below ground and the pumping rate used.

Inverse modelling

In inverse mode, DISOLV adjusts the depth and flow rate of each fracture as well as the dispersivity. The range of variability of these parameters can be controlled by setting constraints on the minimum and maximum depth of each fracture, the total flow rate in the borehole and the dispersivity. Solving the inverse problem presents some challenges (see comprehensive reviews on the topic such as Carrera et al. 2004; Doherty 2015). For interpreting SBDT data, setting appropriate constraints are important in improving the stability of the inversion and the robustness of the results. In this sense, visual inspection can help identify inflow and outflow points (e.g. Maurice et al. 2011) as well as the application of the mass integral method (Doughty and Tsang, 2005). Constraints on fracture depth are contained in the input file *flows.csv* and constraints on dispersivity (`AlphaMin`, `AlphaMax`) and total flow (`FlowMin`, `FlowMax`) are applied in the file *in.csv* as follows:

```
# Bounds for automatic calibration of alpha and total inflow/outflow
# (AlphaMin AlphaMax FlowMin FlowMax)
0.1 0.3 0.001 1
```

In inverse mode, the values given for dispersivity in *in.csv* and fracture depth and flow rate in *flows.csv* are taken as the initial guesses of these parameters in the calibration.

In the case of FFEC logging data, the total flow rate in the borehole is equal to the constant pumping rate for the test. Therefore, the minimum and maximum total flow need to be set equal to this value and the model will only adjust the rates of individual inflowing fractures.

The package can be run in calibration mode as follows:

```
disolv.run(’Input’,’Output’,calibrate=True,convertFEC=False,method
=’SLSQP’)
```

Calibration is carried out by minimizing the root mean squared error (RMSE) between simulated and measured concentration profiles. DISOLV uses the *scipy.optimize* package, which includes several optimization algorithms including gradient-based methods, direct search, and heuristic approaches. The default optimization method is the sequential least squares programming algorithm (SLSQP) (Kraft, 1988), but a different optimization approach can be chosen by changing the final argument of the above command. Other parameters of *scipy.optimize.minimize* can also be passed though the `disolv.run` command, such as the method by which the gradient vector is calculated (i.e. 'jac'). A full list of available optimization methods is provided in the SciPy manual (Jones et al. 2001).

Model benchmarking

DISOLV has been benchmarked against two analytical solutions: the Drost et al (1968) equation for simulating dilution at a point due to horizontal flow across a borehole, and the Ogata and Banks (1961) analytical solution of Equation (1) assuming a continuous source (see Equation 1) at z=0. The Drost et al (1968) solution for the change in solute concentration at a point in the borehole due only to cross-flow across the borehole (i.e. no vertical flow up or down the borehole) is as follows:

$$C(t) \;=\; C_0 - (C_0 - \; C(0))\, \mathrm{e}^{-\frac{2\, t\, \alpha_h\, v_h}{\pi\, r}} \qquad (6)$$

where *C*(*t*) and *C*(0) are solute concentration in the borehole at times *t* and *t* = 0, respectively, v_h is the far-field fracture flow, α_h is the aquifer-to-wellbore convergence factor and *r* is the radius of the borehole. The Ogata−Banks analytical equation for 1D advection−dispersion – in this case, longitudinal to the borehole – is as follows:

$$C(z,t) = \frac{C(z,0)}{2}\;(\,\mathrm{erfc}\left(\frac{z-v_v t}{2\sqrt{Dt}}\right) + e^{\frac{v_v z}{D}}\,\mathrm{erfc}\left(\frac{z+v_v t}{2\sqrt{Dt}}\right)) \qquad (7)$$

where $C(z,t)$ and $C(z,0)$ are solute concentration in the borehole at times t and $t = 0$ and at position z down the borehole, v_v is the component of the velocity longitudinal to the borehole and erfc is the complementary error function. As can be seen from Figure 4, DISOLV is able to produce accurate estimations of both analytical solutions.

Table 1 Parameters used for model benchmarking (see Figure 4)

Parameter	**Model benchmark**	
	Drost et al (1968)	**Ogata-Banks (1961)**
Initial concentration (kg m^{-3})	1000	0
Concentration of inflowing water (kg m^{-3})	10	1
Dispersivity (m)		1
Convergence factor (–)	2.85	
Far-field fracture flow (m d^{-1})	0.4	–
Longitudinal flow (m d^{-1})	–	1
Borehole radius (m)	0.038	0.564
Time (d)	–	100

Example of application of DISOLV to real-world data

A SBDT was carried out in a gneiss fractured-bedrock aquifer near Gundlupet, Karnataka, southern India. The site is within the Berambadi catchment, which has been used as a hydrological observatory since 2009 (Sekhar et al. 2016; Robert et al. 2017). The fractured aquifer is heavily exploited for irrigation, as surface waters are ephemeral (Buvaneshwari et al. 2017, Robert et al. 2017).

The test was carried out under ambient flow conditions with a methodology similar to those used by Ward et al (1998) and Mathias et al (2007). First, an Aqua Troll 200 (In-Situ Europe Ltd., Redditch, UK) temperature, pressure and salinity probe was lowered down the borehole to obtain a background salinity profile. A hosepipe was then inserted into the borehole and a salt solution (1 kg l^{-1}) injected into the saturated length of the hose pipe. The hosepipe was then removed and electrical conductivity profiles obtained periodically with the probe. The initial analysis of the field data identified the location of two flowing fractures: for example, the comparison of the measured concentration profiles at different times (Figure 5) indicates tracer dilution at around 49 m below ground, suggesting a major inflowing fracture, whereas the initial peak of concentration disappeared after 1.45 hours at the bottom of the borehole, suggesting a major outflowing fracture. DISOLV was run in inverse mode (using SLSQP) to estimate the flow rates of these two fractures and dispersivity. The match between the observed and simulated data was reasonably good (RMSE 0.53 kg m^{-3}). However, another run of DISOLV in inverse mode was able find a better fit between simulated and measured profiles. In particular, DISOLV was able to identify and parameterize an inflowing fracture at

~ 75 m and an outflow fracture at ~ 90 m below ground (Figure 5). The addition of these two fractures in the numerical model resulted in an improvement in RMSE of about 14% (0.46 kg m^{-3}).

Summary and final remarks

In fractured-bedrock aquifers, SBDTs are an inexpensive and effective method of aquifer characterization. We have developed a Python package, DISOLV, that can support the interpretation of SBDT field data by allowing rapid identification of flowing fractures and estimation of the corresponding flow rates. We believe DISOLV is an effective tool for gaining quantitative information that can contribute towards locating water supplies and estimating the sustainability of abstractions in fractured aquifers.

Acknowledgements

The authors are grateful to Somsubhra Chattopadhyay (ATREE, Bangalore), Siva Naga Venkat Nara (Indian Institute of Sciences, Bangalore) and Sian Loveless (British Geological Survey) for collecting the data used in the example. The research underlying this paper was carried out under the UPSCAPE project of the Newton-Bhabha programme “Sustaining Water Resources for Food, Energy and Ecosystem Services”, funded by the UK Natural Environment Research Council (NERC-UKRI), grant number NE/N016270/1. The authors publish with the permission of the Executive Director of the British Geological Survey (BGS-UKRI).

DISOLV is free and open-source software under the GNU GPLv3 licence. The code, example input files and a brief user guide describing file structures are available at https://github.com/BritishGeologicalSurvey/disolv. DISOLV is listed in the Python packaging index PyPI and can thus be installed with the following command: `pip install disolv`.

References

Brainerd, R.J., and G.A. Robbins. 2004. A tracer dilution method for fracture characterization in bedrock wells. *Groundwater* 42, no 5: 774−780.

Buvaneshwari, S., J. Riotte, M. Sekhar, M.M. Kumar, A.K. Sharma, J.L. Duprey, S. Audry, P. Giriraja, Y. Praveenkumarreddy, and H. Moger. 2017. Groundwater resource vulnerability and spatial variability of nitrate contamination: insights from high density tubewell monitoring in a hard rock aquifer. *Science of the Total Environment* 579: 838−847.

Carrera, J., A. Alcolea, A. Medina, J. Hidalgo, and L.J. Slooten. 2005. Inverse problem in hydrogeology. *Hydrogeology Journal* 13 (1): 206–222. https://doi.org/10.1007/s10040-004-0404-7.

Coleman, T.I., B.L. Parker, C.H. Maldaner, and M.J. Mondanos. 2015. Groundwater flow characterization in a fractured bedrock aquifer using active DTS tests in sealed boreholes. *Journal of Hydrology* 528: 449−462.

Doherty, J. 2015. Calibration and Uncertainty Analysis for Complex Environmental Models. Watermark Numerical Computing, Brisbane, Australia.

Doughty, C., and C.-F. Tsang. 2000. BORE-II-A code to compute dynamic wellbore electrical conductivity logs with multiple inflow/outflow points including the effects of horizontal flow across the well. Rep. LBL-46833, Lawrence Berkeley National Laboratory, Berkeley, CA.

Doughty, C., and C.-F. Tsang. 2005. Signatures in flowing fluid electric conductivity logs. *Journal of Hydrology* 310, nos. 1−4: 157−180.

Doughty, C., C.F. Tsang, K. Hatanaka, S. Yabuuchi, and H. Kurikami. 2008. Application of direct-fitting, mass integral, and multirate methods to analysis of flowing fluid electric conductivity logs from Horonobe, Japan. *Water Resources Research* 44: W08403.

Doughty, C., C.-F. Tsang, J.-E. Rosberg, C. Juhlin, P.F. Dobson, and J.T. Birkholzer. 2017. Flowing fluid electrical conductivity logging of a deep borehole during and following drilling: estimation of transmissivity, water salinity and hydraulic head of conductive zones. *Hydrogeology Journal* 25, no. 2: 501−517.

Drost, W., D. Klotz, A. Koch, H. Moser, F. Neumaier, and W. Rauert, 1968. Point dilution methods of investigating ground water flow by means of radioisotopes. *Water Resources Research* 4, no. 1: 125−146.

Evans, D.G. 1995. Inverting fluid conductivity logs for fracture inflow parameters. *Water Resources Research* 31, no. 12: 2905−2915.

Gustafson, G., and J. Krásný, 1994. Crystalline rock aquifers: their occurrence, use and importance. *Applied Hydrogeology* 2, no. 2: 64−75.

Hale, F., and C.-F. Tsang, 1988. A code to compute borehole fluid conductivity profiles with multiple feed points. Lawrence Berkeley National Laboratory, Berkeley, CA.

Hayashi, M. 2004. Temperature−electrical conductivity relation of water for environmental monitoring and geophysical data inversion. *Environmental Monitoring and Assessment* 96, no. 1−3: 119−128.

Hindmarsh, A.C. 1983. ODEPACK, a systematized collection of ODE solvers, in *Scientific Computing*, R. S. Stepleman et al. (eds.), North-Holland, Amsterdam, vol. 1 of IMACS Transactions on Scientific Computation, pp. 55−64.

Hunter, J. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, no 3: 90−95.

Jones, E., T. Oliphant, and P. Peterson. 2001. SciPy: Open Source Scientific Tools for Python. Available at: https://www.scipy.org [accessed online 19 August 2019].

Kraft, D. 1988. A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt* 88-28.

MacDonald, A.M., H.C. Bonsor, B.É.Ó. Dochartaigh, and R.G. Taylor. 2012. Quantitative maps of groundwater resources in Africa. *Environmental Research Letters* 7, no. 2: 024009.

Mathias, S., A. Butler, D. Peach, and A. Williams. 2007. Recovering tracer test input functions from fluid electrical conductivity logging in fractured porous rocks. *Water Resources Research* 43, no. 7.

Maurice, L., J.A. Barker, T.C. Atkinson, A.T. Williams, and P.L. Smart. A tracer methodology for identifying ambient flows in boreholes. *Groundwater* 49, no. 2: 227−238.

Maurice, L., T. Atkinson, J. Barker, A. Williams, and A. Gallagher. 2012. The nature and distribution of flowing features in a weakly karstified porous limestone aquifer. *Journal of Hydrology* 438: 3−15.

Moir, R., S. Alison, H. Parker, and R.T. Brown. A simple inverse method for the interpretation of pumped flowing fluid electrical conductivity logs. *Water Resources Research* 50, no. 8: 6466−6478.

Oliphant, T.E. 2006. *Guide to NumPy*. USA: Trelgol Publishing.

Ogata, A., and R. Banks, 1961. A solution of the differential equation of longitudinal dispersion in porous media: fluid movement in earth materials, 411. US Government Printing Office.

Parker, A.H., L.J. West, and N.E. Odling. 2018. Well flow and dilution measurements for characterization of vertical hydraulic conductivity structure of a carbonate aquifer. *Quarterly Journal of Engineering Geology and Hydrogeology* 52, no. 1: 74−82.

Paillet, F.L., 1998. Flow modeling and permeability estimation using borehole flow logs in heterogeneous fractured formations. *Water Resources Research* 34, no. 5: 997−1010.

Pitrak, M., S. Mares, and M. Kobr. 2007. A simple borehole dilution technique in measuring horizontal ground water flow. *Groundwater* 45, no. 1: 89−92.

Poeter, E. & Hill, M. C. Inverse Models: A Necessary Next Step in Ground-Water Modeling. *Groundwater* 35, no. 2, 250–260 (1997).

Robert, M., A. Thomas, M. Sekhar, S. Badiger, L. Ruiz, M. Willaume, D. Leenhardt, and J.-E. Bergez. 2017. Farm typology in the Berambadi Watershed (India): Farming systems are determined by farm size and access to groundwater. *Water* 9, no. 1: 51.

Sekhar, M. 2016. Influences of climate and agriculture on water and biogeochemical cycles: Kabini critical zone observatory. *Proceedings of the Indian National Science Academy* 82, no. 3: 833−846.

Sorenson, J.A., and G.E. Glass. 1987. Ion and temperature dependence of electrical conductance for natural waters. *Analytical Chemistry* 59, no. 13: 1594−1597.

Sorensen, J.P., L. Maurice, F.K. Edwards, D.J. Lapworth, D.S. Read, D. Allen, A.S. Butcher, L.K. Newbold, B.R. Townsend, and P.J. Williams. 2013. Using boreholes as windows into groundwater ecosystems. *PLoS One* 8, no. 7: e70264.

Sorensen, J.P., A.S. Butcher, M.E. Stuart, and B.R. Townsend. 2015. Nitrate fluctuations at the water table: implications for recharge processes and solute transport in the Chalk aquifer. *Hydrological Processes* 29, no. 15: 3355−3367.

Tsang, C.F., and C. Doughty. 2003. Multirate flowing fluid electric conductivity logging method. *Water Resources Research* 39, no. 12.

Tsang, C.F., P. Hufschmied, and F.V. Hale, 1990. Determination of fracture inflow parameters with a borehole fluid conductivity logging method. *Water Resources Research* 26, no. 4: 561−578.

Tsang, C.-F., J.-E. Rosberg, P. Sharma, T. Berthet, C. Juhlin, and A. Niemi. 2016. Hydrologic testing during drilling: application of the flowing fluid electrical conductivity (FFEC) logging method to drilling of a deep borehole. *Hydrogeology Journal* 24, no. 6: 1333−1341.

Vinogradov, J., M.Z. Jaafar, M.D. Jackson. 2010. Measurement of streaming potential coupling coefficient in sandstones saturated with natural and artificial brines at high salinity. *Journal of Geophysical Research: Solid Earth* 115: B12204.

Ward, R., A. Williams, and J. Barker, 1998. Groundwater tracer tests: a review and guidelines for their use in British aquifers. Environment Agency.

West, L.J., and N.E. Odling. 2007. Characterization of a multilayer aquifer using open well dilution tests. *Groundwater* 45, no. 1: 74−84.

Williams, A., J. Bloomfield, K. Griffiths, and A. Butler. 2006. Characterising the vertical variations in hydraulic conductivity within the Chalk aquifer. *Journal of Hydrology* 330, nos. 1–2: 53–62.

Williams, J., and F.L. Paillet. 2002. Using flowmeter pulse tests to define hydraulic connections in the subsurface: a fractured shale example. *Journal of Hydrology* 265, nos. 1–4: 100–117.

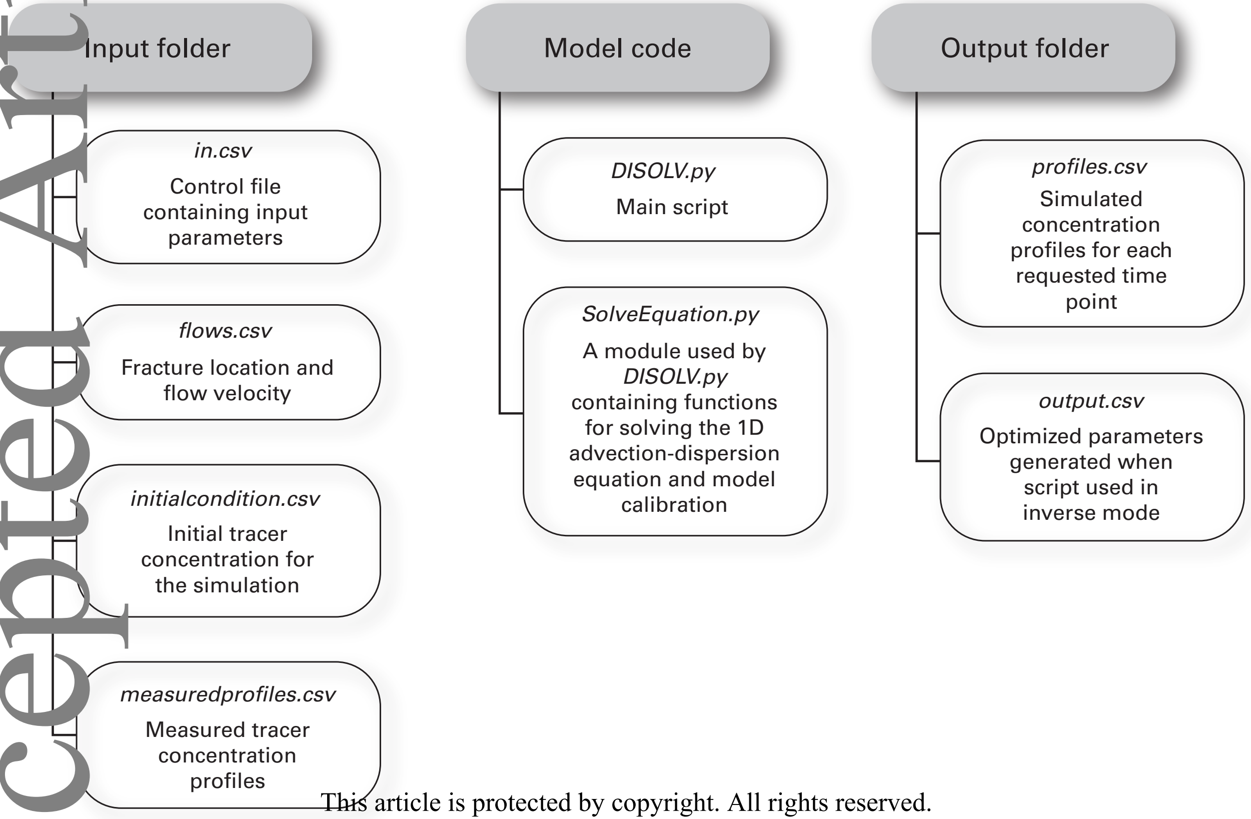
Figure 1 DISOLV file structure.

Figure 2 Format of input file *in.csv*.

Figure 3 Formats of input files *flows.csv*, *initialcondition.csv* and *measuredprofiles.csv*.

Figure 4 Comparison of DISOLV numerical model against the (**a**) Drost et al (1968) and the (**b**) Ogata-Banks (1961) analytical solutions. The parameters for the benchmark simulations are provided in Table 1.

Figure 5 (**b**) Observed (Obs.) and simulated (Sim.) salt dilution profiles in the borehole. (**a**) Expanded view of Box a in part (**b**). (**c**) Calibrated flows used to produce simulated dilution profiles.


Input folder
in.csv
Control file containing input parameters
flows.csv
Fracture location and flow velocity
initialcondition.csv
Initial tracer concentration for the simulation
measuredprofiles.csv
Measured tracer concentration profiles
Model code
DISOLV.py
Main script
SolveEquation.py
A module used by *DISOLV.py* containing functions for solving the 1D advection-dispersion equation and model calibration
Output folder
profiles.csv
Simulated concentration profiles for each requested time point
output.csv
Optimized parameters generated when script used in inverse mode

```
#Depth to groundwater level [L]
<float>
#Borehole depth [L]
<float>
#Spatial discretization of grid [L]
<float>
#Constant borehole cross-sectional area [L^2]
<float>
#Dispersivity [L^-1]
<float>
#Diffusion coefficient [L^2/T]
<float>
#Tracer concentration of water inflow [M/L^3]
<float>
#Water temperature [Celsius] (only read if above is FEC)
<float>
#Bounds for automatic calibration of alpha and total
#inflow/outflow (AlphaMin AlphaMax FlowMin FlowMax)
<float>,<float>,<float>,<float>
#Times at which to output profiles [T]
<float>,...,<float>
```

a) flows.csv

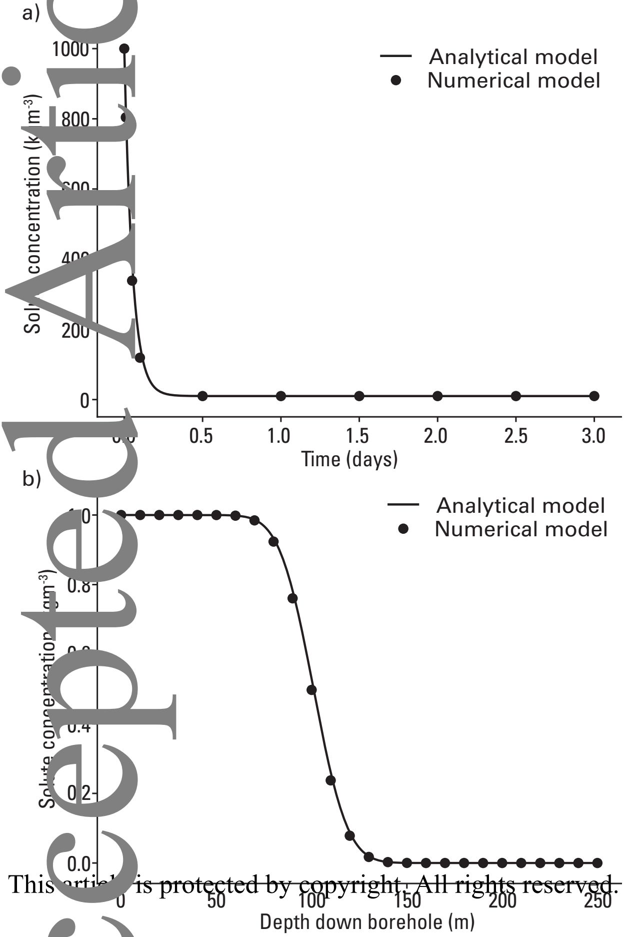
```
#Depth[L],Flow[L^3T^-1],Upper z bound[L],Lower z bound[L]
<float>,<float>,<float>,<float> #Q1
...,...,...,...
<float>,<float>,<float>,<float> #Qn
```

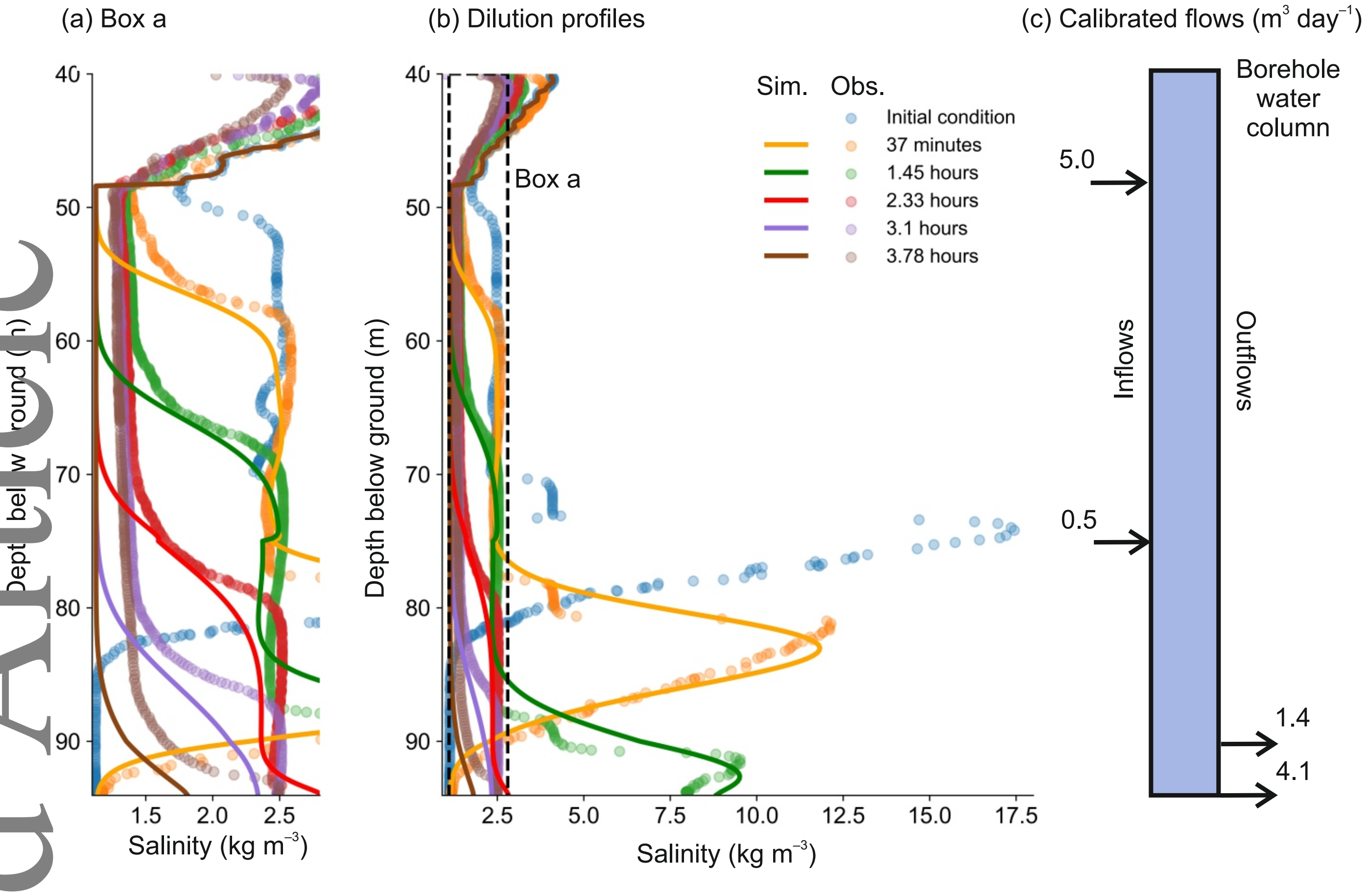
b) initialcondition.csv

```
#Depth[L],Concentration[M L^-3 or microS/cm]
<float>,<float>
...,...
<float>,<float>
```

c) measuredprofiles.csv

```
#Depth1[L],C1[M L^-3 or microS/cm],..., DepthM[L],
#CM[M L^-3 or microS/cm]
<float>,<float>, ...,... <float>,<float>
...,...,...,...,...,...
<float>,<float>, ...,... <float>,<float>
```


a)
Solute concentration (kg m-3)
1000
800
600
400
200
0
Analytical model
Numerical model
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Time (days)
b)
Solute concentration (g m-3)
1.0
0.8
0.6
0.4
0.2
0.0
Analytical model
Numerical model
0
50
100
150
200
250
Depth down borehole (m)


(a) Box a
40
50
60
70
80
90
Depth below ground (m)
1.5
2.0
2.5
Salinity (kg m^{-3})
(b) Dilution profiles
Box a
Depth below ground (m)
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Salinity (kg m^{-3})
Sim.
Obs.
Initial condition
37 minutes
1.45 hours
2.33 hours
3.1 hours
3.78 hours
(c) Calibrated flows (m^3 day^{-1})
Borehole water column
Inflows
Outflows
5.0
0.5
1.4
4.1