



## Review article

## Efficient 3D inversions using the Richards equation

Rowan Cockett<sup>\*,1</sup>, Lindsey J. Heagy, Eldad Haber

*Geophysical Inversion Facility, Department of Earth, Ocean and Atmospheric Sciences, University of British Columbia 6339 Stores Road British Columbia V6T 1Z4, Canada*

## ARTICLE INFO

## Keywords:

Vadose zone  
Inverse modelling  
Richards equation  
Parameter estimation

## ABSTRACT

Fluid flow in the vadose zone is governed by the Richards equation; it is parameterized by hydraulic conductivity, which is a nonlinear function of pressure head. Investigations in the vadose zone typically require characterizing distributed hydraulic properties. Water content or pressure head data may include direct measurements made from boreholes. Increasingly, proxy measurements from hydrogeophysics are being used to supply more spatially and temporally dense data sets. Inferring hydraulic parameters from such datasets requires the ability to efficiently solve and optimize the nonlinear time domain Richards equation. This is particularly important as the number of parameters to be estimated in a vadose zone inversion continues to grow. In this paper, we describe an efficient technique to invert for distributed hydraulic properties in 1D, 2D, and 3D. Our technique does not store the Jacobian matrix, but rather computes its product with a vector. Existing literature for the Richards equation inversion explicitly calculates the sensitivity matrix using finite difference or automatic differentiation, however, for large scale problems these methods are constrained by computation and/or memory. Using an implicit sensitivity algorithm enables large scale inversion problems for any distributed hydraulic parameters in the Richards equation to become tractable on modest computational resources. We provide an open source implementation of our technique based on the SimPEG framework, and show it in practice for a 3D inversion of saturated hydraulic conductivity using water content data through time.

## 1. Introduction

Studying the processes that occur in the vadose zone, the region between the earth's surface and the fully saturated zone, is of critical importance for understanding our groundwater resources. Fluid flow in the vadose zone is described by the Richards equation and parameterized by hydraulic conductivity, which is a nonlinear function of pressure head (Richards, 1931; Celia et al., 1990). Typically, hydraulic conductivity is heterogeneous and can have a large dynamic range. In any hydrogeological site characterization, the spatial estimation of the hydraulic conductivity function is an important step. Achieving this, however, requires the ability to efficiently solve and optimize the nonlinear, time-domain Richards equation. Rather than working with a full, implicit, 3D time-domain system of equations, simplifications are consistently used to avert the conceptual, practical, and computational difficulties inherent in the parameterization and inversion of the Richards equation. These simplifications typically parameterize the conductivity and assume that it is a simple function in space, often

adopting a homogeneous or layered soil profile (cf. Binley et al. (2002); Deiana et al. (2007); Hinnell et al. (2010); Liang and Uchida (2014)). Due to the lack of constraining hydrologic data, such assumptions are often satisfactory for fitting observed measurements, especially in 2D and 3D as well as in time. However, as more data become available, through spatially extensive surveys and time-lapse proxy measurements (e.g. direct current resistivity surveys and distributed temperature sensing), extracting more information about subsurface hydrogeologic parameters becomes a possibility. The proxy data can be directly incorporated through an empirical relation (e.g. Archie (1942)) or time-lapse estimations can be structurally incorporated through some sort of regularization technique (Haber and Gazit, 2013; Haber and Oldenburg, 1997; Hinnell et al., 2010). Recent advances have been made for the forward simulation of the Richards equation in a computationally-scalable manner (Orgozzo et al., 2014). However, the inverse problem is non-trivial, especially in 3D (Towara et al., 2015; Linde and Doetsch, 2016), and must be considered using modern numerical techniques that allow for spatial estimation of hydraulic parameters.

\* Corresponding author.

E-mail address: [rcockett@eoas.ubc.ca](mailto:rcockett@eoas.ubc.ca) (R. Cockett).

<sup>1</sup> Authorship Statement: Cockett and Haber developed the mathematical derivation and scope. Cockett and Heagy developed the software, numerical testing and performed the numerical simulations. Both Cockett and Heagy contributed to the final version of the manuscript; Haber supervised the project.

Inverse problems in space and time are often referred to as history matching problems (see Oliver and Chen (2011); Dean et al. (2008); Sarma et al. (2007); Oliver and Reynolds (2001); Šimunek and Senja (2012) and references within). These inverse problems use a flow simulation model, combined with some a-priori information, in order to estimate a spatially variable hydraulic conductivity function that approximately yields the observed data within its error bounds. The literature shows a variety of approaches for this inverse problem, including trial-and-error, stochastic methods, and various gradient based methods (Bitterlich et al., 2004; Binley et al., 2002; Carrick et al., 2010; Durner, 1994; Finsterle and Zhang, 2011; Mualem, 1976; Šimunek and van Genuchten, 1996). The way in which the computational complexity of the inverse method scales becomes important as problem size increases (Towara et al., 2015). Computational memory and time often become a bottleneck for solving the inverse problem, both when the problem is solved in 2D and, particularly, when it is solved in 3D (Haber et al., 2000). To solve the inverse problem, stochastic methods are often employed, which have an advantage in that they can examine the full parameter space and give insights into non-uniqueness (Finsterle and Kowalsky, 2011). However, as the number of parameters we seek to recover in an inversion increases, these stochastic methods require that the forward problem be solved many times, which often makes these methods impractical or ‘computationally infeasible’ (Linde and Doetsch, 2016). This scalability, especially in the context of hydrogeophysics has been explicitly noted in the literature (cf. Binley et al. (2002); Deiana et al. (2007); Towara et al. (2015); Linde and Doetsch (2016)).

Derivative-based optimization techniques become a practical alternative when the forward problem is computationally expensive or when there are many parameters to estimate (i.e. thousands to millions). Inverse problems are ill-posed and thus to pose a solvable optimization problem, an appropriate regularization is combined with a measure of the data misfit to state a deterministic optimization problem (Tikhonov and Arsenin, 1977). Alternatively, if prior information can be formulated using a statistical framework, we can use Bayesian techniques to obtain an estimator through the Maximum A Posteriori model (MAP) (Kaipio and Somersalo, 2004). In the context of Bayesian estimation, gradient based methods are also important, as they can be used to efficiently sample the posterior (Bui-Thanh and Ghattas, 2015; Liu et al., 2017; Klein et al., 2017).

A number of authors have sought solutions for the inverse problem, where the forward problem is the Richards equation (cf. Bitterlich and Knabner (2002); Iden and Durner (2007); Šimunek and Senja (2012) and references within). Since the problem is parabolic (therefore stiff), most work discretizes the forward problem by an implicit method in time and a finite volume or finite element in space. Most work uses a Newton-like method for the resulting nonlinear system, which arises from the discretization of the forward problem. For the deterministic inverse problem using the Richards equation, previous work uses some version of a Gauss-Newton method (e.g. Levenberg-Marquardt), with a **direct** calculation of the sensitivity matrix (Finsterle and Kowalsky, 2011; Šimunek and van Genuchten, 1996; Bitterlich and Knabner, 2002). However, while these approaches allow for inversions of moderate scale (i.e. 1D and 2D), they have one major drawback: the sensitivity matrix is large and dense; its computation requires dense linear algebra and a non-trivial amount of memory (cf. Towara et al. (2015)). Previous work used either external numerical differentiation (e.g. PEST) or automatic differentiation in order to directly compute the sensitivity matrix (Finsterle and Zhang, 2011; Bitterlich and Knabner, 2002; Doherty, 2015; Towara et al., 2015; Liu et al., 2017). External numerical differentiation is computationally intensive and limits the number of model parameters that can be estimated.

The goal of this paper is to suggest a modern numerical formulation that allows the inverse problem to be solved **without explicit** computation of the sensitivity matrix by using **exact** derivatives of the discrete formulation (Haber et al., 2000). Our technique is based on the

discretize-then-optimize approach, which discretizes the forward problem first and then uses a deterministic optimization algorithm to solve the inverse problem (Gunzburger, 2003). To this end, we require the discretization of the forward problem. Similar to the work of Celia et al. (1990), we use an implicit Euler method in time and finite volume in space. Given the discrete form, we show that we can analytically compute the derivatives of the forward problem with respect to distributed hydraulic parameters and, as a result, obtain an implicit formula for the sensitivity. The formula involves the solution of a linear time-dependent problem; we avoid computing and storing the sensitivity matrix directly and, rather, suggest a method to efficiently compute the product of the sensitivity matrix and its adjoint times a vector. Equipped with this formulation, we can use a standard inexact Gauss-Newton method to solve the inverse problem for distributed hydraulic parameters in 3D. This large-scale distributed parameter estimation becomes computationally tractable with the technique presented in this paper and can be employed with any iterative Gauss-Newton-like optimization technique.

This paper is structured as follows: in Section 2, we discuss the discretization of the forward problem on a staggered mesh in space and backward Euler in time; in Section 3, we formulate the inverse problem and construct the implicit functions used for computations of the Jacobian-vector product. In Section 4, we demonstrate the validity of the implementation of the forward problem and sensitivity calculation. Finally, in Section 5, we show an example of a 3D inversion for hydraulic conductivity and discuss extensions for inverting for multiple distributed hydraulic parameters from the Richards equation and contrast the scalability of our methodology to standard techniques.

To accelerate both the development and dissemination of this approach, we have built these tools on top of an open source framework for organizing simulation and inverse problems in geophysics (SimPEG) (Cockett et al., 2015). We have released our numerical implementation under the permissive MIT license. Our implementation of the implicit sensitivity calculation for the Richards equation and associated inversion implementation is provided and tested to support 1D, 2D, and 3D forward and inverse simulations with respect to custom empirical relations and sensitivity to any parameters within these functions. The source code can be found at <https://github.com/simpeg/simpeg> and may be a helpful resource for researchers looking to use or extend our implementation.

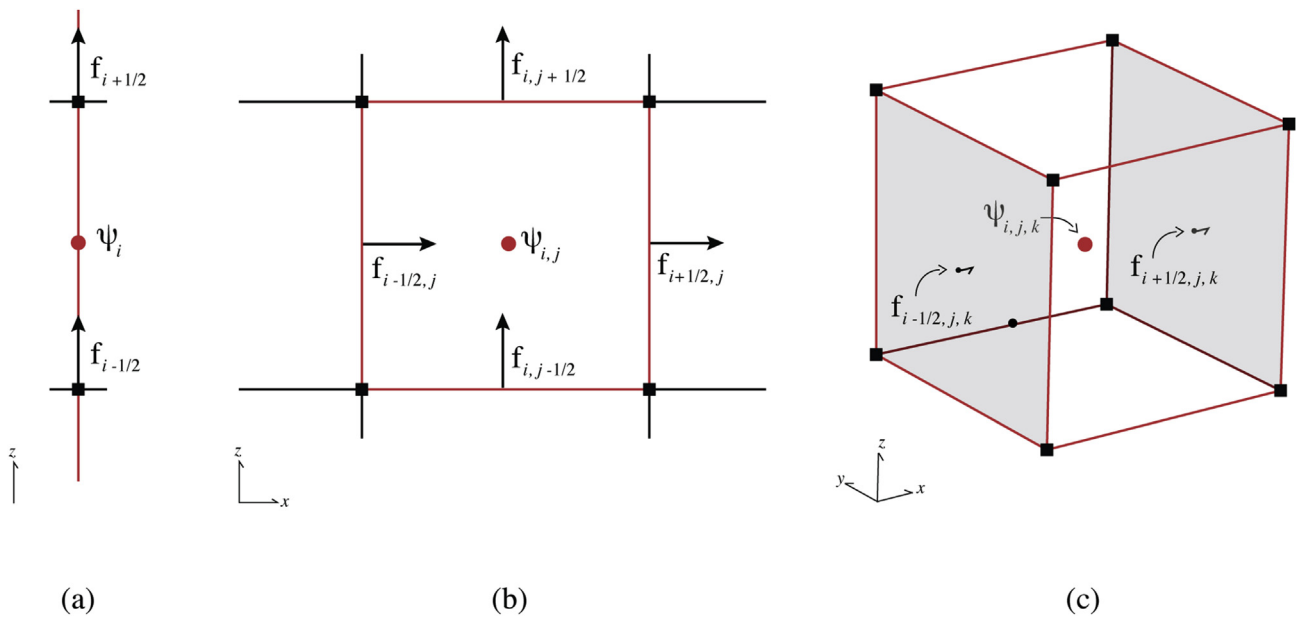
## 2. Forward problem

In this section, we describe the Richards equation and its discretization (Richards, 1931). The Richards equation is a nonlinear parabolic partial differential equation (PDE) and we follow the so-called mixed formulation presented in Celia et al. (1990) with some modifications. In the derivation of the discretization, we give special attention to the details used to efficiently calculate the effect of the sensitivity on a vector, which is needed in any derivative based optimization algorithm.

### 2.1. Richards equation

The parameters that control groundwater flow depend on the effective saturation of the media, which leads to a nonlinear problem. The groundwater flow equation has a diffusion term and an advection term which is related to gravity and only acts in the  $z$ -direction. There are two different forms of the Richards equation; they differ in how they deal with the nonlinearity in the time-stepping term. Here, we use the most fundamental form, referred to as the ‘mixed’-form of the Richards equation (Celia et al., 1990):

$$\frac{\partial \theta(\psi)}{\partial t} - \nabla \cdot k(\psi) \nabla \psi - \frac{\partial k(\psi)}{\partial z} = 0 \quad \psi \in \Omega \quad (1)$$



**Figure 1.** Discretization of unknowns in 1D, 2D and 3D space. Red circles are the locations of the discrete hydraulic conductivity  $K$  and the pressure head  $\psi$ . The arrows are the locations of the discretized flux  $\vec{f}$  on each cell face. Modified after Cockett et al. (2016). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

where  $\psi$  is pressure head,  $\theta(\psi)$  is volumetric water content, and  $k(\psi)$  is hydraulic conductivity. This formulation of the Richards equation is called the ‘mixed’-form because the equation is parameterized in  $\psi$  but the time-stepping is in terms of  $\theta$ . The hydraulic conductivity,  $k(\psi)$ , is a heterogeneous and potentially anisotropic function that is assumed to be known when solving the forward problem. In this paper, we assume that  $k$  is isotropic, but the extension to anisotropy is straightforward (Cockett et al., 2015, 2016). The equation is solved in a domain,  $\Omega$ , equipped with boundary conditions on  $\partial\Omega$  and initial conditions, which are problem-dependent.

An important aspect of unsaturated flow is noticing that both water content,  $\theta$ , and hydraulic conductivity,  $k$ , are functions of pressure head,  $\psi$ . There are many empirical relations used to relate these parameters, including the Brooks-Corey model (Brooks and Corey, 1964) and the van Genuchten-Mualem model (Mualem, 1976; van Genuchten, 1980). The van Genuchten model is written as:

$$\theta(\psi) = \begin{cases} \theta_r + \frac{\theta_s - \theta_r}{(1 + |\alpha\psi|^n)^m} & \psi < 0 \\ \theta_s & \psi \geq 0 \end{cases} \quad (2a)$$

$$k(\psi) = \begin{cases} K_s \theta_e(\psi)^l \left(1 - (1 - \theta_e(\psi)^{-m})^m\right)^2 & \psi < 0 \\ K_s & \psi \geq 0 \end{cases} \quad (2b)$$

where

$$\theta_e(\psi) = \frac{\theta(\psi) - \theta_r}{\theta_s - \theta_r}, \quad m = 1 - \frac{1}{n}, \quad n > 1 \quad (3)$$

Here,  $\theta_r$  and  $\theta_s$  are the residual and saturated water contents,  $K_s$  is the saturated hydraulic conductivity,  $\alpha$  and  $n$  are fitting parameters, and,  $\theta_e(\psi) \in [0, 1]$  is the effective saturation. The pore connectivity parameter,  $l$ , is often taken to be  $\frac{1}{2}$ , as determined by Mualem (1976). Pressure head varies over the domain  $\psi \in (-\infty, 0)$ . When the value is close to zero, the soil behaves most like a saturated soil where  $\theta = \theta_s$  and  $k = K_s$ . Small changes in pressure head can change the hydraulic conductivity by several orders of magnitude; as such,  $k(\psi)$  is a highly nonlinear function, making the Richards equation a nonlinear PDE.

## 2.2. Discretization

The Richards equation is parameterized in terms of pressure head,  $\psi$ . Here, we describe simulating the Richards equation in 1D, 2D, and 3D. We start by discretizing in space and then we discretize in time. This process yields a discrete, nonlinear system of equations; for its solution, we discuss a variation of Newton’s method.

### 2.2.1. Spatial discretization

In order to conservatively discretize the Richards equation, we introduce the flux  $\vec{f}$  and rewrite the equation as a first order system of the form:

$$\frac{\partial\theta(\psi)}{\partial t} - \nabla \cdot \vec{f} - \frac{\partial k(\psi)}{\partial z} = 0 \quad (4a)$$

$$k(\psi)^{-1} \vec{f} = \nabla\psi \quad (4b)$$

We then discretize the system using a standard staggered finite volume discretization (cf. Ascher (2008); Haber (2015); Cockett et al. (2016)). This discretization is a natural extension of mass-conservation in a volume where the balance of fluxes into and out of a volume are conserved (Lipnikov and Misitas, 2013). Here, it is natural to assign the entire cell one hydraulic conductivity value,  $k$ , which is located at the cell center. Such assigning leads to a piecewise constant approximation for the hydraulic conductivity and allows for discontinuities between adjacent cells. From a geologic perspective, discontinuities are prevalent, as it is possible to have large differences in hydraulic properties between geologic layers in the ground. The pressure head,  $\psi$ , is also located at the cell centers and the fluxes are located on cell faces, which lead to the usual staggered mesh or Marker and Cell (MAC) discretization in space (Fletcher, 1988). We demonstrate the discretization in 1D, 2D and 3D on the tensor mesh in Figure 1. We discretize the function,  $\psi$ , on a cell-centered grid, which results in a grid function,  $\psi$ . We use bold letters to indicate other grid functions.

The discretization of a diffusion-like equation on an orthogonal mesh is well-known (see Haber and Ascher (2001); Fletcher (1988); Haber et al. (2007); Ascher and Greif (2011) and references within). We discretize the differential operators by using the usual mass balance

consideration and the elimination of the flux,  $\mathbf{f}$ .<sup>2</sup> This spatial discretization leads to the following discrete nonlinear system of ordinary differential equations (assuming homogeneous Dirichlet boundary conditions):

$$\frac{d\theta(\boldsymbol{\psi})}{dt} - \mathbf{D} \text{diag}(\mathbf{k}_{A_v}(\boldsymbol{\psi}^{n+1}))\mathbf{G}\boldsymbol{\psi} - \mathbf{G}_z(\mathbf{k}_{A_v}(\boldsymbol{\psi}^{n+1})) = 0 \quad (5)$$

Here,  $\mathbf{D}$  is the discrete divergence operator and  $\mathbf{G}$  is the discrete gradient operator. The discrete derivative in the  $z$ -direction is written as  $\mathbf{G}_z$ . The values of  $\psi$  and  $k(\psi)$  are known on the cell-centers and must be averaged to the cell-faces, which we complete through harmonic averaging (Haber and Ascher, 2001).

$$\mathbf{k}_{A_v}(\boldsymbol{\psi}) = \frac{1}{\mathbf{A}_v \mathbf{k}(\boldsymbol{\psi})} \quad (6)$$

where  $\mathbf{A}_v$  is a matrix that averages from cell-centers to faces and the division of the vector is done pointwise; that is, we use the vector notation,  $(1/v)_i = 1/v_i$ . We incorporate boundary conditions using a ghost-point outside of the mesh (Trottenberg et al., 2001).

### 2.2.2. Time discretization and stepping

The Richards equation is often used to simulate water infiltrating an initially dry soil. At early times in an infiltration experiment, the pressure head,  $\psi$ , can be close to discontinuous. These large changes in  $\psi$  are also reflected in the nonlinear terms  $k(\psi)$  and  $\theta(\psi)$ ; as such, the initial conditions imposed require that an appropriate time discretization be chosen. Hydrogeologists are often interested in the complete evolutionary process, until steady-state is achieved, which may take many time-steps. Here, we describe the implementation of a fully-implicit backward Euler numerical scheme. Higher-order implicit methods are not considered here because the uncertainty associated with boundary conditions and the fitting parameters in the Van Genuchten models (eq. (2)) have much more effect than the order of the numerical method used.

The discretized approximation to the mixed-form of the Richards equation, using fully-implicit backward Euler, reads:

$$\begin{aligned} F(\boldsymbol{\psi}^{n+1}, \boldsymbol{\psi}^n) &= \frac{\theta(\boldsymbol{\psi}^{n+1}) - \theta(\boldsymbol{\psi}^n)}{\Delta t} - \mathbf{D} \text{diag}(\mathbf{k}_{A_v}(\boldsymbol{\psi}^{n+1}))\mathbf{G}\boldsymbol{\psi}^{n+1} \\ &\quad - \mathbf{G}_z(\mathbf{k}_{A_v}(\boldsymbol{\psi}^{n+1})) \\ &= 0 \end{aligned} \quad (7)$$

This is a nonlinear system of equations for  $\boldsymbol{\psi}^{n+1}$  that needs to be solved numerically by some iterative process. Either a Picard iteration (as in Celia et al. (1990)) or a Newton root-finding iteration with a step length control can be used to solve the system. Note that to deal with dependence of  $\theta$  with respect to  $\psi$  in Newton's method, we require the computation of  $\frac{d\theta}{d\psi}$ . We can complete this computation by using the analytic form of the hydraulic conductivity and water content functions (e.g. derivatives of eq. (2)). We note that a similar approach can be used for any smooth curve, even when the connection between  $\theta$  and  $\psi$  are determined empirically (for example, when  $\theta(\psi)$  is given by a spline interpolation of field data).

### 2.3. Solving the nonlinear equations

Regardless of the empirical relation chosen, we must solve eq. (7) using an iterative root-finding technique. Newton's method iterates over  $m = 1, 2, \dots$  until a satisfactory estimation of  $\boldsymbol{\psi}^{n+1}$  is obtained. Given  $\boldsymbol{\psi}^{n+1,m}$ , we approximate  $F(\boldsymbol{\psi}^{n+1}, \boldsymbol{\psi}^n)$  as:

$$F(\boldsymbol{\psi}^{n+1}, \boldsymbol{\psi}^n) \approx F(\boldsymbol{\psi}^{n+1,m}, \boldsymbol{\psi}^n) + \mathbf{J}_{\boldsymbol{\psi}^{n+1,m}} \delta \boldsymbol{\psi} \quad (8)$$

where the Jacobian for iteration,  $m$ , is:

$$\mathbf{J}_{\boldsymbol{\psi}^{n+1,m}} = \left. \frac{\partial F(\boldsymbol{\psi}, \boldsymbol{\psi}^n)}{\partial \boldsymbol{\psi}} \right|_{\boldsymbol{\psi}^{n+1,m}} \quad (9)$$

The Jacobian is a large dense matrix, and its computation necessitates the computation of the derivatives of  $F(\boldsymbol{\psi}^{n+1,m}, \boldsymbol{\psi}^n)$ . We can use numerical differentiation in order to evaluate the Jacobian (or its product with a vector). However, in the context of the inverse problem, an exact expression is preferred. Given the discrete forward problem, we obtain that:

$$\begin{aligned} \mathbf{J}_{\boldsymbol{\psi}^{n+1,m}} &= \frac{1}{\Delta t} \frac{d\theta(\boldsymbol{\psi}^{n+1,m})}{d\boldsymbol{\psi}^{n+1,m}} - \frac{d}{d\boldsymbol{\psi}^{n+1,m}} (\mathbf{D} \text{diag}(\mathbf{k}_{A_v}(\boldsymbol{\psi}^{n+1,m}))\mathbf{G}\boldsymbol{\psi}^{n+1,m}) \\ &\quad - \mathbf{G}_z \frac{d\mathbf{k}_{A_v}(\boldsymbol{\psi}^{n+1,m})}{d\boldsymbol{\psi}^{n+1,m}} \end{aligned} \quad (10)$$

Here, recall that  $\mathbf{k}_{A_v}$  is harmonically averaged and its derivative can be obtained by the chain rule:

$$\frac{d\mathbf{k}_{A_v}(\boldsymbol{\psi})}{d\boldsymbol{\psi}} = \text{diag}((\mathbf{A}_v \mathbf{k}^{-1}(\boldsymbol{\psi}))^{-2}) \mathbf{A}_v \text{diag}(\mathbf{k}^{-2}(\boldsymbol{\psi})) \frac{d\mathbf{k}(\boldsymbol{\psi})}{d\boldsymbol{\psi}} \quad (11)$$

Similarly, for the second term in eq. (10) we obtain:

$$\frac{\partial}{\partial \boldsymbol{\psi}} (\mathbf{D} \text{diag}(\mathbf{k}_{A_v}(\boldsymbol{\psi}))\mathbf{G}\boldsymbol{\psi}) = \mathbf{D} \text{diag}(\mathbf{k}_{A_v}(\boldsymbol{\psi}))\mathbf{G} + \mathbf{D} \text{diag}(\mathbf{G}\boldsymbol{\psi}) \frac{\partial \mathbf{k}_{A_v}(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} \quad (12)$$

Here the notation  $^{n+1,m}$  has been dropped for brevity. For the computations above, we need the derivatives of functions  $\mathbf{k}(\boldsymbol{\psi})$  and  $\theta(\boldsymbol{\psi})$ ; note that, since the relations are assumed local (point wise in space) given the vector,  $\boldsymbol{\psi}$ , these derivatives are diagonal matrices. For Newton's method, we solve the linear system:

$$\mathbf{J}_{\boldsymbol{\psi}^{n+1,m}} \delta \boldsymbol{\psi} = -F(\boldsymbol{\psi}^{n+1,m}, \boldsymbol{\psi}^n) \quad (13)$$

For small-scale problems, we can solve the linear system using direct methods; however, for large-scale problems, iterative methods are more commonly used. The existence of an advection term in the PDE results in a non-symmetric linear system. Thus, when using iterative techniques to solve this system, an appropriate iterative method, such as BIGSTAB or GMRES (Saad, 1996; Barrett et al., 1994), must be used.

At this point, it is interesting to note the difference between the Newton iteration and the Picard iteration suggested in Celia et al. (1990). We can verify that the Picard iteration uses an approximation to the Jacobian  $\mathbf{J}_{\boldsymbol{\psi}^{n+1,m}} \delta \boldsymbol{\psi}$  given by dropping the second term from (14). This term can have negative eigenvalues and dropping it is typically done when considering the lagged diffusivity method (Vogel, 2001). However, as discussed in Vogel (2001), ignoring this term can slow convergence.

Finally, a new iterate is computed by adding the Newton update to the last iterate:

$$\boldsymbol{\psi}^{n+1,m+1} = \boldsymbol{\psi}^{n+1,m} + \alpha \delta \boldsymbol{\psi}$$

where  $\alpha$  is a parameter that guarantees that

$$\|F(\boldsymbol{\psi}^{n+1,m+1}, \boldsymbol{\psi}^n)\| < \|F(\boldsymbol{\psi}^{n+1,m}, \boldsymbol{\psi}^n)\|$$

To obtain  $\alpha$ , we perform an Armijo line search (Nocedal and Wright, 1999). In our numerical experiments, we have found that this method can fail when the hydraulic conductivity is strongly discontinuous and changes rapidly. In such cases, Newton's method yields a poor descent direction. Therefore, if the Newton iteration fails to converge to a solution, the update is performed with the mixed-form Picard iteration. Note that Picard iteration can be used, even when Newton's method fails, because Picard iteration always yields a descent direction (Vogel, 2001).

At this point, we have discretized the Richards equation in both time

<sup>2</sup> Here we assume an isotropic conductivity that leads to a diagonal mass matrix and this yields easy elimination of the fluxes.

and space while devoting special attention to the derivatives necessary in Newton's method and the Picard iteration as described in Celia et al. (1990). The exact derivatives of the discrete problem will be used in the following two sections, which outline the implicit formula for the sensitivity and its incorporation into a general inversion algorithm.

### 3. Inverse problem

The location and spatial variability of, for example, an infiltration front over time is inherently dependent on the hydraulic properties of the soil column. As such, direct or proxy measurements of the water content or pressure head at various depths along a soil profile contain information about the soil properties. We pose the inverse problem, which is the estimation of distributed hydraulic parameters, given either water content or pressure data. We frame this problem under the assumption that we wish to estimate hundreds of thousands to millions of distributed model parameters. Due to the large number of model parameters that we aim to estimate in this inverse problem, stochastic techniques or external numerical differentiation, such as the popular PEST toolbox (Doherty, 2015), are not computationally feasible. Instead, we will employ a direct method by calculating the exact derivatives of the discrete Richards equation and solving the sensitivity implicitly. For brevity, we show the derivation of the sensitivity for an inversion model of only saturated hydraulic conductivity,  $K_s$ , from pressure head data,  $\mathbf{d}_{\text{obs}}$ . This derivation can be readily extended to include the use of water content data and inverted for other distributed parameters in the heterogeneous hydraulic conductivity function. We will demonstrate the sensitivity calculation for multiple distributed parameters in the numerical examples (Section 5).

The Richards equation simulation produces a pressure head field at all points in space as well as through time. Data can be predicted,  $\mathbf{d}_{\text{pred}}$ , from these fields and compared to observed data,  $\mathbf{d}_{\text{obs}}$ . To be more specific, we let  $\Psi = [(\psi^1)^T, \dots, (\psi^{n_t})^T]^T$  be the (discrete) pressure field for all space and  $n_t$  time steps. When measuring pressure head recorded only in specific locations and times, we define the predicted data,  $\mathbf{d}_{\text{pred}}$ , as  $\mathbf{d}_{\text{pred}} = \mathbf{P}\Psi(\mathbf{m})$ . Here, the vector  $\mathbf{m}$  is the vector containing all of the parameters which we are inverting for (e.g.  $K_s$ ,  $\alpha$ ,  $n$ ,  $\theta_r$ , or  $\theta_s$  when using the van Genuchten empirical relation). The matrix,  $\mathbf{P}$ , interpolates the pressure head field,  $\Psi$ , to the locations and times of the measurements. Since we are using a simple finite volume approach and backward Euler in time, we use linear interpolation in both space and time to compute  $\mathbf{d}_{\text{pred}}$  from  $\Psi$ . Thus, the entries of the matrix  $\mathbf{P}$  contain the interpolation weights. For linear interpolation in 3D,  $\mathbf{P}$  is a sparse matrix that contains up to eight non-zero entries in each row. Note that the time and location of the data measurement is independent and decoupled from the numerical discretization used in the forward problem. A water retention curve, such as the van Genuchten model, can be used for computation of predicted water content data, which requires another nonlinear transformation,  $\mathbf{d}_{\text{pred}}^\theta = \mathbf{P}\theta(\Psi(\mathbf{m}), \mathbf{m})$ . Note here that the transformation to water content data, in general, depends on the model to be estimated in the inversion, which will be addressed in the numerical examples. For brevity in the derivation that follows, we will make two simplifying assumptions: (1) that the data are pressure head measurements, which requires a linear interpolation that is not dependent on the model; and, (2) that the model vector,  $\mathbf{m}$ , describes only distributed saturated hydraulic conductivity. Our software implementation *does not* make these assumptions; our numerical examples will use water content data, a variety of empirical relations, and calculate the sensitivity to multiple heterogeneous empirical parameters.

#### 3.1. Solution through optimization

We can now formulate the discrete inverse problem to estimate saturated hydraulic conductivity,  $\mathbf{m}$ , from the observed pressure head data,  $\mathbf{d}_{\text{obs}}$ . We frame the inversion as an optimization problem, which minimizes a data misfit and a regularization term.

$$\hat{\mathbf{m}} = \underset{\mathbf{m}}{\text{argmin}} \Phi(\mathbf{m}) = \frac{1}{2} \|\mathbf{W}_d(\mathbf{d}_{\text{pred}}(\mathbf{m}) - \mathbf{d}_{\text{obs}})\|_2^2 + \frac{\beta}{2} \|\mathbf{W}_m(\mathbf{m} - \mathbf{m}_{\text{ref}})\|_2^2. \quad (14)$$

The first term in the objective function is the data misfit,  $\phi_d$ ; it contains a weighted difference between the observed and predicted data. Assuming that the observed data is noisy, with independent distributed Gaussian noise having standard deviation  $\sigma$ , the weighting matrix,  $\mathbf{W}_d$ , is a diagonal matrix that contains the entries  $\sigma_i^{-1}$  on its diagonal. The matrix,  $\mathbf{W}_m$ , is a regularization matrix that is chosen based on a-priori information and assumptions about the geologic setting (cf. Oldenburg and Li (2005)). In a regularized inversion framework, if we estimate only the log of the static conductivity,  $K_s$  in (2.1), then we typically choose  $\mathbf{W}_m$  as a combination of the first order gradient matrix and an identity matrix. In a Bayesian framework, where one assumes that  $\mathbf{m}$  is Gaussian, the matrix  $\mathbf{W}_m^{-1}\mathbf{W}_m$  is the inverse covariance matrix. If  $\mathbf{m}$  contains more than a single distributed hydrogeologic parameter, then a correlation term is typically added (Haber and Gazit, 2013). The reference model,  $\mathbf{m}_{\text{ref}}$ , is chosen based on a-priori information about the background. Finally, we add the misfit and regularization terms using a trade-off parameter,  $\beta$ , to balance the influence of a-priori information, or smoothness, with the fit to noisy data. This is the standard approach in geophysical inversions (Tikhonov and Arsenin, 1977; Oldenburg and Li, 2005; Constable et al., 1987; Haber, 2015) where hundreds of thousands to millions of distributed parameters are commonly estimated in a deterministic inversion. The hydrogeologic literature also shows the use of these techniques; however, there is also a large community advancing stochastic inversion techniques and geologic realism (cf. Linde et al. (2015)). Regardless of the inversion algorithm used, an efficient method to calculate the sensitivity is crucial; this method is the focus of our work.

To minimize the objective function, we use an inexact Gauss-Newton method that is well-suited for large-scale problems (Nocedal and Wright, 1999). For the Gauss-Newton method, the gradient of equation (14) is computed and iteratively driven towards zero by using an approximate Hessian,  $\mathbf{H}$ . For the problem at hand, the gradient can be expressed as:

$$\nabla_{\mathbf{m}}\Phi(\mathbf{m}) = \mathbf{J}^T \mathbf{W}_d^T \mathbf{W}_d (\mathbf{d}_{\text{pred}}(\mathbf{m}) - \mathbf{d}_{\text{obs}}) + \beta \mathbf{W}_m^T \mathbf{W}_m (\mathbf{m} - \mathbf{m}_{\text{ref}}) \quad (15)$$

where the sensitivity matrix,  $\mathbf{J} = \nabla_{\mathbf{m}}\mathbf{d}_{\text{pred}}(\mathbf{m})$ , is the derivative of the predicted data with respect to the model parameters. The Hessian of the objective function,  $\mathbf{H}$ , is approximated with the first order terms and is guaranteed to be positive definite.

$$\nabla_{\mathbf{m}}^2\Phi(\mathbf{m}) \approx \mathbf{H} = \mathbf{J}^T \mathbf{W}_d^T \mathbf{W}_d \mathbf{J} + \beta \mathbf{W}_m^T \mathbf{W}_m \quad (16)$$

Finally, the (inexact) Gauss-Newton update,  $\delta\mathbf{m}$ , is computed by (approximately) solving the system

$$\mathbf{H}\delta\mathbf{m} = -\nabla_{\mathbf{m}}\Phi(\mathbf{m})$$

using some iterative technique. In this work, we use the preconditioned conjugate gradient (PCG) method, allowing us to work with large-scale problems. To precondition the conjugate gradient algorithm, we use a standard limited-memory BFGS algorithm initiated with the inverse of  $\mathbf{W}_m^T \mathbf{W}_m$  instead of the identity. For more details on the preconditioner, see (Haber, 2005).

It is important to note that the sensitivity matrix,  $\mathbf{J}$ , as well as the approximate Hessian,  $\mathbf{H}$ , are large, dense matrices and their explicit computations result in an algorithm that is constrained by computational memory. However, as we show in the following section, one can avoid this computation and efficiently compute the products of  $\mathbf{J}$  and  $\mathbf{J}^T$  with a vector. This computation, coupled with an iterative optimization method, results in an efficient algorithm.

Once the model update,  $\delta\mathbf{m}$ , is found, a Gauss-Newton step is taken:

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \delta\mathbf{m}$$

where  $\alpha_k$  is the current line search parameter. We use a backtracking Armijo line search (Armijo, 1966) to enforce sufficient decrease in our objective function (Nocedal and Wright, 1999). We repeat the minimization procedure, which updates the linearization until the norm of the gradient falls below a certain tolerance, the model update is sufficiently small, or until we have exhausted the maximum number of iterations. Values for these stopping criteria are often problem-dependent. Running synthetic models with elements similar to observed data can often give insight into appropriate stopping criteria (Aster et al., 2004).

### 3.2. Implicit sensitivity calculation

The optimization problem requires the derivative of the pressure head with respect to the model parameters,  $\frac{\partial \Psi}{\partial \mathbf{m}}$ . An approximation of the sensitivity matrix can be obtained through a finite difference method on the forward problem (Simunek and van Genuchten, 1996; Finsterle and Kowalsky, 2011; Finsterle and Zhang, 2011). One forward problem, or two, when using central differences, must be completed for each col-

Note that  $\Psi^{n+1}$  and  $\Psi^n$  are also functions of  $\mathbf{m}$ . In general,  $\theta^{n+1}$  and  $\theta^n$  are also dependent on the model; however, for brevity, we will omit these derivatives. The derivatives of  $F$  to the change in the parameters  $\mathbf{m}$  can be written as:

$$\nabla_{\mathbf{m}} F(\Psi^n, \Psi^{n+1}, \mathbf{m}) = \frac{\partial F}{\partial \mathbf{k}_{Av}} \frac{\partial \mathbf{k}_{Av}}{\partial \mathbf{m}} + \frac{\partial F}{\partial \Psi^n} \frac{\partial \Psi^n}{\partial \mathbf{m}} + \frac{\partial F}{\partial \Psi^{n+1}} \frac{\partial \Psi^{n+1}}{\partial \mathbf{m}} = 0 \quad (18)$$

or, in more detail:

$$\begin{aligned} \frac{1}{\Delta t} \left( \frac{\partial \theta^{n+1}}{\partial \Psi^{n+1}} \frac{\partial \Psi^{n+1}}{\partial \mathbf{m}} - \frac{\partial \theta^n}{\partial \Psi^n} \frac{\partial \Psi^n}{\partial \mathbf{m}} \right) - \mathbf{D} \operatorname{diag}(\mathbf{G}\Psi^{n+1}) \left( \frac{\partial \mathbf{k}_{Av}}{\partial \mathbf{m}} + \frac{\partial \mathbf{k}_{Av}}{\partial \Psi^{n+1}} \frac{\partial \Psi^{n+1}}{\partial \mathbf{m}} \right) \\ - \mathbf{D} \operatorname{diag}(\mathbf{k}_{Av}(\Psi^{n+1})) \mathbf{G} \frac{\partial \Psi^{n+1}}{\partial \mathbf{m}} - \mathbf{G}_z \left( \frac{\partial \mathbf{k}_{Av}}{\partial \mathbf{m}} + \frac{\partial \mathbf{k}_{Av}}{\partial \Psi^{n+1}} \frac{\partial \Psi^{n+1}}{\partial \mathbf{m}} \right) \\ = 0 \end{aligned} \quad (19)$$

The above equation is a linear system of equations and, to solve for  $\frac{\partial \Psi}{\partial \mathbf{m}}$ , we rearrange the block-matrix equation:

$$\overbrace{\left[ \frac{1}{\Delta t} \frac{\partial \theta^{n+1}}{\partial \Psi^{n+1}} - \mathbf{D} \operatorname{diag}(\mathbf{G}\Psi^{n+1}) \frac{\partial \mathbf{k}_{Av}}{\partial \Psi^{n+1}} - \mathbf{D} \operatorname{diag}(\mathbf{k}_{Av}(\Psi^{n+1}, \mathbf{m})) \mathbf{G} - \mathbf{G}_z \frac{\partial \mathbf{k}_{Av}}{\partial \Psi^{n+1}} \right]}^{A_0(\Psi^{n+1})} \frac{\partial \Psi^{n+1}}{\partial \mathbf{m}} + \underbrace{\left[ -\frac{1}{\Delta t} \frac{\partial \theta^n}{\partial \Psi^n} \right]}_{A_{-1}(\Psi^n)} \frac{\partial \Psi^n}{\partial \mathbf{m}} = \underbrace{\left[ -\mathbf{D} \operatorname{diag}(\mathbf{G}\Psi^{n+1}) \frac{\partial \mathbf{k}_{Av}}{\partial \mathbf{m}} - \mathbf{G}_z \frac{\partial \mathbf{k}_{Av}}{\partial \mathbf{m}} \right]}_{B(\Psi^{n+1})} \quad (20)$$

umn in the Jacobian at every iteration of the optimization algorithm. This style of differentiation proves advantageous in that it can be applied to any forward problem; however, it is highly inefficient and introduces errors into the inversion that may slow the convergence of the scheme (Doherty, 2015). Automatic differentiation (AD) can also be used (Nocedal and Wright, 1999). Bitterlich and Knabner (2002) present three algorithms (finite difference, adjoint, and direct) to directly compute the elements of the dense sensitivity matrix for the Richards equation. As problem size increases, the memory required to store this dense matrix often becomes a practical computational limitation (Haber et al., 2004; Towara et al., 2015). As we show next, it is possible to explicitly write the derivatives of the Jacobian and evaluate their products with vectors using only sparse matrix operations. The algorithm computes matrix-vector and adjoint matrix-vector products with the Jacobian matrix. We can use these products for the solution of the Gauss-Newton system when using the conjugate gradient method, which bypasses the need for the direct calculation of the sensitivity matrix. Other geophysical inverse problems have used this idea extensively, especially in large-scale electromagnetics (cf. Haber et al. (2000)). The challenge in both the derivation and implementation for the Richards equation lies in differentiating the nonlinear time-dependent forward simulation with respect to multiple distributed hydraulic parameters.

The approach to implicitly constructing the derivative of the Richards equation in time involves writing the whole time-stepping process as a block bi-diagonal matrix system. The discrete Richards equation can be written as a function of the model. For a single time-step, the equation is written:

$$\begin{aligned} F(\Psi^{n+1}(\mathbf{m}), \Psi^n(\mathbf{m}), \mathbf{m}) &= \frac{\theta^{n+1}(\Psi^{n+1}) - \theta^n(\Psi^n)}{\Delta t} - \mathbf{D} \operatorname{diag}(\mathbf{k}_{Av}(\Psi^{n+1}, \mathbf{m})) \\ &\quad \times \mathbf{G}\Psi^{n+1} - \mathbf{G}_z \mathbf{k}_{Av}(\Psi^{n+1}, \mathbf{m}) \\ &= 0 \end{aligned} \quad (17)$$

In this case,  $\mathbf{m}$  is a vector that contains all the parameters of interest.

Here, we use the subscript notation of  $A_0(\Psi^{n+1})$  and  $A_{-1}(\Psi^n)$  to represent two block-diagonals of the large sparse matrix  $\mathbf{A}(\Psi, \mathbf{m})$ . Note that all of the terms in these matrices are already evaluated when computing the Jacobian of the Richards equations in Section 2 and that they contain only basic sparse linear algebra manipulations without the inversion of any matrix. If  $\Psi_0$  does not depend on the model, meaning the initial conditions are independent, then we can formulate the block system as:

$$\underbrace{\begin{bmatrix} A_0(\Psi_1) & & & & & & \\ A_{-1}(\Psi_1) & A_0(\Psi_2) & & & & & \\ & A_{-1}(\Psi_2) & A_0(\Psi_3) & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & A_{-1}(\Psi_{n-1}) & A_0(\Psi_n) & & \end{bmatrix}}_{\mathbf{A}(\Psi, \mathbf{m})} \underbrace{\begin{bmatrix} \frac{\partial \Psi}{\partial \mathbf{m}} \\ \frac{\partial \Psi_1}{\partial \mathbf{m}} \\ \frac{\partial \Psi_2}{\partial \mathbf{m}} \\ \vdots \\ \frac{\partial \Psi_{n-1}}{\partial \mathbf{m}} \\ \frac{\partial \Psi_n}{\partial \mathbf{m}} \end{bmatrix}}_{\mathbf{B}(\Psi, \mathbf{m})} = \underbrace{\begin{bmatrix} B_1(\Psi_1) \\ B_2(\Psi_2) \\ \vdots \\ B_{n-1}(\Psi_{n-1}) \\ B_n(\Psi_n) \end{bmatrix}}_{\mathbf{B}(\Psi, \mathbf{m})} \quad (21)$$

This is a block matrix equation and solving it is equivalent to the discrete adjoint method (Bitterlich and Knabner, 2002; Oliver and Chen, 2011). The adjoint method is widely applied in other fields, but to the best of our knowledge has not yet been applied for the Richards equation in 3D. The storage of this system and the *explicit* computation of its solution,  $\frac{\partial \Psi}{\partial \mathbf{m}}$ , are both expensive operations and not scalable to 3D problems.

Since only matrix vector products are needed for the inexact Gauss-Newton optimization method, the matrix  $\mathbf{J}$  is never needed explicitly and only the products of the form  $\mathbf{J}\mathbf{v}$  and  $\mathbf{J}^\top \mathbf{z}$  are needed for arbitrary vectors  $\mathbf{v}$  and  $\mathbf{z}$ . Projecting the full sensitivity matrix onto the data-space using  $\mathbf{P}$  results in the following equations for the Jacobian:

$$\mathbf{J} = \mathbf{P}\mathbf{A}(\Psi, \mathbf{m})^{-1}\mathbf{B}(\Psi, \mathbf{m}) \quad (22a)$$

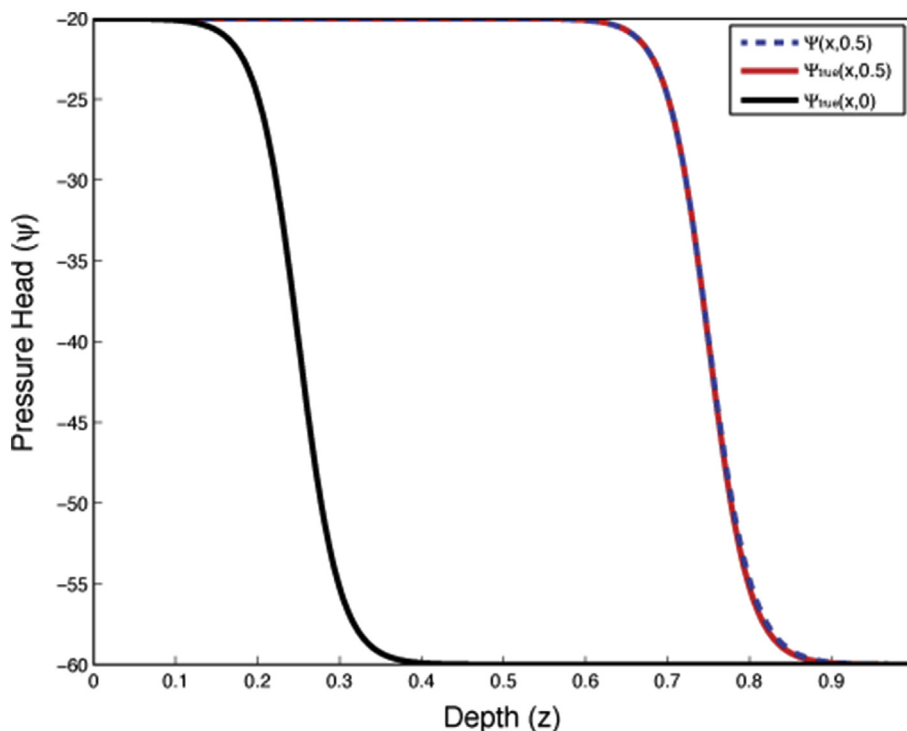


Figure 2. Fictitious source test in 1D showing the analytic function  $\Psi_{true}$  at times 0.0 and 0.5 and the numerical solution  $\Psi(x, 0.5)$  using the mixed-form Newton iteration.

$$\mathbf{J}^T = \mathbf{B}(\Psi, \mathbf{m})^T \mathbf{A}(\Psi, \mathbf{m})^{-T} \mathbf{P}^T \tag{22b}$$

In these equations, we are careful to not write  $\frac{\partial \Psi}{\partial \mathbf{m}}$ , as it is a large dense matrix which we do not want to explicitly compute or store. Additionally, the matrices  $\mathbf{A}(\Psi, \mathbf{m})$  and  $\mathbf{B}(\Psi, \mathbf{m})$  do not even need to be explicitly formed because the matrix  $\mathbf{A}(\Psi, \mathbf{m})$  is a triangular block-system, which we can solve using forward or backward substitution with only one block-row being solved at a time (this is equivalent to a single time step). To compute the matrix vector product,  $\mathbf{J}\mathbf{v}$ , we use a simple algorithm:

1. Given the vector  $\mathbf{v}$  calculate  $\mathbf{y} = \mathbf{B}\mathbf{v}$
2. Solve the linear system  $\mathbf{A}\mathbf{w} = \mathbf{y}$  for the vector  $\mathbf{w}$
3. Set  $\mathbf{J}\mathbf{v} = \mathbf{P}\mathbf{w}$

Here, we note that we complete steps (1) and (2) using a for-loop with only one block-row being computed and stored at a time. As such, only the full solution,  $\Psi$ , is stored and all other block-entries may be computed as needed. There is a complication here if data is in terms of water content or effective saturation, as the data projection is no longer linear and may have model dependence. The data projection,  $\mathbf{P}$  also changes if the temporal discretization changes, for example, when using adaptive time-stepping between different forward simulations. Both of these complications can be dealt with using the chain rule during step (3). Similarly, to compute the adjoint  $\mathbf{J}^T \mathbf{z}$  involves the intermediate solve for  $\mathbf{y}$  in  $\mathbf{A}^T \mathbf{y} = \mathbf{P}^T \mathbf{z}$  and then computation of  $\mathbf{B}^T \mathbf{y}$ . Again, we solve the block-matrix via backward substitution with all block matrix entries being computed as needed. The backward substitution algorithm can be viewed as time stepping, which means that it moves from the final time back to the initial time. This time stepping is equivalent to the adjoint method that is discussed in [Oliver and Chen \(2011\)](#) and references within. The main difference between our approach and the continuous adjoint-based method is that our approach yields the *exact* gradient of the discrete system; no such guarantee is given for adjoint-based methods.

The above algorithm and the computations of all of the different derivatives summarizes the technical details of the computations of the

Table 1

Fictitious source test for Richards equation in 1D using the mixed-form Newton iteration.

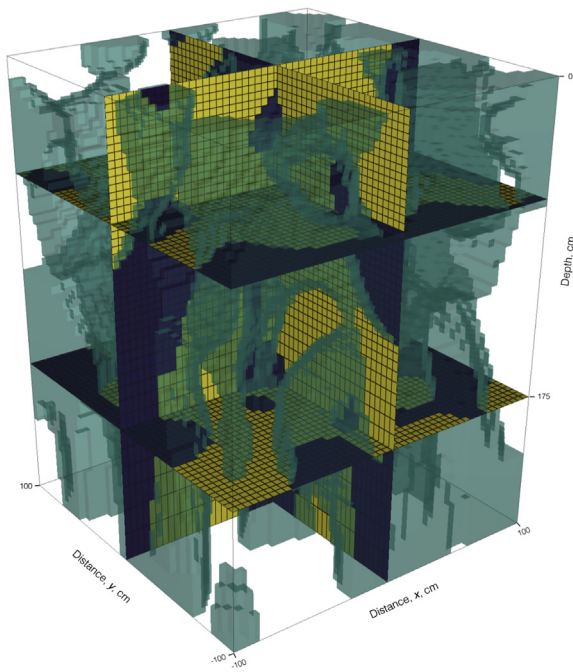
Mesh Size (n)	$\ \Psi(x, 0.5) - \Psi_{true}(x, 0.5)\ _{\infty}$	Order Decrease, $\mathcal{O}(\delta)$
64	5.485569e+00	
128	2.952912e+00	0.894
256	1.556827e+00	0.924
512	8.035072e-01	0.954
1024	4.086729e-01	0.975
2048	2.060448e-01	0.988
4096	1.034566e-01	0.994
8192	5.184507e-02	0.997

sensitivities. Equipped with this “machinery”, we now demonstrate the validity of our implementation and our ability to solve large-scale problems using modest computational resources.

## 4. Validation

### 4.1. Forward problem

The Richards equation has no analytic solution, which makes testing the code more involved. Code-to-code comparisons have been completed for comparison to [Celia et al. \(1990\)](#), which can be found in [Cockett \(2017\)](#). Here we have chosen to use a fictitious source experiment to rigorously test the code. In this experiment, we approximate an infiltration front by an arctangent function in 1D, which is centered over the highly nonlinear part of the van Genuchten curves, with  $\psi \in [-60, -20]$  centimeters. The arctangent curve advects into the soil column with time. The advantage of using an analytic function is that the derivative is known explicitly and can be calculated at all times. However, it should be noted that the Richards equation does not satisfy the analytic solution exactly, but differs by a function,  $S(x, t)$ . We refer to this function as the fictitious source term. The analytic function that we used has similar boundary conditions and shape to an example in [Celia et al. \(1990\)](#) and is considered over the domain  $x \in [0, 1]$ .



**Figure 3.** Soil structure in three dimensions showing four sections and the boundary between two soil types of sand (yellow) and loamy sand (purple). The two cross sections and the shallower depth section are shown in subsequent figures. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

$$\Psi_{\text{true}}(x, t) = -20\arctan(20((x - 0.25) - t)) - 40 \quad (23)$$

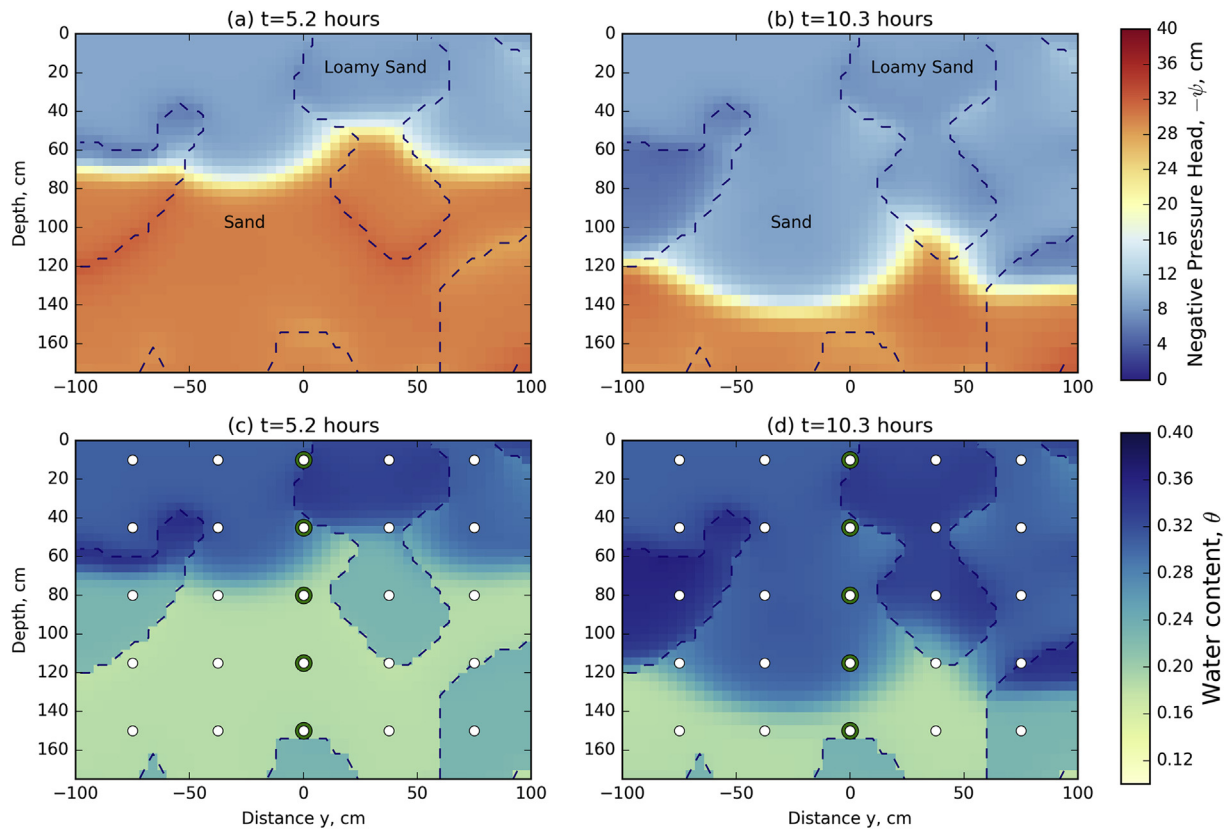
This analytic function is shown at times 0 and 0.5 in Figure 2 and has a pressure head range of  $\psi \in [-60, -20]$ . Evaluating the known pressure head field in the Richards equation (eq. (1)) we calculate the analytic derivatives and equate them to a source term,  $S(x, t)$ . Knowing this source term and the analytic boundary conditions, we can solve the discretized form of the Richards equation, which should reproduce the analytic function in equation (23). Table 1 shows the results of the fictitious source test when the number of mesh-cells is doubled and the time-discretization is both fixed and equivalent to the mesh size (i.e.  $k = h$ ). Backward Euler is a first order method used for the time discretization, as such expect first order accuracy in the solution. The final column of Table 1 indeed shows that the order of accuracy is  $\mathcal{O}(\delta)$ . The higher errors in the coarse discretization are due to high discontinuities and changes in the source term, which the coarse discretization does not resolve.

#### 4.2. Inverse problem

In order to test the implicit sensitivity calculation, we employ derivative and adjoint tests as described in Haber (2015). Given that the Taylor expansion of a function  $f(\mathbf{m} + h\Delta\mathbf{m})$  is

$$f(\mathbf{m} + h\Delta\mathbf{m}) = f(\mathbf{m}) + h\mathbf{J}\Delta\mathbf{m} + \mathcal{O}(h^2), \quad (24)$$

for any of the model parameters considered, we see that our approximation of  $f(\mathbf{m} + h\Delta\mathbf{m})$  by  $f(\mathbf{m}) + h\mathbf{J}\Delta\mathbf{m}$  should converge as  $\mathcal{O}(h^2)$  as  $h$  is reduced. This allows us to verify our calculation of  $\mathbf{J}\mathbf{v}$ . To verify the adjoint,  $\mathbf{J}^T\mathbf{v}$ , we check that



**Figure 4.** Vertical cross-sections ( $x = 0$  cm) through the pressure head and saturation fields from the numerical simulation at two times: (a) pressure head field at  $t = 5.2$  hours and (b)  $t = 10.3$  hours; and (c) saturation field at  $t = 5.2$  hours and (d)  $t = 10.3$  hours. The saturation field plots also show measurement locations and green highlighted regions that are shown in Figure 5. The true location of the two soils used are shown with a dashed outline. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)



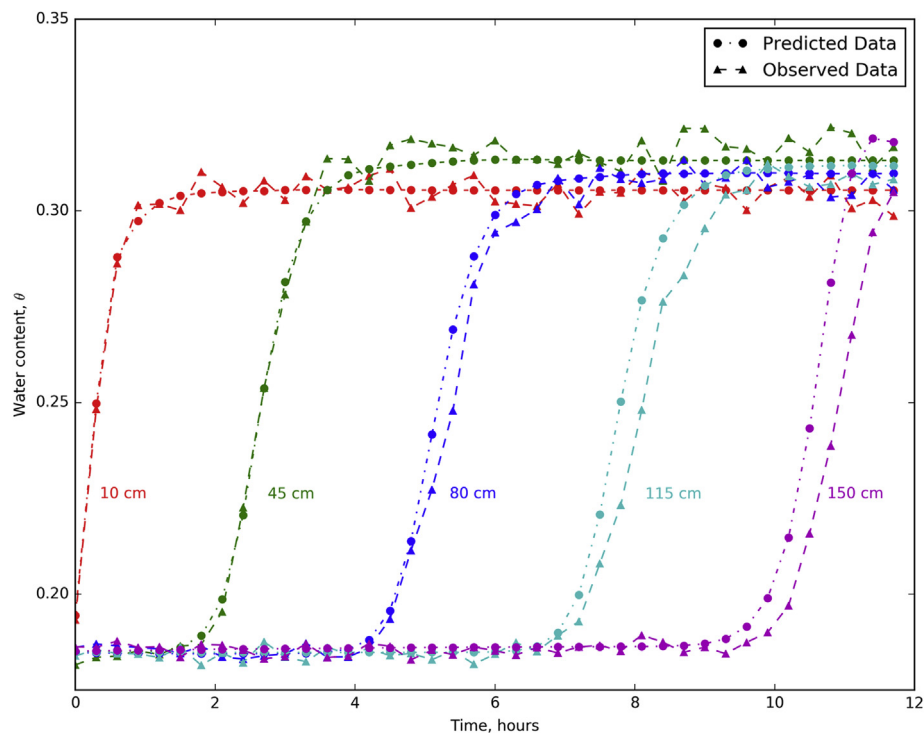


Figure 5. Observed and predicted data for five measurements locations at depths from 10 cm to 150 cm from the center of the model domain.

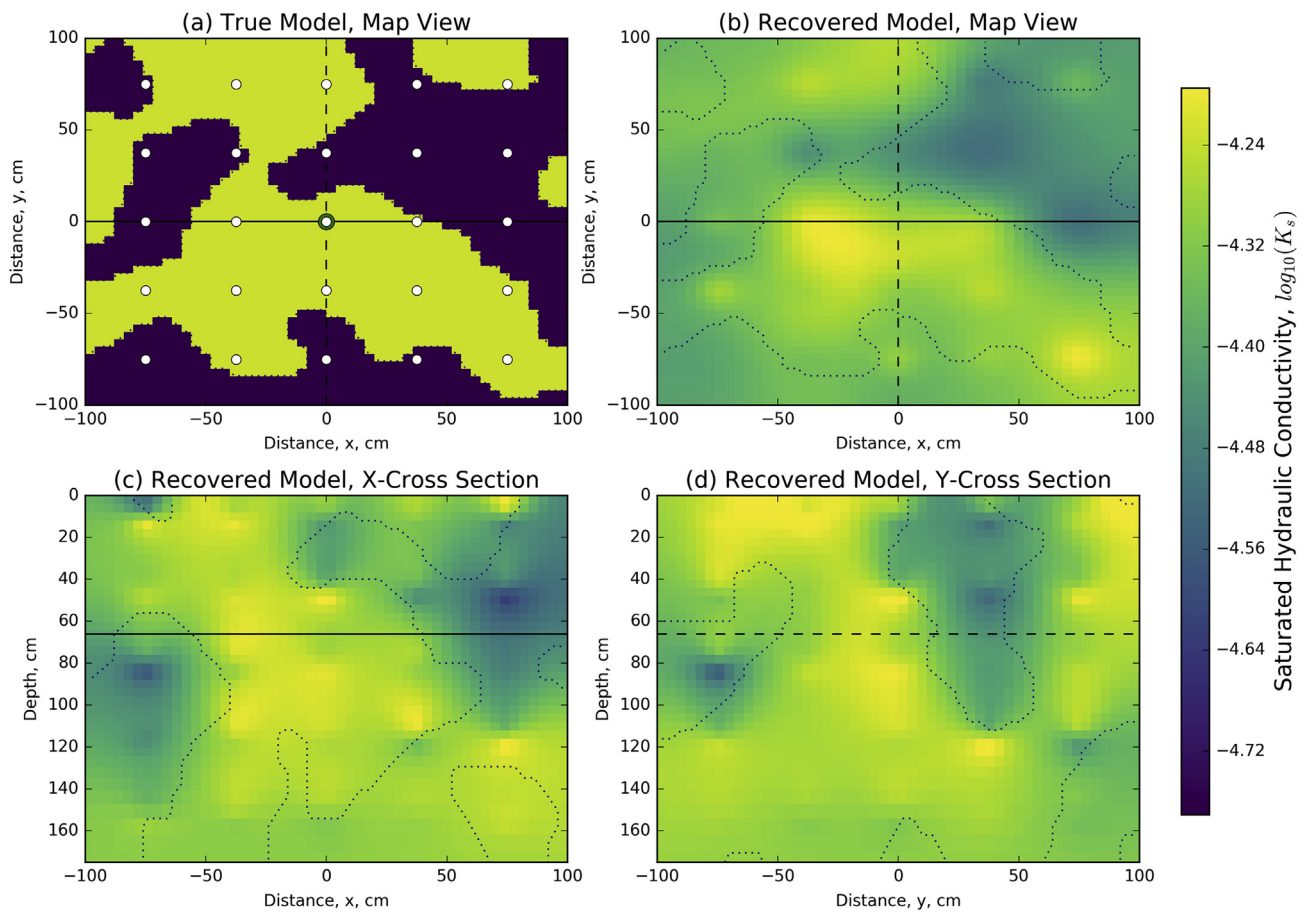


Figure 6. The 3D distributed saturated hydraulic conductivity model recovered from the inversion compared to the (a) synthetic model map view section, using (b) the same map view section, (c) an X-Z cross section and (d) a Y-Z cross-section. The synthetic model is shown as an outline on all sections, and tie lines are shown on all sections as solid and dashed lines, all location measurements are in centimeters.

$$\mathbf{w}^\top \mathbf{J} \mathbf{v} = \mathbf{v}^\top \mathbf{J}^\top \mathbf{w} \tag{25}$$

for any two random vectors,  $\mathbf{w}$  and  $\mathbf{v}$ . These tests are run for all of the parameters considered in an inversion of the Richards equation. Within our implementation, both the derivative and adjoint tests are included as unit tests which are run on any updates to the implementation (<https://travis-ci.org/simpeg/simpeg>).

### 5. Three dimensional inversion

In this section, we turn our attention to recovering a 3D soil structure given water content data. The example, motivated by a field experiment introduced in [Pidlisecky et al. \(2013\)](#), shows a time-lapse electrical resistivity tomography survey completed in the base of a managed aquifer recharge pond. The goal of this management practice is to infiltrate water into the subsurface for storage and subsequent recovery. Such projects require input from geology, hydrology, and geophysics to map the hydrostratigraphy, to collect and interpret time-lapse geophysical measurements, and to integrate all results to make predictions and decisions about fluid movement at the site. As such, the hydraulic properties of the aquifer are important to characterize, and information from hydrogeophysical investigations has been demonstrated to inform management practices ([Pidlisecky et al., 2013](#)). We use this context to motivate both the model domain setup of the following synthetic experiment and the subsequent inversion. The inversion results and script for reproducing the results can be found on FigShare ([Cockett and Haber, 2015](#)).

For the inverse problem solved here, we assume that time-lapse water-content information is available at many locations in the subsurface. In reality, water content information may only be available through proxy techniques, such as electrical resistivity methods. These proxy data can be related to hydrogeologic parameters using inversion techniques based solely on the geophysical inputs (cf. [Mawer et al. \(2013\)](#)). For the following numerical experiments, we do not address complications in empirical transformations, such as Archie’s equation ([Archie, 1942](#)). For calculation of synthetic data, the initial conditions are a dry soil with a homogeneous pressure head ( $\psi = -30$  cm). The boundary conditions applied simulate an infiltration front applied at the top of the model,  $\psi = -10$  cm  $\in \delta\Omega^{\text{top}}$ . Neumann (no-flux) boundary conditions are used on the sides of the model domain. The synthetic numerical model has a domain with dimensions 2.0 m  $\times$  2.0 m  $\times$  2.6 m for the  $x$ ,  $y$ , and  $z$  dimensions, respectively. The finest discretization used is 4 cm in each direction, after 120 cm depth we increase the cell size in the  $z$ -direction by a factor of 1.1 to distance the measurement locations from the bottom of the model domain ([Figure 4](#)). We use an exponentially expanding time discretization with 40 time steps and a total time of 12.3 h. This choice in discretization leads to a mesh with  $1.125 \times 10^5$  cells in space (50  $\times$  50  $\times$  45). To create a 3D varying soil structure, we construct a model for this domain using a 3D, uniformly random field,  $\in [0, 2]$ , that is convolved with an anisotropic smoothing kernel for a number of iterations. We create a binary distribution from this random field by splitting the values above and below unity. [Figure 3](#) shows the resulting model, which reveals potential flow paths. We then map van Genuchten parameters to this synthetic model as either a sand or a loamy-sand. The van Genuchten parameters for sand are:  $K_s$ : 5.83e-05 m/s,  $\alpha$ : 13.8,  $\theta_r$ : 0.02,  $\theta_s$ : 0.417, and  $n$ : 1.592; and for loamy-sand are:  $K_s$ : 1.69e-05 m/s,  $\alpha$ : 11.5,  $\theta_r$ : 0.035,  $\theta_s$ : 0.401, and  $n$ : 1.474.

[Figure 4a](#) and [4b](#) shows two cross-sections at time 5.2 h and 10.3 h of the pressure head field from the forward simulation. These figures show true soil type model as an outline, where the inclusions are the less hydraulically-conductive loamy sand. The pressure head field is continuous across soil type boundaries and shows the infiltration moving vertically down in the soil column. We can compute the water content field from the pressure head field using the nonlinear van Genuchten model chosen; [Figure 4c](#) and [4d](#) shows this computation at the same times. The loamy sand has a higher relative water content for the same

pressure head and the water content field is discontinuous across soil type boundaries, as expected.

The observed data, which will be used for the inversion, is collected from the water content field at the points indicated in both [Figure 4c](#) and [4d](#). The sampling location and density of this 3D grid within the model domain is similar to the resolution of a 3D electrical resistivity survey. Our implementation supports data as either water content or pressure head; however, proxy water content data is more realistic in this context. Similar to the field example in [Pidlisecky et al. \(2013\)](#), we collect water content data every 18 min over the entire simulation, leading to a total of 5000 spatially and temporally extensive measurements. The observed water content data for a single infiltration curve is plotted through depth in [Figure 5](#). The green circles in [Figure 4](#) show the locations of these water content measurements. The depth of each observation location is annotated, with the shallow measurements being first to increase in water content over the course of the infiltration experiment. To create the observed dataset,  $\mathbf{d}_{\text{obs}}$ , from the synthetic water content field, 1% Gaussian noise is added to the true water content field. This noise is below what can currently be expected from a proxy geophysical measurement of the water content. However, with the addition of more noise, we must reduce our expectations of our ability to recover the true parameter distributions from the data. In this experiment, we are interested in examining what is possible to recover under the best of circumstances, and therefore have selected a low noise level.

We parameterize these soil types using the van Genuchten empirical model (eq. (2)) with five spatially distributed properties. Inverting for all  $5.625 \times 10^5$  parameters in this simulation with only  $N_{\text{data}} = 5000$  data points is a highly underdetermined problem, and thus there are many possible models that may fit those data. In this 3D example, we will invert solely for saturated hydraulic conductivity and assume that all other van Genuchten parameters are equivalent to the sand; that is, they are parameterized to the *incorrect values* in the loamy sand. Note that this assumption, while reasonable in practice, will handicap the results, as the van Genuchten curves between these two soils differ. Better results can, of course, be obtained if we assume that the van Genuchten parameters are known; this assumption is unrealistic in practice, which means that we will not be able to recreate the data exactly. However, the distribution of saturated hydraulic conductivity may lead to insights about soil distributions in the subsurface. The inversion fell below the target misfit of 5000,  $\phi_d^* \approx N_{\text{data}}$ , at iteration twenty with  $\phi_d = 4.893 \times 10^3$ ; the target misfit was determined based on the interpretation of the data misfit as a chi-squared variable (cf. [Oldenburg and Li \(2005\)](#)). [Figure 6](#) shows the results of the inversion for saturated hydraulic conductivity as a map view slice at 66 cm depth and two vertical sections through the center of the model domain. The recovered model shows good correlation to the true distribution of hydraulic properties, which is superimposed as a dashed outline. [Figure 5](#) shows the predicted data overlaid on the true data for five water content measurement points through time; these data are from the center of the model domain. As seen in [Figure 5](#), we do a good job of fitting the majority of the data. However, there is a tendency for the predicted infiltration front to arrive before the observed data, which is especially noticeable at deeper sampling locations. This is likely

**Table 2**

Comparison of the memory necessary for storing the dense explicit sensitivity matrix compared to the peak memory used for the implicit sensitivity calculation excluding the matrix solve. The calculations are completed on a variety of mesh sizes for a single distributed parameter ( $K_s$ ) as well as for five distributed van Genuchten model parameters ( $K_s, \alpha, n, \theta_r$ , and  $\theta_s$ ). Values are reported in gigabytes (GB).

Mesh Size	Explicit Sensitivity		Implicit Sensitivity	
	1 parameter	5 parameters	1 parameter	5 parameters
32 $\times$ 32 $\times$ 32	1.31	6.55	0.136	0.171
64 $\times$ 64 $\times$ 64	10.5	52.4	0.522	0.772
128 $\times$ 128 $\times$ 128	83.9	419	3.54	4.09

because of the assumption we imposed in the inversion that all other van Genuchten parameters act as sand rather than loamy sand. In the distribution of the data residuals this is seen as a long tail, with the majority of data being fit well and the infiltration front being less resolved. To better fit the infiltration front a different weighting scheme or data misfit could be designed; opening up the inversion to other van Genuchten parameters may also improve the fit of the inversion to the infiltration front.

### 5.1. Scalability of the implicit sensitivity

For the forward simulation presented, the Newton root-finding algorithm took 4–12 iterations to converge to a tolerance of  $1 \times 10^{-4}$  m on the pressure head. The inverse problem took 20 iterations of inexact Gauss-Newton with five internal CG iterations used at each iteration. This led to a total of 222 calls to functions to solve the products  $\mathbf{J}\mathbf{v}$  and  $\mathbf{J}^T\mathbf{z}$ . For these experiments, we used a single Linux Debian Node on Google Compute Engine (Intel Sandy Bridge, 16 vCPU, 14.4 GB memory) to run the simulations and inversion. The forward problem takes approximately 40 min to solve. In this simulation, the dense Jacobian matrix would have 562.5 million elements. If we used a finite difference algorithm to explicitly calculate each of the  $1.125\text{e}5$  columns of the Jacobian with a simple forward difference, we would require a calculation for each model parameter – or approximately 8.5 years of computational time. Furthermore, we would need to recompute the Jacobian at each iteration of the optimization algorithm. In contrast, if we use the implicit sensitivity algorithm presented in this paper, we can solve the entire inverse problem in 34.5 h.

Table 2 shows the memory required to store the explicit sensitivity matrix for a number of mesh sizes and contrasts them to the memory required to multiply the implicit sensitivity by a vector. These calculations are modifications on the example presented above and use 5000 data points. The memory requirements are calculated for a single distributed parameter ( $K_s$ ) as well as five spatially distributed parameters ( $K_s$ ,  $\alpha$ ,  $n$ ,  $\theta_r$ , or  $\theta_s$ ). Neither calculation includes the memory required to solve the matrix system, as such, the reported numbers underestimate the actual memory requirements for solving the inverse problem. The aim of this comparison is to demonstrate how the memory requirements scale, an appropriate solver must also be chosen for either method to solve the forward problem. When using an explicitly calculated sensitivity matrix to invert for additional physical properties, the memory footprint increases proportionally to the number of distributed physical properties; this is not the case for the implicit sensitivity calculation. For example, on a  $128 \times 128 \times 128$  mesh, the explicit formation of the sensitivity requires 419 GB for five spatially distributed model parameters, which is five times the requirement for a single distributed model parameter (83.9 GB). For the implicit sensitivity on the same mesh, only 4.09 GB of memory is required, which is 1.2 times the requirement for a single distributed model parameter (3.54 GB). For this mesh, inverting for five spatially distributed parameters requires over 100 times less memory when using the implicit sensitivity algorithm, allowing these calculations to be run on modest computational resources.

## 6. Conclusions

The number of parameters that are estimated in inversions using the Richards equation has grown and will continue to grow as time-lapse data and geophysical data integration become standard in hydrogeological site characterizations. The increase in data quantity and quality provides the opportunity to estimate spatially distributed hydraulic parameters from the Richards equation; doing so requires efficient simulation and inversion strategies. In this paper, we have shown a computationally efficient derivative-based optimization algorithm that does not store the Jacobian, but rather computes its effect on a vector (i.e.  $\mathbf{J}\mathbf{v}$  or  $\mathbf{J}^T\mathbf{z}$ ). By not storing the Jacobian, the size of the problem that we

can invert becomes much larger. We have presented efficient methods to compute the Jacobian that can be used for any empirical hydraulic parameters, even if the functional relationship between parameters is obtained from laboratory experiments.

Our technique allows a deterministic inversion, which includes regularization, to be formulated and solved for any of the empirical parameters within the Richards equation. For a full 3D simulation, as many as ten spatially distributed parameters may be needed, resulting in a highly non-unique problem. As such, the techniques presented in this paper will likely have to be used in conjunction with similar advances in regularization, optimization and joint inversion techniques. Depending on the setting, amount of a-priori knowledge, quality and quantity of data, the selection of which parameters to invert for may vary. Our methodology enables practitioners to experiment in 1D, 2D and 3D with full simulations and inversions, in order to explore the parameters that are important to a particular dataset. We have provided a numerical implementation as an open-source library to encourage experimentation, application and extension of these ideas.

## Acknowledgments

The funding for this work is provided through the Vanier Canada Graduate Scholarships Program and grants from The University of British Columbia (NSERC 22R47082). Thank you to the current and future contributors of the SimPEG project (<http://simpeg.xyz>) and to Ole Klein and two anonymous reviewers for improving the quality of this manuscript.

## References

- Archie, G., 1942. The electrical resistivity log as an aid in determining some reservoir characteristics. *Transactions of the AIME* 146 (1), 54–62.
- Armijo, L., 1966. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics* 16 (1), 1–3.
- Ascher, U.M., 2008. Numerical methods for evolutionary differential equations. Society for Industrial and Applied Mathematics. <https://books.google.ca/books?id=nCrV8NcttWc-&pg=gsbs-similarbooks>.
- Ascher, U.M., Greif, C., Jun 2011. A First Course in Numerical Methods. Society for Industrial and Applied Mathematics, Philadelphia, PA. <http://epubs.siam.org/doi/book/10.1137/9780898719987>.
- Aster, R.C., Borchers, B., Thurber, C., 2004. *Parameter Estimation and Inverse Problems*. Elsevier Inc.
- Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., der Vorst, H.V., 1994. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia.
- Binley, A., Cassiani, G., Middleton, R., Winship, P., 2002. Vadose zone flow model parameterisation using cross-borehole radar and resistivity imaging. *Journal of Hydrology* 267 (3), 147–159.
- Bitterlich, S., Durner, W., Iden, S.C., Knabner, P., Aug 2004. Inverse estimation of the unsaturated soil hydraulic properties from column outflow experiments using free-form parameterizations. *Vadose Zone Journal* 3 (3), 971–981. <http://vzj.scijournals.org/cgi/doi/10.2113/3.3.971>.
- Bitterlich, S., Knabner, P., 2002. An efficient method for solving an inverse problem for the Richards equation. *Journal of Computational and Applied Mathematics* 147, 153–173.
- Brooks, R.H., Corey, A.T., 1964. Hydraulic properties of porous media and their relation to drainage design. *Trans. ASAE* 7.1 26 (28).
- Bui-Thanh, T., Ghattas, O., Jan 2015. A scalable algorithm for MAP estimators in Bayesian inverse problems with besov priors. *Inverse Problems and Imaging* 9 (1), 27–53. <http://www.aims-science.org/journals/displayArticlesnew.jsp?paperID=10739>.
- Carrick, S., Almond, P., Buchan, G., Smith, N., Dec 2010. In situ characterization of hydraulic conductivities of individual soil profile layers during infiltration over long time periods. *European Journal of Soil Science* 61 (6), 1056–1069. <http://doi.wiley.com/10.1111/j.1365-2389.2010.01271.x>.
- Celia, M.A., Bouloutas, E.T., Zarba, R.L., 1990. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research* 26 (7), 1483–1496. <http://www.agu.org/pubs/crossref/1990/QWR026i007p01483.shtml>.
- Cockett, R., 2017. Richards Equation Comparison to Celia 1990. FigShare. <https://doi.org/10.6084/m9.figshare.4833605.v1>.
- Cockett, R., Haber, E., 2015. Richards Equation Inversion. FigShare. <https://doi.org/10.6084/m9.figshare.1450842>.
- Cockett, R., Heagy, L.J., Oldenburg, D.W., 2016. Pixels and their neighbors: finite volume. *The Leading Edge* 35 (8), 703–706.
- Cockett, R., Kang, S., Heagy, L.J., Pidlisecky, A., Oldenburg, D.W., 2015. SimPEG: an open source framework for simulation and gradient based parameter estimation in geophysical applications. *Computers and Geosciences* 85, 142–154. <http://www.sciencedirect.com/science/article/pii/S009830041530056X>.

- Constable, S.C., Parker, R.L., Constable, C.G., 1987. Occam's inversion: a practical algorithm for generating smooth models from electromagnetic sounding data. *Geophysics* 52 (3), 289–300. <http://link.aip.org/link/?GPY/52/289/1>.
- Dean, O.S., Reynolds, A.C., Liu, N., 2008. *Inverse Theory for Petroleum Reservoir Characterization and History Matching*. Cambridge University Press, Cambridge.
- Deiana, R., Cassiani, G., Kemna, A., Villa, A., Bruno, V., Bagliani, A., 2007. An experiment of non-invasive characterization of the vadose zone via water injection and cross-hole time-lapse geophysical monitoring. *Near Surface Geophysics* 183–194.
- Doherty, J.E., 2015. *Calibration and Uncertainty Analysis for Complex Environmental Models*. Watermark Numerical Computing, Brisbane, Australia.
- Durner, W., 1994. Hydraulic conductivity estimation for soils with heterogeneous pore structure. *Water Resources Research* 30 (2), 211. <http://www.agu.org/pubs/crossref/1994/93WR02676.shtml>.
- Finsterle, S., Kowalsky, M.B., Jun 2011. A truncated Levenberg Marquardt algorithm for the calibration of highly parameterized nonlinear models. *Computers and Geosciences* 37 (6), 731–738. <http://linkinghub.elsevier.com/retrieve/pii/S0098300410003729>.
- Finsterle, S., Zhang, Y., Jul 2011. Solving iTOUGH2 simulation and optimization problems using the PEST protocol. *Environmental Modelling and Software* 26 (7), 959–968. <http://linkinghub.elsevier.com/retrieve/pii/S1364815211000521>.
- Fletcher, C.A.J., 1988. *Computational Techniques for Fluid Dynamics, II*. Springer-Verlag.
- Gunzburger, M.D., 2003. *Perspectives in Flow Control and Optimization*. SIAM.
- Haber, E., 2005. Quasi-Newton methods for large scale electromagnetic inverse problems. *Inverse Problems* 21.
- Haber, E., 2015. Computational methods in geophysical electromagnetics. *Mathematics in Industry*. <http://books.google.ca/books?id=favjoQEACAAJ>.
- Haber, E., Ascher, U., Oldenburg, D., 2000. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems* 16, 1263–1280.
- Haber, E., Ascher, U., Oldenburg, D., 2004. Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach. *Geophysics* 69, 1216–1228.
- Haber, E., Ascher, U.M., Jan 2001. Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients. *SIAM Journal on Scientific Computing* 22 (6), 1943–1961. <http://epubs.siam.org/doi/abs/10.1137/S1064827599360741>.
- Haber, E., Gazit, M.H., 2013. Model fusion and joint inversion. *Reviews in Geophysics to Appear* 35–54.
- Haber, E., Heldmann, S., Ascher, U., 2007. Adaptive finite volume method for the solution of discontinuous parameter estimation problems. *Inverse Problems* 13, 63–67.
- Hinnell, A.C., Ferré, T.P.a., Vrugt, J.a., Huisman, J.a., Moysey, S., Rings, J., Kowalsky, M.B., Apr 2010. Improved extraction of hydrologic information from geophysical data through coupled hydrogeophysical inversion. *Water Resources Research* 46 (4). W00D40. <http://www.agu.org/pubs/crossref/2010/2008WR007060.shtml>. <http://doi.wiley.com/10.1029/2008WR007060>.
- Iden, S.C., Durner, W., Jul 2007. Free-form estimation of the unsaturated soil hydraulic properties by inverse modeling using global optimization. *Water Resources Research* 43 (7). W07451. <http://www.agu.org/pubs/crossref/2007/2006WR005845.shtml>.
- Kaipio, J., Somersalo, E., 2004. *Statistical and Computational Inverse Problems*. Springer Verlag.
- Klein, O., Cirpka, O.A., Bastian, P., Ippisch, O., 2017. Efficient geostatistical inversion of transient groundwater flow using preconditioned nonlinear conjugate gradients. *Advances in Water Resources* 102, 161–177. <https://doi.org/10.1016/j.advwatres.2016.12.006>.
- Liang, W.-L., Uchida, T., Mar 2014. Effects of topography and soil depth on saturated-zone dynamics in steep hillslopes explored using the three-dimensional Richards' equation. *Journal of Hydrology* 510, 124–136. <http://linkinghub.elsevier.com/retrieve/pii/S0022169413009268>.
- Linde, N., Doetsch, J., 2016. Joint Inversion in Hydrogeophysics and Near-Surface Geophysics, pp. 117–135. <https://doi.org/10.1002/9781118929063.ch7>.
- Linde, N., Renard, P., Mukerji, T., Caers, J., 2015. Geological realism in hydrogeological and geophysical inverse modeling: a review. *Advances in Water Resources* 86, 86–101. <https://doi.org/10.1016/j.advwatres.2015.09.019>.
- Lipnikov, C., Misitas, O., 2013. Second-order accurate monotone finite volume scheme for Richards' equation. *Journal of Computational Physics* 239, 123–137.
- Liu, Y., Pau, G.S.H., Finsterle, S., 2017. Implicit sampling combined with reduced order modeling for the inversion of vadose zone hydrological data. *Computers and Geosciences* 108 (May 2016), 21–32. <https://doi.org/10.1016/j.cageo.2017.04.001>.
- Mawer, C., Kitanidis, P., Pidlisecky, A., Knight, R., 2013. Electrical resistivity for characterization and infiltration monitoring beneath a managed aquifer recharge pond. *Vadose Zone Journal* 12 (1), 1–20.
- Mualem, Y., 1976. A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research* 12 (3), 513. <http://www.agu.org/pubs/crossref/1976/QWR012i003p00513.shtml>.
- Nocedal, J., Wright, S.J., 1999. *Numerical Optimization*. Springer, New York, NY.
- Oldenburg, D.W., Li, Y., 2005. Inversion for applied geophysics: a Tutorial. In: *Near-surface Geophysics*, pp. 89–150. No. 13. Ch. 5. <http://library.seg.org/doi/abs/10.1190/1.9781560801719.ch5>.
- Oliver, D.S., Chen, Y., Jan 2011. Recent progress on reservoir history matching: a review. *Computational Geosciences* 15 (1), 185–221. <http://link.springer.com/10.1007/s10596-010-9194-2>.
- Oliver, Y.A.D., Reynolds, A., 2001. Efficient history-matching using subspace vectors. *Computational Geosciences* 5, 151–172.
- Orgogozo, L., Renon, N., Soulaine, C., Hénon, F., Tomer, S., Labat, D., Pokrovsky, O., Sekhar, M., Ababou, R., Quintard, M., 2014. An open source massively parallel solver for Richards equation: mechanistic modelling of water fluxes at the watershed scale. *Computer Physics Communications* 185 (12), 3358–3371. <http://linkinghub.elsevier.com/retrieve/pii/S0010465514002719>.
- Pidlisecky, A., Cockett, a.R., Knight, R., 2013. Electrical conductivity probes for studying vadose zone processes: advances in data acquisition and analysis. *Vadose Zone Journal* 12 (1), 1–12. <https://www.soils.org/publications/vzj/abstracts/12/1/O1vzj2012.0073>.
- Richards, L.A., 1931. Capillary conduction of liquids through porous mediums. *Journal of Applied Physics* 1 (5), 318–333.
- Saad, Y., 1996. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company.
- Sarma, P., Durlinsky, L.J., Aziz, K., Chen, W., 2007. A new approach to automatic history matching using kernel PCA. In: *SPE Reservoir Simulation Symposium*, Houston, Texas.
- Šimunek, J., Senja, M., 2012. Hydrus: model use, calibration, and validation. *American Society of Agricultural and Biological Engineers* 55 (4), 1261–1274.
- Šimunek, J., van Genuchten, M.T., 1996. Estimating unsaturated soil hydraulic properties from tension disc infiltrometer data by numerical inversion. *Water Resources Research* 32 (9), 2683. <http://www.agu.org/pubs/crossref/1996/96WR01525.shtml>.
- Tikhonov, A., Arsenin, V., 1977. *Solutions of Ill-posed Problems*. W.H. Winston and Sons.
- Towara, M., Schanen, M., Naumann, U., 2015. Mpi-Parallel discrete adjoint OpenFOAM. *Procedia Computer Science* 51 (1), 19–28. <https://doi.org/10.1016/j.procs.2015.05.181>.
- Trottenberg, U., Oosterlee, C., Schuller, A., 2001. *Multigrid*. Academic Press.
- van Genuchten, M.T., 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal* 44 (5), 892. <https://www.soils.org/publications/sssaj/abstracts/44/5/QSS0440050892>.
- Vogel, C., 2001. *Computational Methods for Inverse Problem*. SIAM, Philadelphia.