Adaptive regularization for the Richards equation

François Févotte, Ari Rappaport, Martin Vohralík

Adaptive regularization for the Richards equation

François Févotte[‡] Ari Rappaport[*†] Martin Vohralík[*†]

July 23, 2024

Abstract

The Richards equation is ubiquitous in the modelling of flows in porous media. It serves as a model in its own right, but also as a stepping stone to more complex models of multiphase flows. Despite its relative simplicity, it features many challenges from a computational point of view due to the nonsmooth and degenerate nature of the nonlinear state functions. In this paper, we replace these functions with regularized (smooth and nondegenerate) counterparts where the amount of added regularization is controlled by a single regularization parameter. We introduce a set of a simple a posteriori error estimators that we use to adaptively steer the regularization and linearization. In particular, we stop the iterative linearization when the corresponding estimator is dominated by the regularization estimator and adaptively choose the regularization parameter so that the regularization estimator does not dominate the discretization one. The full adaptive algorithm is tested on a suite of numerical examples adapted from recent works on improving the robustness of solvers for the Richards equation and on benchmark cases.

Key words: the Richards equation, adaptivity, regularization, Newton's method

1 Introduction

One of the most fundamental equations for modeling flows in porous media is the Richards equation. It can be viewed as a simplified two-phase model, e.g., for water and air, where one of the phases is assumed to be of constant pressure. For a detailed review of the role of the Richards equation in porous media modeling see, e.g., [11, 12]. The equation describes the evolution in space and in time of the pressure p and saturation s for a fluid in a porous medium. Given a domain $\Omega \subset \mathbb{R}^d$, for $d = 1, 2, 3$, and a final time $T > 0$, the Richards equation is given by

$$\phi \partial_t s - \nabla \cdot [\mathbf{K} \kappa(s)(\nabla p + \boldsymbol{g})] = f, \quad \text{in } \Omega \times (0, T) \tag{1}$$

where the constant vector $-\boldsymbol{g}$ represents the force of gravity, $\mathbf{K} : \Omega \to \mathbb{R}^{d \times d}$ is an absolute permeability tensor, $\kappa : [0, 1] \to [0, 1]$ is a relative permeability function, $f : \Omega \times [0, T] \to \mathbb{R}$ a source term, and ϕ is the porosity, all considered as data. Suitable initial and boundary conditions need to be added. The system is closed by an algebraic relationship expressing the saturation as a function of the pressure, i.e., there exists a function $S : \mathbb{R} \to [0, 1]$ such that

$$s = S(p). \tag{2}$$

For the well-posedness of this initial boundary value problem see, e.g., [1].

Realistic choices for the saturation function S of (2) and relative permeability κ are nonlinear, cf. Figures 1 and 2. This means that once the equation (1) is discretized with an implicit timestep cutting scheme, a linearization procedure must be applied at each timestep. Moreover, these functions are typically nonsmooth and degenerate, which is the central bottleneck.

The design of robust and efficient linearization schemes for the Richards equation is an active area of research. Newton's method [27, 28, 14] is an attractive choice due to its potentially quadratic convergence. A sufficient condition for the convergence of Newton's method in the context of the Richards equation was derived in [20]. In particular, the authors of [20] considered the lowest order continuous Galerkin finite element method (FEM) as a spatial discretization and an implicit Euler time discretization and derived a condition of the form

$$\tau < C S_{\mathrm{m}}^{\frac{2+r}{r}} h^d, \tag{3}$$

[*]Inria, 2 rue Simone Iff, 75589 Paris, France.
[†]Université Paris-Est, CERMICS (ENPC), 77455 Marne-la-Vallée, France.
[‡]Triscale innov, 7 rue de la Croix Martre, 91120 Palaiseau, France.

where τ is the timestep size, h is the mesh size, $S_{\mathrm{m}} := \inf S' \geq 0$, and $C, r > 0$ depend on the functions S and κ. In practice, satisfying this condition may render the timestep τ prohibitively small, or the condition may even be impossible to satisfy if the derivative $S' = 0$ (elliptic degeneracy). Other linearization schemes include the modified Picard method [10], L-schemes [36, 28, 31, 29], and the Jäger–Kačur method [21, 22]. These methods are generally more robust than Newton's method at the cost of slower convergence. In particular, the L-scheme was shown to be unconditionally convergent in [35], though it only converges linearly. The tradeoff between the robustness of the linearly-converging methods and the speed of Newton's method have motivated hybrid methods such as those studied in [28] where the authors apply several iterations of the slower scheme to provide an initial guess to Newton. More recently, this strategy was taken further in [37], where an a posteriori error estimator was designed to provide a criterion to switch between the L-scheme and Newton's method.

The degenerate nature of the Richards equation (see §3) partially explains the difficulty encountered by Newton's method. One way to address the degeneracy involves the choice of the unknown in (1). In particular, whenever the function S in (2) is invertible, one has the choice of whether to solve for the pressure p or the saturation s in (1). This idea led to the so-called primary variable switching methods [19, 15]. Initially, these methods required a local choice (by looping over degrees of freedom in the context of a Galerkin method) for which variable to solve for. This idea was elegantly adapted in [7], where the authors achieved a continuous variable switch by introducing a global C^1 parameterization of the saturation curve (2). The parameter is chosen so that it is proportional to the saturation in the dry regions (where $s \ll 1$) and otherwise it is proportional to the pressure. The continuous variable switching was recently generalized in [5] as well as in the PhD thesis [3] where the authors consider an additional switch that aids in the case of heterogeneous media.

On top of difficulties related to degeneracy, the specific forms of the nonlinearities for the most common models (Brooks–Corey and van Genuchten–Mualem, see §2) also suffer from low regularity. Approaches to address this while keeping good convergence of Newton's method include the line search or trust region methods, where the step size of Newton is limited in certain critical zones of the nonlinearity, see, e.g., [23, 41, 4] and the references therein. Another alternative to handle low regularity is the so-called semismooth Newton method [25, 34], where the main tenant is to work with elements of the subdifferential to a nonsmooth function.

A useful tool in the context of the Richards equation, both theoretically and practically, is regularization, i.e., considering an auxiliary perturbed problem to obtain some desired properties. The authors of [32] rely on regularization to prove well-posedness of a certain case of the Richards equation. Regularization has already been explored for improving the performance of iterative schemes in, for example, [22, 4]. In [22], regularized versions of the nonlinear functions are introduced to control the degeneracy whereas in [4], a kind of slope limiting method was proposed to handle the case where the derivative of the relative permeability κ tends to infinity. However, in practice, a natural question to ask is how much regularization should be added to obtain a tradeoff between model error and performance.

In this work, we seek to provide a possible answer to the above questions by introducing regularization and adaptively updating the regularization parameter. In particular, the adaptive choice is steered by a posteriori error estimators. Our a posteriori estimators follow the spirit of those derived in [30] in the context of the fully degenerate the Richards equation, where a rigorous a posteriori analysis leads to a reliable and efficient estimator.

We also take inspiration from our recent work [18], where we introduced an adaptive algorithm for regularizing a nonsmooth nonlinearity based on an additive decomposition of an estimator. The central observation is that *regularization (model) error is often dominated by the discretization error and hence does not impact the overall accuracy of the scheme*. In the setting of [18], we were able to recover the optimal rate of convergence with respect to total degrees of freedom by solving a sequence of regularized problems without ever sending the regularization parameter to 0. We seek to apply this same strategy to this setting, where component estimators will guide an adaptive algorithm and in particular ensure that the regularization component estimator remains sufficiently below that of discretization. Iterative linearization error is then made subordinate to the regularization one.

Our scheme has several advantages to those already mentioned. Firstly, it does not require the modification of the underlying linearization solver and allows for the Newton one. For the test cases we consider, the regularization allows Newton to converge where it takes hundreds of iterations or does not converge without, or requires substantial timestep cuts. Finally, by adaptively lowering the level of regularization, we are able to produce a solution that matches well with a solution obtained without regularization.

The rest of the paper is organized as follows. In §2, we introduce the necessary notation as well as the

assumptions on the data. In §3, we discuss the various difficulties for a nonlinear solver and introduce our proposed regularization. In §4, we present the backward Euler–finite element discretization of the Richards equation with its corresponding regularized and linearized problems. In §5 we detail our adaptive algorithm. In §6, we present numerical experiments and we we present our conclusions and outlook in §7.

2 Setting and specification of the data

In this section we detail the necessary information to describe the problem under consideration. We use the standard notation from functional analysis. Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$ be a domain with a Lipschitz polygonal boundary. For $\omega \subseteq \Omega$, let $(\cdot,\cdot)_\omega$ and $\|\cdot\|_\omega$ correspond to the $L^2(\omega)$ inner product and norm, respectively. We drop the subscripts when $\omega = \Omega$. Let $H^1(\Omega)$ be the Sobolev space of functions defined on Ω with first-order weak derivatives in $L^2(\Omega)$. We also introduce the space $\boldsymbol{H}(\mathrm{div}, \Omega) := \{\boldsymbol{v} \in [L^2(\Omega)]^d : \nabla\cdot\boldsymbol{v} \in L^2(\Omega)\}$.

We now specify the initial and boundary conditions for the problem (1). We consider a partition of the boundary $\partial\Omega = \Gamma_\mathrm{D} \cup \Gamma_\mathrm{N}$ into Dirichlet and Neumann boundaries, where Γ_D has strictly positive $(d-1)$-dimensional measure. The boundary conditions are specified as

$$p = p_\mathrm{D} \qquad \text{on } \Gamma_\mathrm{D} \times (0, T], \tag{4a}$$

$$\mathbf{K}\kappa(s)(\nabla p + \boldsymbol{g}) \cdot \boldsymbol{n} = q_\mathrm{N} \qquad \text{on } \Gamma_\mathrm{N} \times (0, T], \tag{4b}$$

and the corresponding set incorporating the Dirichlet boundary condition is given by

$$H^1_\mathrm{D}(\Omega) := \{v \in H^1(\Omega) : v|_{\Gamma_\mathrm{D}} = p_\mathrm{D}\}. \tag{5}$$

The initial condition is imposed on the saturation,

$$s(\boldsymbol{x}, 0) = s_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega. \tag{6}$$

We introduce a conforming triangulation $\mathcal{T}_h$ of Ω, i.e., $\mathcal{T}_h = \cup_K K$ where the intersection of two simplices $K, K' \in \mathcal{T}_h$ is either empty or an l-dimensional simplex for $0 \le l \le d-1$. We consider uniform timestep cutting with N timesteps so that the interval $(0, T)$ is partitioned with fixed step size $\tau := T/N$ and the time points $\{t_n\}_{n=0}^N$ are given by $t_n = n\tau$ for all $n = 0, \ldots, N$.

We consider the following assumptions on the data of problem (1)–(6).

Assumption 2.1 (Assumptions on the data)**.** *The following holds for a given mesh $\mathcal{T}_h$ and time points $\{t_n\}_{n=0}^N$:*

(A1) The absolute permeability tensor $\mathbf{K} : \Omega \to \mathbb{R}^{d\times d}$ is piecewise constant with respect to $\mathcal{T}_h$ and satisfies the ellipticity and boundedness conditions, i.e., there exist constants $K_\mathrm{m}, K_\mathrm{M} > 0$ such that, for almost every $\boldsymbol{x} \in \Omega$ and for all $\boldsymbol{\xi} \in \mathbb{R}^d$, there holds

$$K_\mathrm{m}|\boldsymbol{\xi}|^2 \le \boldsymbol{\xi}^T\mathbf{K}(\boldsymbol{x})\boldsymbol{\xi} \le K_\mathrm{M}|\boldsymbol{\xi}|^2.$$

(A2) The boundary condition p_D is piecewise affine on $\mathcal{T}_h$ and continuous in space and constant in time.

(A3) The initial saturation s_0 is piecewise constant on $\mathcal{T}_h$.

(A4) The source function f is piecewise constant on $\mathcal{T}_h$ in space and piecewise constant in time.

(A5) The porosity $\phi : \Omega \to (0, 1]$ is piecewise constant on $\mathcal{T}_h$ and constant in time.

We work here with the *effective* saturation given by

$$\mathscr{S}(s) = \frac{s - S_\mathrm{R}}{S_\mathrm{V} - S_\mathrm{R}}, \tag{7}$$

where S_V is the maximum saturation, and S_R is the residual saturation. Unless otherwise stated, we assume $S_\mathrm{V} = 1$ and $S_\mathrm{R} = 0$.

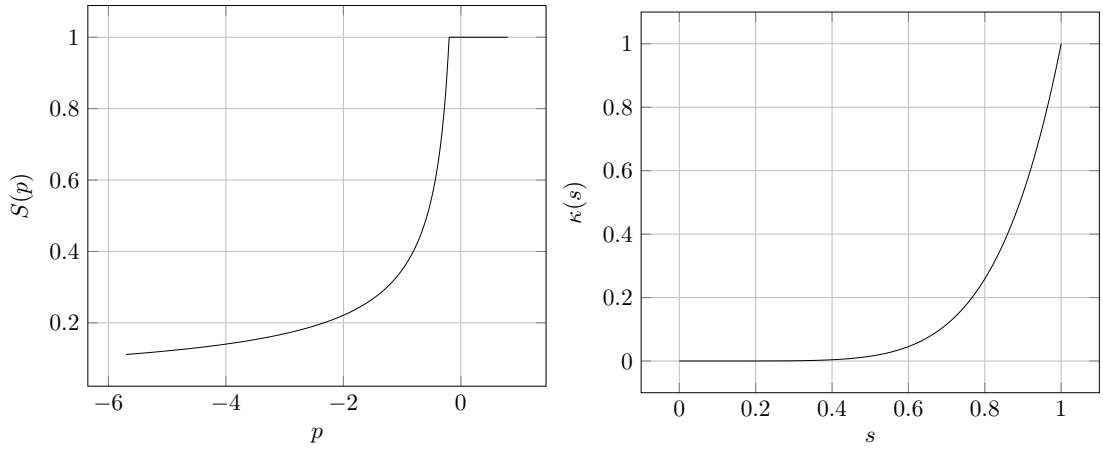

Figure 1: $[p_{\mathrm{M}} = -0.2, \lambda_1 = 0.66]$ Saturation and relative permeability functions for the Brooks–Corey constitutive law (8).

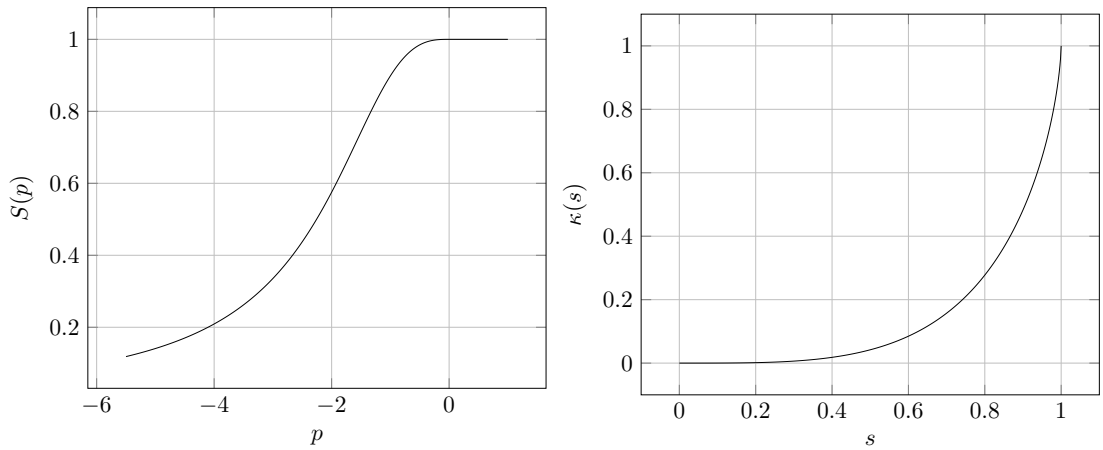

Figure 2: $[p_{\mathrm{M}} = 0, \lambda_2 = 0.66, S_{\mathrm{R}} = 0, S_{\mathrm{V}} = 1, \alpha = 0.551,\ \kappa_{\mathrm{c}} = 1]$ Saturation and relative permeability functions for the van Genuchten–Mualem constitutive law (9).

We now introduce the two most common constitutive laws for the nonlinear functions S and κ in (1)–(2). We consider the Brooks–Corey model [9]

$$\kappa(s) = \mathscr{S}(s)^{\frac{2+3\lambda_1}{\lambda_1}} \tag{8a}$$

$$S(p) = \begin{cases} (-p/p_{\mathrm{M}})^{-\lambda_1} & p \le p_{\mathrm{M}}, \\ 1 & \text{otherwise}, \end{cases} \tag{8b}$$

for parameters $p_{\mathrm{M}} < 0, \lambda_1 \in (0,1)$, as well as the van Genuchten–Mualem model [38], for $p_{\mathrm{M}} \in \mathbb{R}$,

$$\kappa(s) = \kappa_{\mathrm{c}} \sqrt{\mathscr{S}(s)} (1 - (1 - \mathscr{S}(s)^{1/\lambda_2})^{\lambda_2})^2, \tag{9a}$$

$$S(p) = \begin{cases} \left[(1 + (-\alpha p)^{\frac{1}{1-\lambda_2}} \right]^{-\lambda_2} & p \le p_{\mathrm{M}}, \\ 1 & \text{otherwise}. \end{cases} \tag{9b}$$

3 Difficulties related to the nonlinearities and proposed regularization

We first outline the possible difficulties a nonlinear solver can encounter in the context of the Richards equation. We summarize them in the following list:

1. Hyperbolic degeneracy: if $\kappa = 0$, the terms containing spatial derivatives vanish and the PDE changes type from parabolic to an ODE.

2. Elliptic degeneracy: If $S' = 0$, the PDE changes type from parabolic to elliptic. This is typically not a serious problem for solvers in the pressure formulation that we have chosen.

3. For the van Genuchten–Mualem model (9), the derivative of the relative permeability function blows up, i.e., $\kappa'(s) \to \infty$ as $s \to S_{\mathrm{R}}$ and consequently, $\mathscr{S}(s) \to 1$.

4. For the Brooks–Corey model (8), the saturation function $S(p)$ is non-differentiable at $p = p_{\mathrm{M}}$.

We thus replace the saturation function S and the relative permeability function κ by their regularized counterparts denoted by S_ϵ and κ_ϵ, respectively. The parameter $\epsilon > 0$ determines the amount of added regularization, and is adaptively updated to balance the incurred model error (see §5) with the discretization error. The regularization is designed with the intention of alleviating the problems mentioned in the previous list. We now state our conditions on the regularization.

Assumption 3.1 (Assumptions on the regularization)**.** *The approximations S_ϵ to the saturation function S and κ_ϵ to the relative permeability κ satisfy the following:*

(A1) The regularized relative permeability function satisfies $\lim_{\epsilon\to 0} \kappa_\epsilon(s) = \kappa(s)$ *for all* $s \in [0, 1]$.

(A2) The regularized saturation function satisfies $\lim_{\epsilon\to 0} S_\epsilon(p) = S(p)$ *for all* $p \in \mathbb{R}$.

(A3) For any $\epsilon > 0$*, there exists a constant* $\underline{\kappa}_\epsilon > 0$ *such that* $\kappa_\epsilon(s) > \underline{\kappa}_\epsilon$ *for all* $s \in [0, 1]$.

(A4) The regularized saturation function satisfies $S_\epsilon \in C^1(\mathbb{R})$

(A5) The regularized composite function satisfies $\kappa_\epsilon \circ S_\epsilon \in C^1(\mathbb{R})$.

Assumptions (A1) and (A2) ensure that the regularized functions are good pointwise approximations of the true functions S and κ. The Assumption (A3), ensures that the regularized function κ_ϵ does not induce a hyperbolic degeneracy. The last two assumptions are smoothness requirements that appear whenever the derivatives of S_ϵ and $\kappa_\epsilon \circ S_\epsilon$ are employed in the nonlinear solver, as we will see below in §4.3.

We now introduce our choices of regularization which satisfy Assumption 3.1. The choices depend on the chosen models, namely the Brooks–Corey model (8) and the van Genuchten–Mualem model (9). First, in the case of the Brooks–Corey model, the regularization of the relative permeability is simply

$$\kappa_\epsilon(s) := \kappa(s) + \epsilon. \tag{10}$$

This ensures that $\kappa_\epsilon \geq \epsilon > 0$ for all $s \in [0, 1]$. The regularized saturation for the Brooks–Corey model is given by

$$S_\epsilon(p) = \begin{cases} S_\epsilon^k(p) & \text{if } |p - p_{\mathrm{M}}| < \epsilon, \\ S(p) & \text{otherwise,} \end{cases} \tag{11}$$

where $S_\epsilon^k(p)$ is determined by polynomial interpolation so that $S_\epsilon \in C^k(\mathbb{R})$ is k-times continuously differentiable. In particular we choose $k = 2$, over satisfying Assumption (A4) but advantageous to the Newton linearization. A plot is given in Figure 3 for several values of ϵ. For the van Genuchten–Mualem model, to satisfy Assumption (A5) we follow the approach in [4] where the relative permeability (9a) is replaced by a second degree polynomial near the critical point $s = 1$ ($\mathscr{S}(s) = S_{\mathrm{V}}$):

$$\kappa_\epsilon(s) := \begin{cases} \kappa(s) + \epsilon, & \text{if } s \leq 1 - \epsilon, \\ \tilde{\kappa}(s) + \epsilon, & \text{otherwise,} \end{cases} \tag{12}$$

where

$$\begin{aligned} \tilde{\kappa}(s) := {} & \frac{\kappa''(1-\epsilon)}{2}(s - (1-\epsilon))^2 \\ & + \kappa'(1-\epsilon)(s - (1-\epsilon)) + \kappa(1-\epsilon), \end{aligned}$$

see Figure 4 for a plot with a range of values of ϵ.

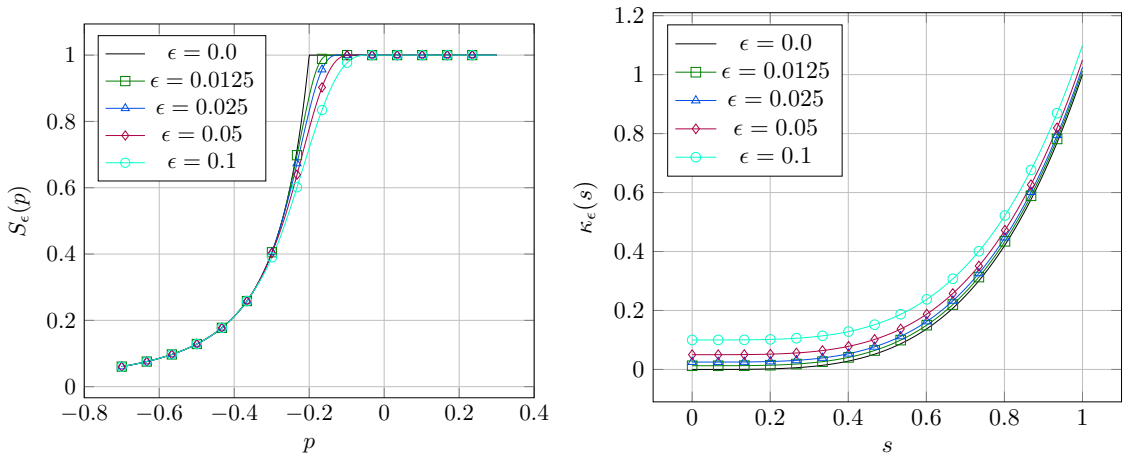

Figure 3: $[\lambda_1 = 2, p_{\mathrm{M}} = -0.2, k = 2]$ Regularization of the relative permeability (left) (10) and of the saturation (11) (right) for the Brooks–Corey model (8).

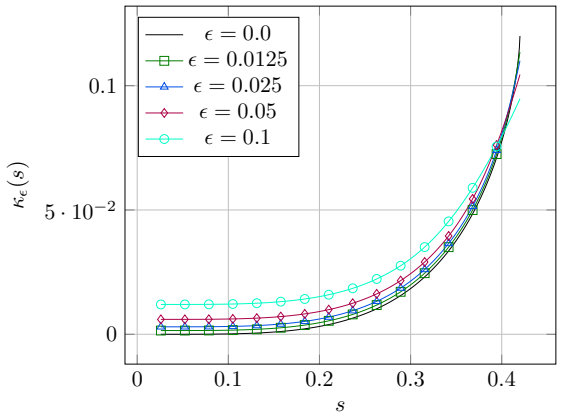

Figure 4: $[\lambda_2 = 0.66, \kappa_{\mathrm{c}} = 0.12]$ Regularization of the relative permeability (12) of the van Genuchten–Mualem model (9). We do not regularize the saturation in this case, since it it already smooth.

4 Discrete problem and solution method

In this section we give details about the discretization strategy we employ to solve the Richards equation (1). We define the lowest-order continuous finite element trial set by

$$V_h^{\mathrm{D}} := \{u_h \in H_{\mathrm{D}}^1(\Omega) : u_h|_K \in \mathcal{P}_1(K) \quad \forall K \in \mathcal{T}_h\} \tag{13}$$

as well as the test space

$$V_h^0 := \left\{u_h \in H_0^1(\Omega),\ u_h|_K \in \mathcal{P}_1(K) \quad \forall K \in \mathcal{T}_h\right\}. \tag{14}$$

4.1 Discretization

For the time discretization we use the lowest order implicit method, i.e., the backward Euler method. For each $n \in \{1, \dots, N\}$ and a given $p_{n-1,h} \in V_h^{\mathrm{D}}$, we need to find the approximate pressure $p_{n,h} \in V_h^{\mathrm{D}}$ satisfying

$$\tfrac{1}{\tau}(\phi S(p_{n,h}) - S(p_{n-1,h}), \varphi_h) + (\boldsymbol{F}(p_{n,h}), \nabla\varphi_h) = (f(\cdot,t_n), \varphi_h) + (q_{\mathrm{N}}, \varphi_h)_{\Gamma_{\mathrm{N}}} \quad \forall \varphi_h \in V_h^0, \tag{15}$$

where the flux function is defined as

$$\boldsymbol{F}(q) := \mathbf{K}\kappa(S(q))[\nabla q + \boldsymbol{g}]. \tag{16}$$

For $n = 1$, we use directly s_0 in place of $S(p_{0,h})$.

4.2 Regularization

We also consider a regularized version of problem (15). First, for a given timestep t_n, we introduce a positive sequence $\{\epsilon_n^j\}_{j\geq 1}$ such that ϵ_n^1 is independent of n (see §5). The regularized problem is then: given $p_{n-1,h}^{\bar{j}} \in V_h^{\mathrm{D}}$ find $p_{n,h}^j \in V_h^{\mathrm{D}}$ satisfying

$$\tfrac{1}{\tau}(S_{\epsilon_n^j}(p_{n,h}^j) - S_{\epsilon_n^j}(p_{n-1,h}^{\bar{j}}), \varphi_h) + (\boldsymbol{F}_{\epsilon_n^j}(p_{n,h}^j), \nabla\varphi_h) = (f(\cdot,t_n), \varphi_h) + (q_{\mathrm{N}}, \varphi_h)_{\Gamma_{\mathrm{N}}} \quad \forall \varphi_h \in V_h^0, \tag{17}$$

where the corresponding regularized flux is given by

$$\boldsymbol{F}_{\epsilon_n^j}(q) := \mathbf{K}\kappa_{\epsilon_n^j}(S_{\epsilon_n^j}(q))[\nabla q + \boldsymbol{g}] \tag{18}$$

and $\bar{j}$ is a stopping index that will be defined in §5. For $n = 1$, we still use s_0 in place of $S(p_{0,h}^j)$.

4.3 Linearization

Due to the nonlinear nature of problem (17), iterative linearization is usually used to approximate $p_{n,h}^j$. To this end, we consider the following linearized problem: given an initial guess $p_{n,h}^{j,k-1}$, find $p_{n,h}^{j,k} \in V_h^{\mathrm{D}}$ such that

$$\tfrac{1}{\tau}(\phi S_{\epsilon_n^j}(p_{n,h}^{j,k-1}) - S_{\epsilon_n^j}(p_{n-1,h}^{\bar{j},\bar{k}}), \varphi_h) + \tfrac{1}{\tau}(\phi L(p_{n,h}^{j,k} - p_{n,h}^{j,k-1}), \varphi_h) + (\boldsymbol{F}_{\epsilon_n^j}^k, \nabla\varphi_h) + (q_{\mathrm{N}}, \varphi_h)_{\Gamma_{\mathrm{N}}} = (f(\cdot,t_n), \varphi_h) \quad \forall \varphi_h \in V_h^0, \tag{19}$$

where $\bar{j}$ and $\bar{k}$ are stopping indices that will be defined in §5 and the linearized flux is given by

$$\boldsymbol{F}_{\epsilon_n^j}^k := \mathbf{K}\kappa_{\epsilon_n^j}(S_{\epsilon_n^j}(p_{n,h}^{j,k-1}))[\nabla p_{n,h}^{j,k} + \boldsymbol{g}] + \boldsymbol{\xi}(p_{n,h}^{j,k} - p_{n,h}^{j,k-1}). \tag{20}$$

Here, $(L, \boldsymbol{\xi}) \in \mathbf{L}^\infty(\Omega; \mathbb{R}^{d+1})$ depend on the specific linearization used. For the case of the modified Picard iteration [10], we set

$$L := S'_{\epsilon_n^j}(p_{n,h}^{j,k-1}), \quad \boldsymbol{\xi} := \mathbf{0}. \tag{21}$$

For Newton's method, we set

$$\begin{aligned} L &:= S'_{\epsilon_n^j}(p_{n,h}^{j,k-1}) \\ \boldsymbol{\xi} &:= \mathbf{K}(\kappa_{\epsilon_n^j} \circ S_{\epsilon_n^j})'(p_{n,h}^{j,k-1})[\nabla p_{n,h}^{j,k-1} + \boldsymbol{g}]. \end{aligned} \tag{22}$$

As before, for $n = 1$, we use s_0 in place of $S(p_{0,h}^{\bar{j},\bar{k}})$.

Remark 4.1 (Appearance of derivatives in the linearization)**.** *We note that the for both the modified Picard scheme and Newton's method, the derivative* $S'(p_{n,h}^{j,k-1})$ *appears. Additionally, in the case of Newton's method, the derivative of the composite function* $(\kappa_{\epsilon_n^j} \circ S_{\epsilon_n^j})'(p_{n,h}^{j,k-1})$ *appears. This is the motivation for the regularity requirements we impose on the regularization, i.e., Assumptions (A4) and (A5).*

4.4 A posteriori component error estimators by flux reconstruction

The key to our a posteriori error estimators will be a postprocessed approximation $\boldsymbol{\sigma}_{n,h}^{j,k}$ of the flux $\boldsymbol{F}_{\epsilon_n^j}^k$ (20) that satisfies $\boldsymbol{\sigma}_{n,h}^{j,k} \subset \boldsymbol{H}(\mathrm{div}, \Omega)$. The main tool to achieve this is the Raviart–Thomas–Nédélec (RTN) finite element space [8]. We first introduce the lowest-order broken RTN space,

$$\boldsymbol{RT}_0(\mathcal{T}_h) := \{\boldsymbol{v}_h \in [L^2(\omega)]^d : \boldsymbol{v}_h|_K \in [\mathcal{P}_0(K)]^d + \boldsymbol{x}\mathcal{P}_0(K), \forall K \in \mathcal{T}_h\}, \tag{23}$$

and the $\boldsymbol{H}(\mathrm{div}, \Omega)$-conforming space

$$\boldsymbol{RT}_0(\Omega) := \boldsymbol{RT}_0(\mathcal{T}_h) \cap \boldsymbol{H}(\mathrm{div}, \Omega). \tag{24}$$

Our general approach is in the spirit of equilibrated flux reconstruction. The method of flux reconstruction in the context of a posteriori error analysis has origins in the works of Prager and Synge [33] as well as in Ladevèze and Leguillon [26], Destuynder and Métivet [13], Braess and Schöberl [6], and Ern and Vohralík [17]. However, in this work we do not consider a full equilibration by solving local minimization problems, but rather a flux based on averaging and prescription of the degrees of freedom in $\boldsymbol{H}(\mathrm{div}, \Omega)$ as in [40, 16]. In general this type of estimator satisfies the equilibration with the source term in a weak sense.

First, we introduce some additional notation for the mesh. Let $\mathcal{F}$ denote the set of faces in the mesh and for a face $F \in \mathcal{F}$, let $\mathcal{T}_F$ denote the edge patch of F, i.e.,

$$\mathcal{T}_F := \{K \in \mathcal{T}_h : F \subset \overline{K}\}. \tag{25}$$

Then we define the reconstructed flux $\boldsymbol{\sigma}_{n,h}^{j,k} \in \boldsymbol{RT}_0(\Omega)$ by

$$\frac{1}{|F|}\int_F \boldsymbol{\sigma}_{n,h}^{j,k} \cdot \boldsymbol{n}_F \,\mathrm{d}S = \frac{1}{|\mathcal{T}_F||F|}\sum_{K\in\mathcal{T}_F}\int_F -\boldsymbol{F}_{\epsilon_n^j}^k \cdot \boldsymbol{n}_F \,\mathrm{d}S \quad \forall F \in \mathcal{F}, \tag{26}$$

where $|\mathcal{T}_F|$ denotes the cardinality of $\mathcal{T}_F$.

The conditions in (26) totally determine the function $\boldsymbol{\sigma}_{n,h}^{j,k}$, see, e.g., [8]. We thus define the following estimators with the help of the reconstructed flux (26):

$$\eta_{\mathrm{dis}}^{n,j,k} := \|\boldsymbol{F}_{\epsilon_n^j}^k + \boldsymbol{\sigma}_{n,h}^{j,k}\| \qquad \text{(discretization)}, \tag{27a}$$

$$\eta_{\mathrm{lin}}^{n,j,k} := \|\boldsymbol{F}_{\epsilon_n^j}(p_{n,h}^{j,k}) - \boldsymbol{F}_{\epsilon_n^j}^k\| \qquad \text{(linearization)}, \tag{27b}$$

$$\eta_{\mathrm{reg}}^{n,j,k} := \|\boldsymbol{F}(p_{n,h}^{j,k}) - \boldsymbol{F}_{\epsilon_n^j}(p_{n,h}^{j,k})\| \qquad \text{(regularization)}. \tag{27c}$$

Remark 4.2 (Choice of estimators)**.** *We take inspiration for the discretization estimator from [17, 18, 30], where our estimator can be thought of a simplified version of* η^{F}*. We make this choice because the current definition is very cheap to compute as it does not require the solution of local problems. The decomposition into component estimators is very much inspired by those established in [17, 18]. In [17], a decomposition was established that identified errors associated with the discretization, linearization, algebra, and quadrature. These estimators were then used to define stopping criteria for the nested nonlinear and linear solvers. More recently in [18], we consider a regularized problem and introduce a corresponding regularization estimator, leading to the same (at least in spirit) choice of estimators as in* (27)*. In [18] we rigorously prove that the estimators tend to zero in their respective limits, i.e.,* $\eta_{\mathrm{lin}}^{n,j,k}$ *tends to zero as* $k \to \infty$ *and* $\eta_{\mathrm{reg}}^{n,j,k}$ *tends to zero as* $j, k \to \infty$*.*

5 Adaptive algorithm

In this section we present an adaptive algorithm for iteratively approximating the solution of the nonlinear algebraic equations (15), Algorithm 1. For a given timestep t_n, the algorithm constructs a sequence

of regularized problems, with regularization parameter ϵ_n^j, and linearization iterations indexed by k, producing intermediate solutions $p_{n,h}^j$ and $p_{n,h}^{j,k}$ as per §4.2 and §4.3. The algorithm takes some user-specified parameters, starting with an initial regularization parameter $\bar{\epsilon}^1 > 0$ and an initial contraction factor $\bar{C}^1 \in (0,1)$. We take inspiration from Algorithm 1 in [18] to define the following stopping criteria, where the bars denote stopping indices,

$$\eta_{\text{lin}}^{n,j,\bar{k}} < \gamma_{\text{lin}} \eta_{\text{reg}}^{n,j,\bar{k}}, \tag{28a}$$

$$\eta_{\text{reg}}^{n,\bar{j},\bar{k}} < \gamma_{\text{reg}} \eta_{\text{dis}}^{n,\bar{j},\bar{k}}, \tag{28b}$$

where $\gamma_{\text{reg}}, \gamma_{\text{lin}} > 0$ are user-specified parameters. The first criterion (28a) states that the linearization procedure should not continue on a given regularized problem if it has sufficiently converged. The second criterion (28b) states that, on a given timestep t_n, the error introduced by regularization should only be γ_{reg}-times smaller than the inherent error due to discretization.

In more details, the algorithm proceeds as follows: we start on a given timestep t_n with the initial regularization parameter $\epsilon_n^j := \bar{\epsilon}^1$ and contraction factor $C_n^j := \bar{C}^1$. We proceed to iterate in the linearization until the first stopping criterion (28a) is satisfied. However, we also have a safety measure (line 16 of the algorithm) to check whether the linearization error does not increase between the previous and current linearization iterates $k-1$ and k. If this is the case, we revert the regularization parameter and reset the approximate pressure, with the help of p_{prev}. Indeed, p_{prev} acts as a checkpoint, as it is initialized with the initial guess, and then updated to store $p_{n,h}^{j,\bar{k}}$ every time the linearization has converged successfully on line 24. After this reset, we increase the current contraction factor C_n^j which limits the amount we decrease the regularization parameter between the steps j and $j+1$. This strategy has some common aspects to the usual practice of cutting the timestep to provide a better initial guess, but the advantage here is that we only "go back" one value of the regularization parameter and not to the beginning of the timestep.

We also remark that the initial guess can be taken as $p_{0,h}^{0,0} := S^{-1}(s_0)$ in the regime where S is invertible, namely $s_0 < 1$. For the points $x \in \Omega$ where $s_0(x) = 1$, we simply take the initial guess $p_{0,h}^{0,0}(x) := p_{\text{M}}(x)$.

6 Numerical experiments

In this section we detail numerical experiments using our adaptive regularization of Algorithm 1. In particular, we consider five examples where a plain Newton solver struggles to converge. In all cases our adaptive algorithm succeeds. All numerical experiments are conducted with the help of the `Gridap.jl` library [2, 39] in the Julia programming language. For all the experiments, we take the linearization parameters $\gamma_{\text{reg}} = 0.2, \gamma_{\text{lin}} = 0.3$, $\bar{C}^1 = 0.1$, and $\bar{\epsilon}^1 = 0.1$. For comparison, we also test the unregularized Newton's method (corresponding to taking $\bar{\epsilon}^1 = 0$ in Algorithm 1), unregularized Newton's method with adaptive time step cutting and the modified Picard scheme (21). In the unregularized case, instead of criterion (28a), we ensure that

$$\eta_{\text{lin}}^{n,j,k} < 10^{-7}. \tag{29}$$

Remark 6.1 (Choice of the stopping criterion in the unregularized case)**.** *We use a fixed stopping criterion for the linearization in* (29) *because we would like to compare our adaptive strategy with a nonadaptive one, which is the much more common approach. For example, even in [37] where the solver is chosen adaptively, the authors use a fixed stopping criterion for terminating the linearization procedure. Namely, their criterion ensures that the difference of two consecutive iterates of the approximate measured in an iteration-dependent norm is less than 1e-7. We therefore adopt the same tolerance in our test cases, but for the linearization estimator* $\eta_{\text{lin}}^{n,j,k}$*.*

Remark 6.2 (Adaptive timestep cutting)**.** *In order to compare with the common approach of adaptive timestep cutting, we implement a simple timestep cut in the case of* $\bar{\epsilon}^1 = 0$ *for the unregularized Newton method in Algorithm 2. There are 3 parameters that dictate the behavior of this algorithm. The basic idea is to cut the timestep when the number of Newton iterations on a given timestep exceeds* n_{prev}*, and to increase the timestep if the average number of Newton steps* k_n *over the previous* n_{prev} *timesteps is less than* k_{ave}*; however, we only increase the timestep at most back to the initial chosen timestep size* τ_0*. For all our experiments, we take* $k_{\max} = 20, k_{\text{ave}} = 10$*, and* $n_{\text{prev}} = 10$*.*

Algorithm 1: Adaptive regularization for the Richards equation

Initialization: Choose an initial guess $p_{1,h}^{1,0} := p_{0,h}^{0,0} \in V_h^{\mathrm{D}}$ and initialize the timestep counter $n := 0$, $\bar{j} = \bar{k} = 0$

Parameters: $\gamma_{\mathrm{reg}}, \gamma_{\mathrm{lin}}, \bar{\epsilon}^1, \bar{C}^1 \in (0,1)$, τ_0

$t_0 := 0$, $p_{\mathrm{prev}} := p_{0,h}^{\bar{j},\bar{k}}$

while $t_n < T$ **do**

  Update $n := n + 1$

  Update $t_n := t_{n-1} + \tau_0$

  Initialize $j := 0, \bar{k} := 0, \bar{j} := 1$

  Reset $C_n^1 := \bar{C}^1, \epsilon_n^1 := \bar{\epsilon}^1$

  Loop for regularization

    Increment $j := j + 1$

    Initialize $k := 0$

    $\eta_{\mathrm{lin}}^{n,j,k} := \infty$

    Loop for linearization

      Increment $k := k + 1$

      Solve for $p_{n,h}^{j,k}$ in (19)

      Compute the estimators (27)

      if $\eta_{\mathrm{lin}}^{n,j,k} > \eta_{\mathrm{lin}}^{n,j,k-1}$ **then**

        Reset $p_{n,h}^{j,k} := p_{\mathrm{prev}}$

        Increase $\epsilon_n^{j+1} := \epsilon_n^j / C_n^j$

        Increase $C_n^{j+1} := \sqrt{C_n^j}$

        go to line 8

      end

    until $\eta_{\mathrm{lin}}^{n,j,k} < \gamma_{\mathrm{lin}} \eta_{\mathrm{reg}}^{n,j,k}$

    Update $\bar{k} := k$

    $p_{n,h}^{j,0} := p_{\mathrm{prev}} := p_{n,h}^{j,\bar{k}}$

    Decrease $\epsilon_n^{j+1} := C_n^j \, \epsilon_n^j$

    Update $C_n^{j+1} := C_n^j$

  until $\eta_{\mathrm{reg}}^{n,j,\bar{k}} < \gamma_{\mathrm{reg}} \eta_{\mathrm{dis}}^{n,j,\bar{k}}$

  Update $\bar{j} := j$

  $p_{n+1,h}^{1,0} := p_{n,h}^{\bar{j},\bar{k}}$

end

return $\{p_{n,h}^{\bar{j},\bar{k}}\}_{n=1}^N$

6.1 Strictly unsaturated medium

In this test case, we seek to reproduce the results obtained in [37, §4.1]. This means we have the following data:

- $\Omega = \Omega_1 \cup \Omega_2$, $\Omega_1 = (0,1) \times (0, 1/4], \Omega_2 = (0,1) \times (1/4, 1)$
- Uniform mesh with $40 \times 40 \times 2$ elements
- $T = 1$
- $\tau_0 = 1$
- $\Gamma_{\mathrm{D}} = \partial\Omega \cap \{y = 1\}$
- $\Gamma_{\mathrm{N}} = \partial\Omega \setminus \Gamma_{\mathrm{D}}$
- $\boldsymbol{g} = (0,1)^T$
- $\mathbf{K} = \boldsymbol{I}$

Algorithm 2: Timestep cutting for the Richards equation

Initialization: Choose an initial guess $p_{1,h}^{j,k} := p_{0,h}^{0,0} \in V_h^{\mathrm{D}}$ and initialize the timestep counter $n := 0$, $j := 1$, $\bar{k} := \bar{j} := 1$

Parameters : $k_{\max}, k_{\mathrm{ave}}, n_{\mathrm{prev}}, \tau_0$

1 $t_0 := 0, \tau := \tau_0$, $p_{\mathrm{prev}} := p_{0,h}^{0,0}$
2 **while** $t_n < T$ **do**
3 Update $n := n + 1$
4 Update $t_n := t_{n-1} + \tau$
5 Initialize $k := 0$, $\epsilon_n^j := 0$
6 **Loop** for linearization
7 Solve for $p_{n,h}^{j,k}$ in (19)
8 Compute $\eta_{\mathrm{lin}}^{n,j,k}$ in (27)
9 Increment $k := k + 1$
10 **if** $k_n > k_{\max}$ **then**
11 $\tau := \tau/2$
12 $t_n := t_{n-1}$
13 $n := n - 1$
14 $p_{n,h}^{j,k} = p_{\mathrm{prev}}$
15 **go to line** 2
16 **end**
17 Update $k_n = \bar{k} := k$
18 $p_{n+1,h}^{j,0} := p_{\mathrm{prev}} := p_{n,h}^{j,\bar{k}}$
19 **until** $\eta_{\mathrm{lin}}^{n,j,k} < 1 \cdot 10^{-7}$
20 **if** $\mathrm{mod}(n, n_{\mathrm{prev}}) = 0$ **then**
21 **if** $\frac{1}{n_{\mathrm{prev}}} \left(\sum_{m=n-n_{\mathrm{prev}}+1}^{n} k_n \right) < k_{\mathrm{ave}}$ **then**
22 $\tau := \min\{2\tau, \tau_0\}$
23 **end**
24 **end**
25 **return** $\{p_{n,h}^{\bar{j},\bar{k}}\}_{n=1}^{N}$
26 **end**

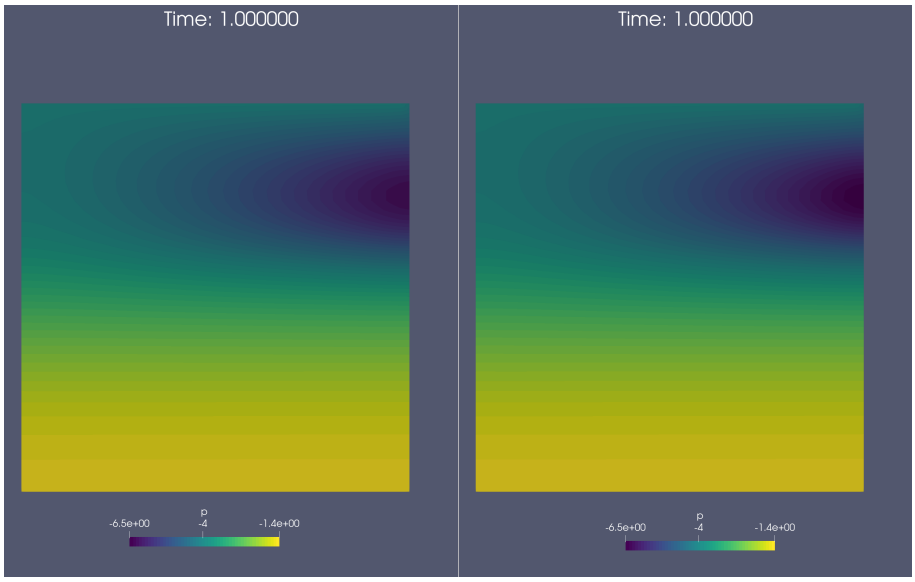

Figure 5: [§6.1, van Genuchten–Mualem model (9) with $p_{\mathrm{M}} = 0, S_{\mathrm{R}} = 0.026, S_{\mathrm{V}} = 0.42, \kappa_{\mathrm{c}} = 0.12, \alpha = 0.551, \lambda_2 = 0.655$, solver parameters $\gamma_{\mathrm{reg}} = 0.2, \gamma_{\mathrm{lin}} = 0.3$, $\bar{C}^1 = 0.1$] Approximate pressure $p_{n,h}^{\bar{j},\bar{k}}$ for the problem in §6.1 using Algorithm 1 with Newton's method (22) and adaptive regularization $\bar{\epsilon}^1 = 0.1$ (left) and modified Picard with no regularization $\bar{\epsilon}^1 = 0$ (right).

- $f(x,y) = \begin{cases} 0 & (x,y) \in \Omega_1 \\ 0.06\cos(\frac{4}{3}\pi y)\sin(x) & (x,y) \in \Omega_2 \end{cases}$
- $p_0(x,y) = \begin{cases} -y - 1/4 & (x,y) \in \Omega_1 \\ -4 & (x,y) \in \Omega_2, \end{cases}$
- $s_0 = S(p_0)$
- $p_{\mathrm{D}} = p_0|_{\Gamma_{\mathrm{D}}}$
- $q_{\mathrm{N}} = 0$
- $\phi = 1$

We use the van Genuchten–Mualem model (9) with the parameters specified in Figure 5. Please note that there is only 1 timestep. We first plot the approximate pressure at the final step of both Algorithm 1 with Newton's linearization (22), as well as the modified Picard iteration (21) with no regularization, see Figure 5. We observe that the two not only match well but are also comparable with the results in [37, §4.1]. In this case, Newton's method without regularization diverged, which is consistent with what is reported in [37, §4.1].

We now look more carefully at the evolution of the estimators in the adaptive algorithm for this example. In Figure 6, we plot the component estimators as a function of cumulative linearization steps. The components are all of the order of 0.1 on the first iteration. We see that the linearization estimator converges very rapidly for a given value of the regularization parameter $\epsilon_1^1 = \bar{\epsilon}^1 = 0.1$, then $\epsilon_1^2 = 0.01, \epsilon_1^3 = 0.001$ and $\epsilon_1^4 = 0.0001$. Furthermore, once we lower the regularization parameter, the regularization component estimator clearly decreases. On the final iteration, we see the discretization and regularization estimators stabilize with a constant gap between the two.

6.2 Injection test case

This test case is inspired by the one presented in [7, §4.1]. In particular, we use the following model parameters:

- $\Omega = (0,1)^2$
- quasi uniform mesh with $h = 2.82 \cdot 10^{-2}$

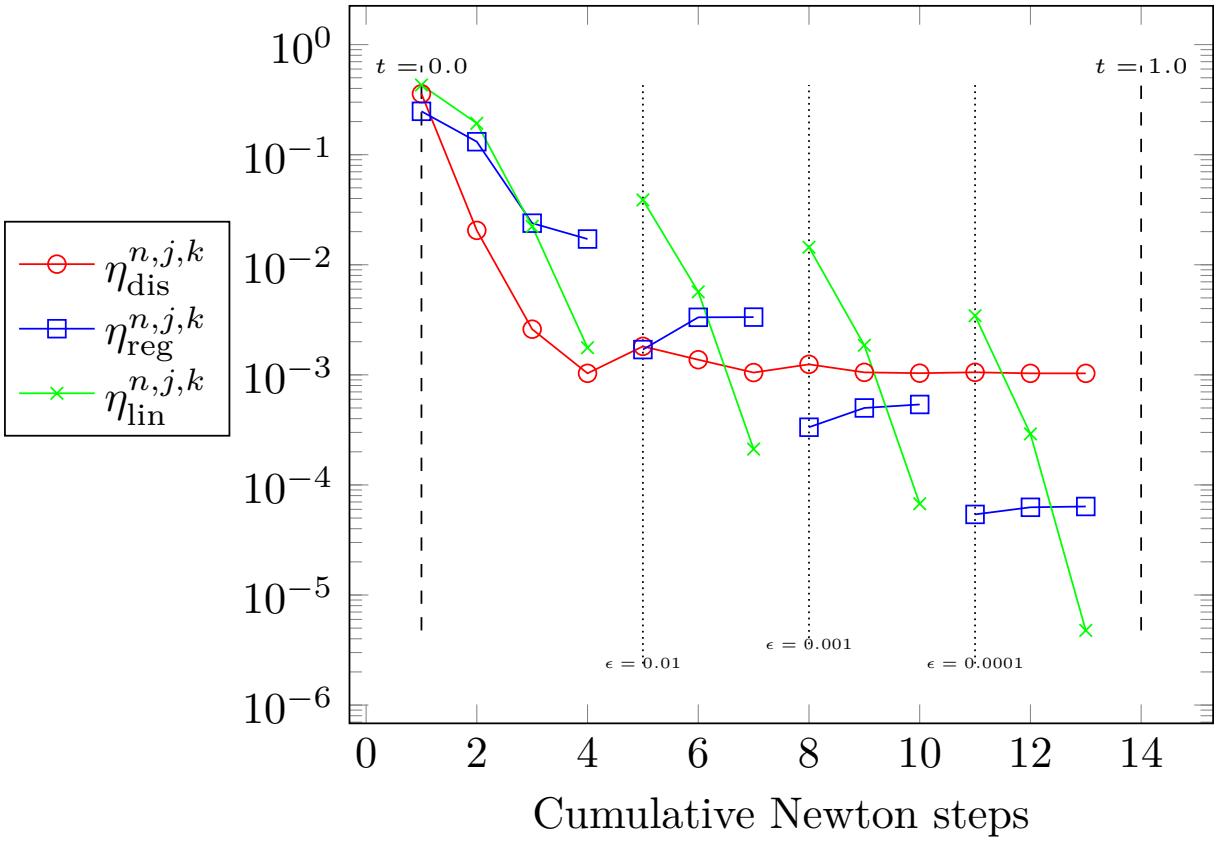

Figure 6: [§6.1, van Genuchten–Mualem model (9) with $p_{\text{M}} = 0, S_{\text{R}} = 0.026, S_{\text{V}} = 0.42, \kappa_{\text{c}} = 0.12, \alpha = 0.551, \lambda_2 = 0.655$, solver parameters $\gamma_{\text{reg}} = 0.2, \gamma_{\text{lin}} = 0.3$, $\bar{C}^1 = 0.1$, $\bar{\epsilon}^1 = 0.1$] Evolution of the component estimators (27) for the Algorithm 1 with Newton's method (22) and adaptive regularization with $\bar{\epsilon}^1 = 0.1$ applied to the test problem in §6.1.

- $T = 1$
- $\tau_0 = 2.82 \cdot 10^{-2}$
- $\Gamma_{\mathrm{D}} = \{(x_1, x_2) | x_1 \in (0, 0.3), x_2 = 1\}$
- $\Gamma_{\mathrm{N}} = \partial\Omega \setminus \Gamma_{\mathrm{D}}$
- $\boldsymbol{g} = (0, -1)^T$
- $\mathbf{K} = \boldsymbol{I}$
- $f = 0$
- $p_0 = -1$
- $s_0 = S(p_0)$
- $p_{\mathrm{D}} = 1$
- $q_{\mathrm{N}} = 0$
- $\phi = 1$

We use the Brooks–Corey model (8) with parameters specified in Figure 8. We note firstly that there is an inconsistency between the trace of p_0 and the imposed boundary condition p_{D} at $t = 0$. This is mathematically valid, but can cause problems for the solver as we shall see shortly. The domain is initially "mildly dry" with $S(p_0) = s_0 = 0.027$. We remark that we were not able to consider a smaller value of s_0 as was done in, e.g., [7, 4]. It would likely be necessary to implement their variable switching strategy for this, which we discuss in §7. However, the current test case still remains challenging for Newton's method.

In Figure 7, the total stepwise and cumulative iterations are plotted for Algorithm 1 using Newton's method with and without regularization as well as the modified Picard method without regularization. First of all, it is clear that Newton's method without regularization is not feasible and we stop the solver after 300 iterations on the first step. Next, we note that modified Picard takes consistently more iterations than the Newton solver with regularization, resulting in an approximate 3.3x speedup at the end of the simulation (1004 cumulated iterations for modified Picard vs 297 for the regularized Newton solver). We also note that the number of iterations is somewhat less stable for modified Picard with peaks of 56 iterations at $t = 0.42$, and 48 iterations at $t = 0.22$. In contrast, the regularized Newton solver takes no more than 13 iterations per timestep. In fact, this only occurs at the beginning of the simulation. Finally, the unregularized Newton method with adaptive timestep cutting converges always in a few linearization iterations, but many timesteps are necessary, so that it becomes the most expensive approach overall.

In Figure 8 we see that Newton's method has trouble converging for the regularization parameter $\epsilon_n^j = 0.1$ and the linearization estimator increases during the second and third timesteps, thus triggering the if statement on line 16 of Algorithm 1. Indeed, we see consequently the algorithm recovers by simultaneously increasing the regularization parameter to $\epsilon_n^2 := \epsilon_n^1 / C_n^1$ and then increasing the contraction factor $C_n^1 = \sqrt{C_n^1}$. This combination allows the estimator to converge on the following series of regularized problem until the stopping criterion (28b) is achieved and the solver advances to the next timestep.

We now consider the effect of the regularization on the solution. In particular, in Figure 9, we compare side by side the plots of the saturation profile for the regularized and unregularized solutions. The profiles match well, and we also note that the regularized profile appears smoother at the interface.

6.3 Realistic test case

In this test, we take inspiration from [30, §6.3] by using the following model parameters:

- $\Omega = (0, 1)^2$
- quasi uniform mesh with $h = 2.02 \cdot 10^{-2}$
- $T = 1$
- $\tau_0 = 2.02 \cdot 10^{-2}$
- $\boldsymbol{g} = (-1, 0)^T$

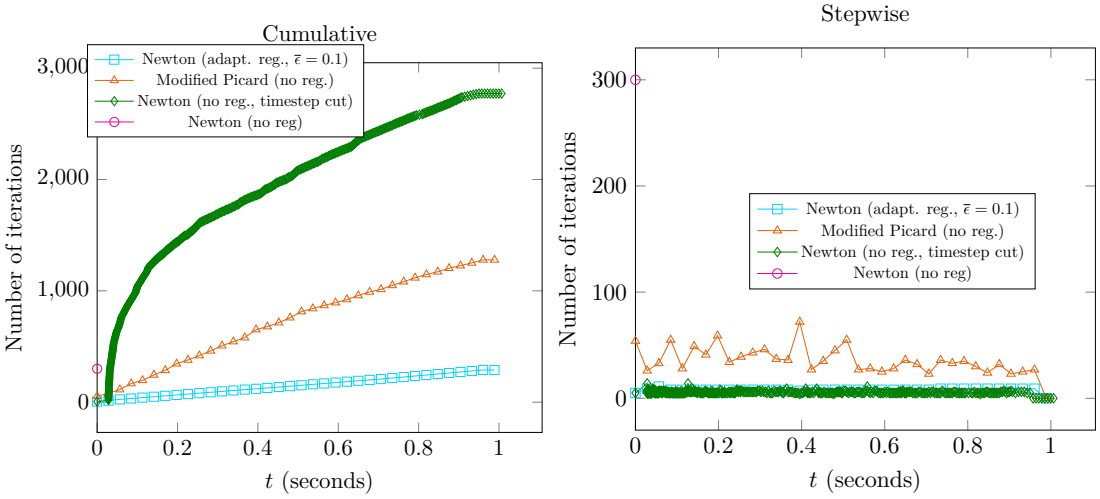

Figure 7: [§6.2, Brooks–Corey model (8) with $p_{\mathrm{M}} = -0.2, \lambda_1 = 2.239$, solver parameters $\gamma_{\mathrm{reg}} = 0.2, \gamma_{\mathrm{lin}} = 0.3$] Comparison of the total cumulative and stepwise iterations for the four strategies.

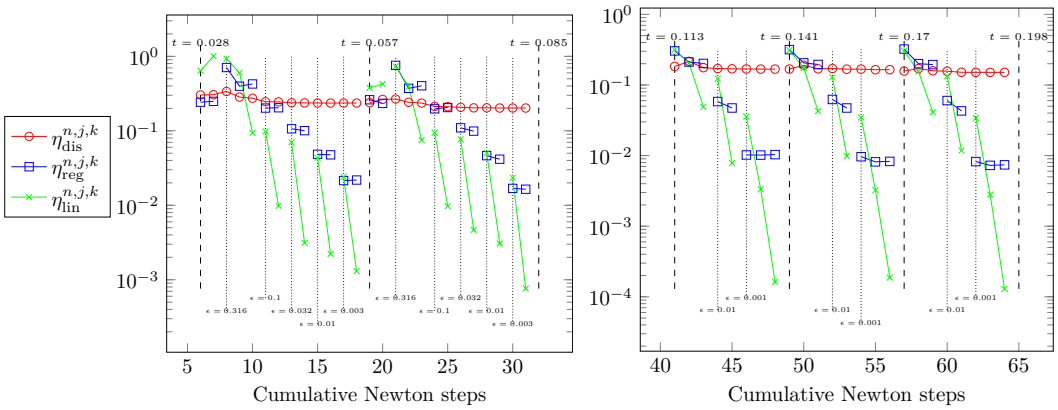

Figure 8: [§6.2, Brooks–Corey model (8) with $p_{\mathrm{M}} = -0.2, \lambda_1 = 2.239$, solver parameters $\gamma_{\mathrm{reg}} = 0.2, \gamma_{\mathrm{lin}} = 0.3, \bar{C}^1 = 0.1, \bar{\epsilon}^1 = 0.1$] Plots of the evolution of the estimators on the second and third timesteps (left) and of the fifth, sixth and seventh timesteps (right).

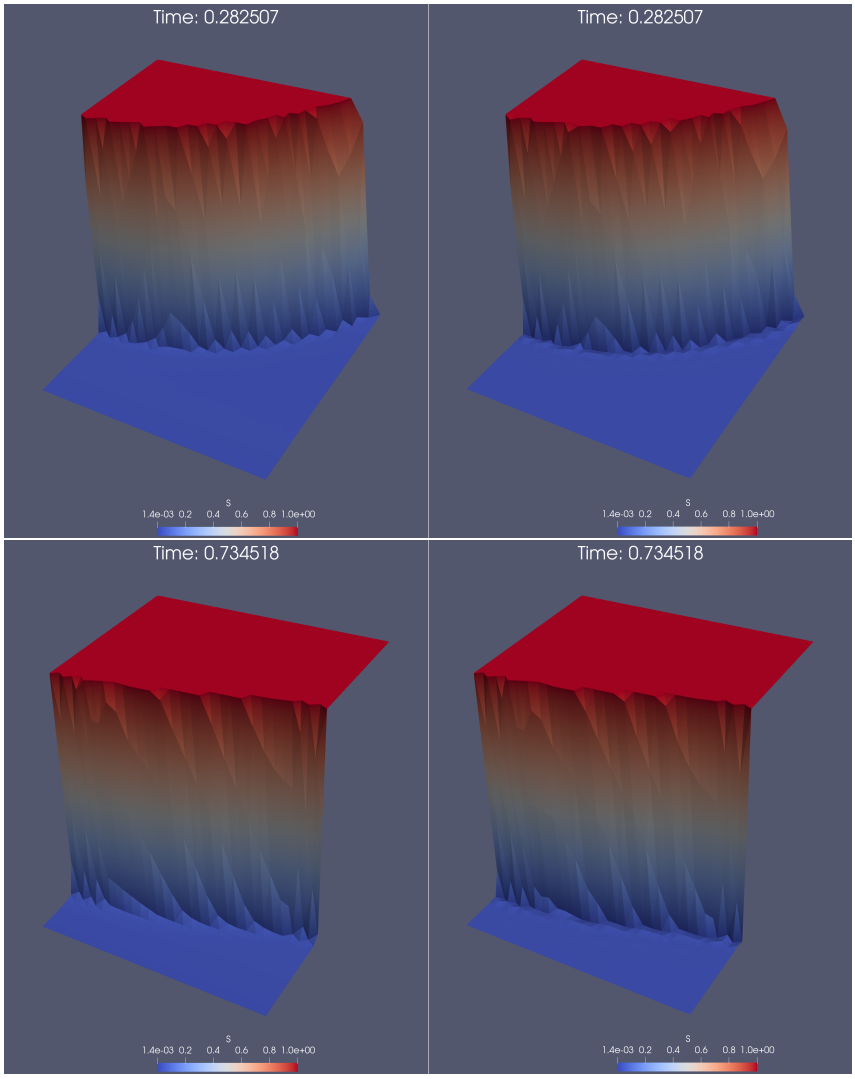

Figure 9: [§6.2, Brooks–Corey model (8) with $p_{\mathrm{M}} = -0.2, \lambda_1 = 2.239$, solver parameters $\gamma_{\mathrm{reg}} = 0.2, \gamma_{\mathrm{lin}} = 0.3, \bar{C}^1 = 0.1$] Two snapshots comparing the evolution of the saturation field $s = S(p_{n,h}^{\bar{j},\bar{k}})$ using Algorithm 1 with Newton's method and adaptive regularization $\bar{\epsilon}^1 = 0.1$ (left) and modified Picard with no regularization $\bar{\epsilon}^1 = 0$ (right).

- $\mathbf{Q} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$
- $K_\phi = 0.1$
- $f = 0$
- $p_{\mathrm{L}}(\boldsymbol{x}) = \left(\frac{p_{\mathrm{out}} - p_{\mathrm{in}}}{0.5}\right) \boldsymbol{x}$
- $p_{\mathrm{out}} = -2.0$
- $p_{\mathrm{in}} = -0.2$
- $p_{\mathrm{D}} = p_0|_{\Gamma_{\mathrm{D}}}$
- $q_{\mathrm{N}} = 0$
- $\phi = 1$

where the initial condition and boundary conditions are fully specified with the help of the schema in Figure 10. We use the Brooks–Corey model (8) with parameters specified in Figure 12.

We begin by comparing the stepwise and cumulative number of iterations in Figure 11. We first remark that Newton's method without regularization takes an unreasonable number of iterations on the first step (we stop the solver at 300 iterations). We then consider our timestep cutting strategy as presented in Algorithm 2. We see that for this problem, the timestep is cut by the algorithm initially until many very small steps are taken. Then, the timestep is slowly increased, but the cumulative number of linearization steps ends up being the greatest of all the successful methods. The modified Picard is able to finish the simulation but has some big peaks, namely 208 at $t = 0.82$, 101 iterations at $t = 0.54$, and 77 at $t = 0.78$. In contrast, the number of iterations per step for the regularized Newton solver does not exceed 20. This gain is reflected clearly when comparing the cumulative number of iterations where by the end, modified Picard has taken almost 5-times as many iterations as the regularized Newton solver (2359 vs. 576).

To better understand how the adaptive algorithm works, we refer to Figure 12. In the left figure we see that no problems are encountered at the timesteps $t = 0$ through $t = 0.061$. In the right figure we plot the estimators around the time of the contact with the interface at $t = 0.445$–0.486 and we see that the linearization estimator begins to increase on the first timestep for $\epsilon = 0.001$. The condition of the if statement on line 16 is then true, resetting to the result at the previous value of $\epsilon = 0.01$, and increasing the contraction factor C_n^j thereby decreasing the "distance" between the two consecutive regularized problems. This allows the algorithm to proceed, albeit with more intermediate values of ϵ, to the end of the timestep.

Finally, we compare snapshots of the saturation for two timesteps $t = 0.40$ and $t = 0.95$ in Figure 13. As in the previous examples, the two profiles are comparable with the regularized version appearing smoother at the boundary of the evolving interface.

6.4 Celia et al.'s test case

In this test case, we consider the problem first presented in [10] and later in [15]. The problem data is as follows.

- $\Omega = (0, 1)$
- Uniform mesh with with $40 \times 40 \times 2$ triangles
- $T = 84600$ s (one day)
- $\tau_0 = 60$ s
- $\boldsymbol{g} = (0, 0)^T$
- $\mathbf{K} = \boldsymbol{I}$
- $f = 0$
- $\Gamma_{\mathrm{D}} = \partial\Omega$

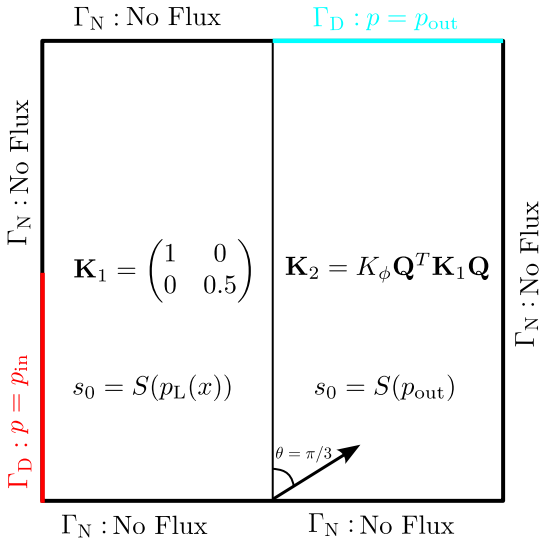

Figure 10: Schematic of the boundary and initial conditions for the test problem considered in §6.3.

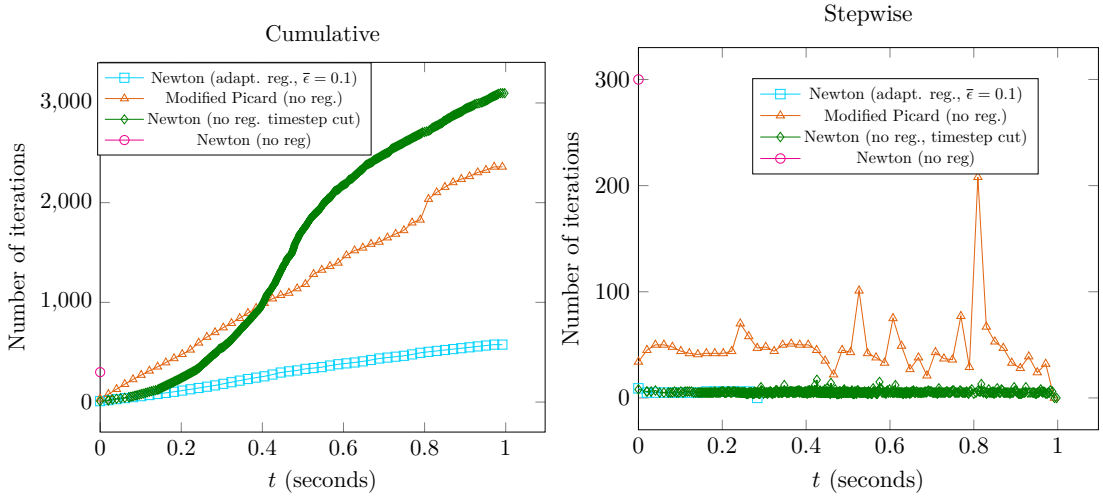

Figure 11: [§6.3, Brooks–Corey model (8) with $p_M = -0.2, \lambda_1 = 2$, solver parameters $\gamma_{reg} = 0.2, \gamma_{lin} = 0.3, \bar{C}^1 = 0.1, \bar{\epsilon}^1 = 0.1$] Comparison of the total cumulative and stepwise iterations for the four strategies.

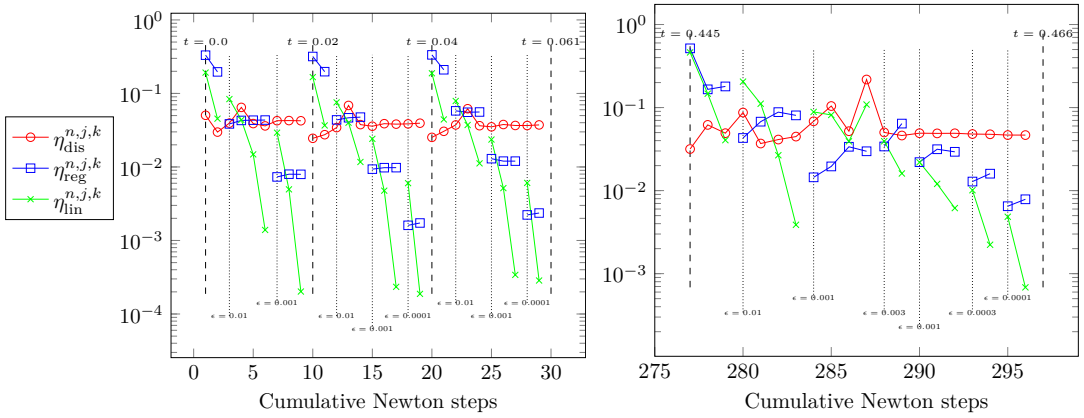

Figure 12: [§6.3, Brooks–Corey model (8) with $p_M = -0.2, \lambda_1 = 2$, solver parameters $\gamma_{reg} = 0.2, \gamma_{lin} = 0.3, \bar{C}^1 = 0.1, \bar{\epsilon}^1 = 0.1$] Evolution of the estimators on the first and second timesteps (left) and of the 22nd, 23rd and 24th timesteps (right).

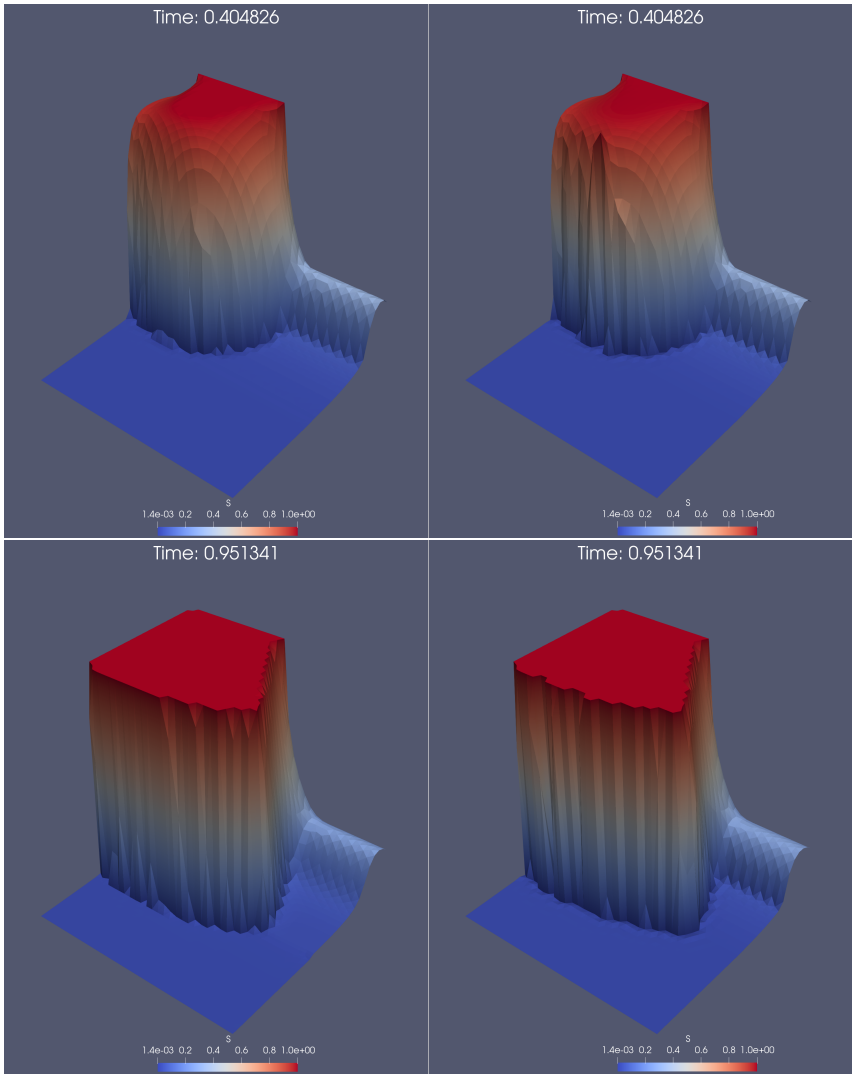

Figure 13: [§6.2, Brooks–Corey model (8) with $p_{\mathrm{M}} = -0.2, \lambda_1 = 2$, solver parameters $\gamma_{\mathrm{reg}} = 0.2, \gamma_{\mathrm{lin}} = 0.3, \bar{C}^1 = 0.1$] Two snapshots comparing the evolution of the saturation field $s = S(p_{n,h}^{\bar{j},\bar{k}})$ using the adaptive regularization Algorithm 1 with Newton's method and $\bar{\epsilon}^1 = 0.1$ (left) and modified Picard with $\bar{\epsilon}^1 = 0$ (right).

- $p = -0.75$ m at $x = 0$, $p = -10$ m at $x = 1$
- $s_0 = S(p_0)$ where $p_0 = -10$ m

The parameters for van Genuchten–Mualem are given in Figure 14. The results for the 1D profile are presented in Figure 14; they are in good agreement with those obtained in [10, 15]. The regularized and unregularized (with adaptive timestep cutting) versions are quite close. For the regularized version, we modify the adaptive algorithm slightly by not resetting the regularization parameter on each timestep as per line 7. This choice is made since this problem does not pose substantial difficulty to the Newton solver and there is a very large number of timesteps. We present comparison results for the performance of the algorithms in Figure 15. In this case, the unregularized version performs slightly better and ultimately takes fewer steps overall. This example therefore demonstrates that an (adaptive) regularization is not necessary in a case where the nonlinear solver does not struggle considerably.

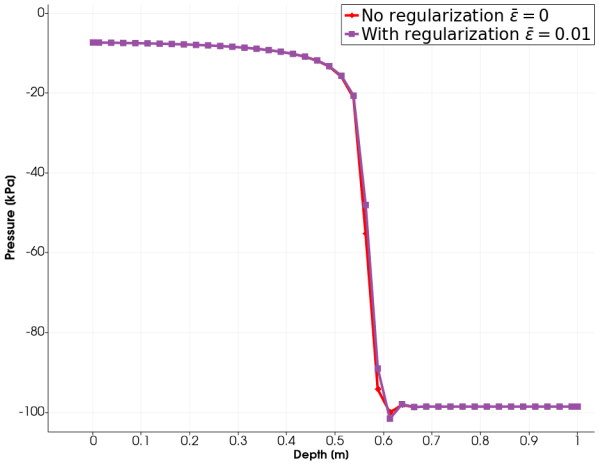

Figure 14: [§6.1, van Genuchten–Mualem model (9) with $p_{\mathrm{m}} = 0, S_{\mathrm{R}} = 0.227, S_{\mathrm{V}} = 1.0, \kappa_{\mathrm{c}} = 0.922 \cdot 10^{-4}, \alpha = 0.551, \lambda_2 = 2.0$, solver parameters: $\gamma_{\mathrm{reg}} = 0.2, \gamma_{\mathrm{lin}} = 0.3$, $\bar{C}^1 = 0.01$, $\bar{\epsilon}^1 = 0.01$] Comparison at the final time of one day for the test case in §6.4.

6.5 Perched water table test case

We now present our final test case, inspired by the one presented in Kirkland et. al [24] and subsequently adapted in [15, 5]. We use the following problem data.

- $\Omega = (-2.5 \text{ m}, 2.5 \text{ m}) \times (-3 \text{ m}, 0 \text{ m})$
- quasi uniform mesh with $h = 8.2 \cdot 10^{-2}$
- $T = 86400$ s (one day)
- $\tau_0 = 60$ s, (increase $\tau_n := 1.2\tau_{n-1}$ for $n \geq 1$ as in [5])
- $\boldsymbol{g} = (-1, 0)^T$
- $\mathbf{K} = \boldsymbol{I}$
- $f = 0$
- $\Gamma_{\mathrm{N}} = \partial\Omega$
- $q_{\mathrm{N}}(\boldsymbol{x}) = \begin{cases} 5.78 \cdot 10^{-3} \text{ ms}^{-1} & \text{if } \boldsymbol{x} \in \Gamma_{\mathrm{in}} \\ 0 \text{ ms}^{-1} & \text{otherwise} \end{cases}$
- $\Gamma_{\mathrm{in}} := (-1.5 \text{ m}, 1.5 \text{ m}) \times \{0 \text{ m}\}$
- Initial condition $s_0 = S(p_0)$ with $p_0 = -300$ m

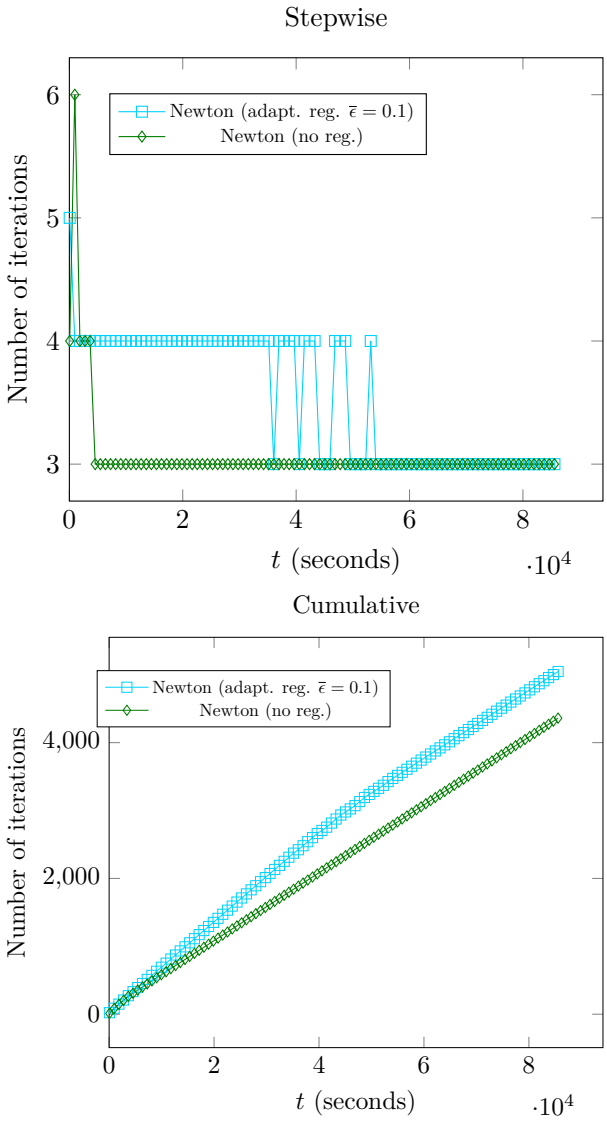

Figure 15: [§6.1, van Genuchten–Mualem model (9) with $p_{\mathrm{m}} = 0, S_{\mathrm{R}} = 0.227, S_{\mathrm{V}} = 1.0, \kappa_{\mathrm{c}} = 0.922 \cdot 10^{-4}, \alpha = 0.551, \lambda_2 = 2.0$, solver parameters: $\gamma_{\mathrm{reg}} = 0.2, \gamma_{\mathrm{lin}} = 0.3$, $\bar{C}^1 = 0.01$, $\bar{\epsilon}^1 = 0.01$]Comparison of the modified regularization strategy and the unregularized version for the test case in §6.4.

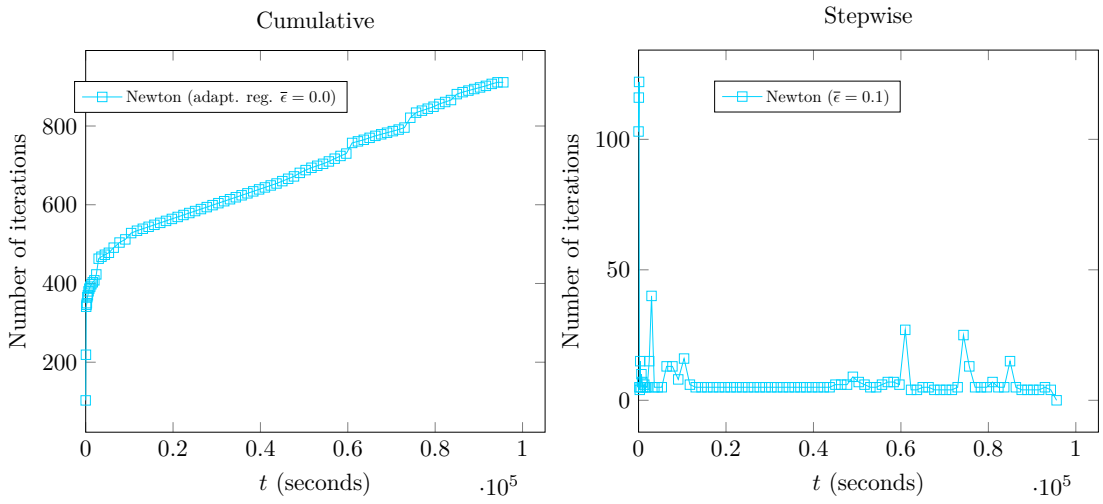

Figure 16: [§6.5, parameters given in Table 1, solver parameter $\bar{\epsilon} = 0.01, \gamma_{\text{reg}} = 1.0, \gamma_{\text{lin}} = 0.3$] Number of cumulative and stepwise iterations counts for the regularized Newton method (the other considered methods failed to converge for this test case).

Table 1: §6.5 parameter values for the van Genuchten–Mualem model (9). $p_{\text{M}} = 0$ for both materials and we take $C_\epsilon = 0.01, \gamma_{\text{reg}} = 1, \gamma_{\text{lin}} = 0.3$.

Material	κ_{c}	ϕ	S_{R}	S_{V}	λ_2	α
Sand	6.262×10^{-5}	0.368	0.07818	1	0.553	2.8
Clay	1.516×10^{-6}	0.4686	0.2262	1	0.2835	1.04

The domain is composed of two material regions, sand and clay, where the regions are shown in Figure 18. The material parameters are given in Table 1. We show the evolution of applying our adaptive Algorithm 1 in Figure 17. The profiles are similar to those in Figure 20 of [5] with similar data. In this case, the adaptive timestep cutting and modified Picard methods could not even advance past the first timestep. More precisely, all the unregularized Newton methods encountered a singular system, where cutting the timestep did not change this, even for timesteps as small as 10^{-15}. We present the results for our adaptive regularization strategy in Figure 16. In this case, as in the test of §6.4, we do not reset the regularization parameter on line line 7. This partially explains the large number of Newton steps are required for the first few timesteps as the regularization parameter is large at the beginning of the first step, but then is much lower from one step to the next. There are also some spikes starting at around 0.6 days that we may attribute to the saturation front arriving at the clay and creating the water table. Otherwise the number of iterations per step is quite reasonable (10.5 on average).

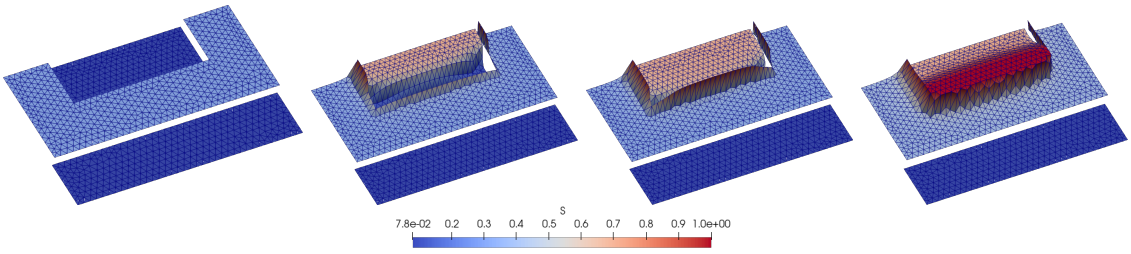

Figure 17: Evolution of the saturation for the perched test case of §6.5 for $t \in \{0\text{ s}, 21{\cdot}10^3\text{ s}, 41{\cdot}10^3\text{ s}, 86.1{\cdot}10^3\text{ s}\}$.

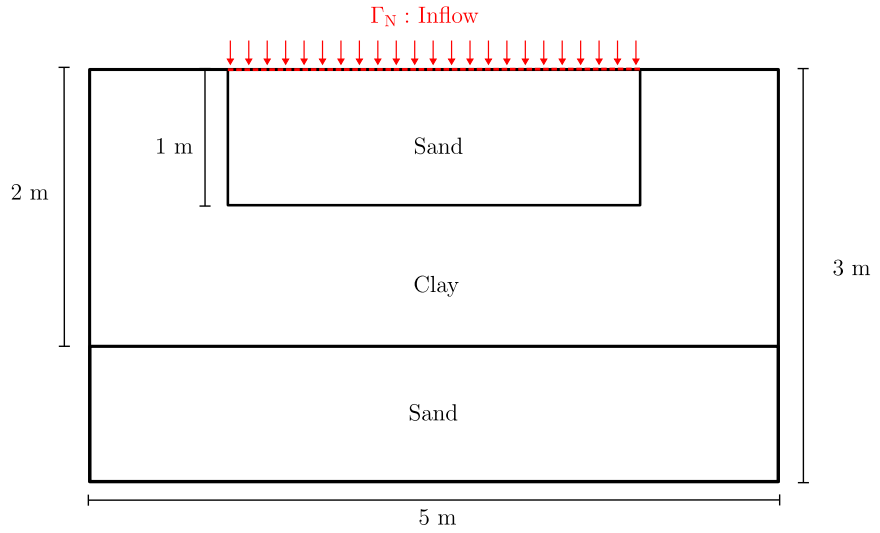

Figure 18: Domain and boundary conditions for the perched water table inspired by [24].

Figure 19:

7 Conclusions and future work

In this work, we introduced an adaptive regularization algorithm to iteratively solve the Richards equation. It works with regularized versions of the nonlinearities present in the Richards equation to improve the performance of Newton's method in solving the resulting nonlinear system. The proposed algorithm more precisely adaptively controls the level of regularization based on a posteriori error estimators. is able to converge where the unregularized version takes excessively many iterations, or does not converge at all. Furthermore, we numerically compared the performance with the standard, unregularized Newton with adaptive timestep cutting as well as modified Picard scheme, which is specific to Richards equation. In all test cases, the adaptive algorithm with regularization outperforms the modified Picard as well as the adaptive timestep cutting strategy and produces a perceptibly comparable solution.

In terms of future work, we note that our proposed algorithm is not able to converge in the dry regime $s \ll 1$. This is a well known difficulty and has been shown to be improved by variable switching techniques, see [7] and the references therein. We would like to that our strategy is not incompatible with these methods, and test a combination of regularization and variable switching to tackle even more difficult benchmark problems. Another future direction would be to study two independent regularization parameters for the functions κ_ϵ and S_ϵ in the case of the Brooks–Corey model.

8 Declarations

8.1 Funding

No funding was received to assist with the preparation of this manuscript.

8.2 Financial interests

The authors have no relevant financial or non-financial interests to disclose.

References

[1] Alt, H. W., and Luckhaus, S. Quasilinear elliptic-parabolic differential equations. *Math. Z. 183*, 3 (1983), 311–341.

[2] Badia, S., and Verdugo, F. Gridap: An extensible Finite Element toolbox in Julia. *J. Open Source Softw. 5*, 52 (2020), 2520.

[3] Bassetto, S. *Vers une prise en compte plus robuste et précise des effets capillaires lors de simulations d'écoulements multiphasiques en milieux poreux*. PhD thesis, Université de Lille, 2021.

[4] Bassetto, S., Cancès, C., Enchéry, G., and Tran, Q. H. Robust Newton solver based on variable switch for a finite volume discretization of Richards equation. In *Finite Volumes for Complex Applications IX—Methods, Theoretical Aspects, Examples—FVCA 9, Bergen, Norway, June 2020*, vol. 323 of *Springer Proc. Math. Stat.* Springer, Cham, 2020, pp. 385–393.

[5] Bassetto, S., Cancès, C., Enchéry, G., and Tran, Q.-H. On several numerical strategies to solve Richards' equation in heterogeneous media with finite volumes. *Comput. Geosci. 26*, 5 (2022), 1297–1322.

[6] Braess, D., and Schöberl, J. Equilibrated residual error estimator for edge elements. *Math. Comp. 77*, 262 (2008), 651–673.

[7] Brenner, K., and Cancès, C. Improving Newton's method performance by parametrization: The case of the Richards equation. *SIAM J. Numer. Anal. 55*, 4 (2017), 1760–1785.

[8] Brezzi, F., and Fortin, M., Eds. *Mixed and Hybrid Finite Element Methods*, vol. 15 of *Springer Series in Computational Mathematics*. Springer, New York, NY, 1991.

[9] Brooks, R. H., and Corey, A. T. Properties of Porous Media Affecting Fluid Flow. *J. Irrig. Drain. Div. 92*, 2 (1966), 61–88.

[10] Celia, M. A., Bouloutas, E. T., and Zarba, R. L. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resour. Res. 26*, 7 (1990), 1483–1496.

[11] Chen, Z., Huan, G., and Ma, Y. *Computational Methods for Multiphase Flows in Porous Media*. Society for Industrial and Applied Mathematics, 2006.

[12] De Boer, R. *Theory of Porous Media*. Springer, Berlin, Heidelberg, 2000.

[13] Destuynder, P., and Métivet, B. Explicit error bounds in a conforming finite element method. *Math. Comp. 68*, 228 (1999), 1379–1396.

[14] Deuflhard, P. *Newton Methods for Nonlinear Problems*, vol. 35 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2004.

[15] Diersch, H. J. G., and Perrochet, P. On the primary variable switching technique for simulating unsaturated–saturated flows. *Advances in Water Resources 23*, 3 (1999), 271–301.

[16] Ern, A., Nicaise, S., and Vohralík, M. An accurate H(div) flux reconstruction for discontinuous Galerkin approximations of elliptic problems. *C. R. Math. 345*, 12 (2007), 709–712.

[17] Ern, A., and Vohralík, M. Adaptive inexact Newton methods with a posteriori stopping criteria for nonlinear diffusion PDEs. *SIAM J. Sci. Comput. 35*, 4 (2013), A1761–A1791.

[18] Févotte, F. c., Rappaport, A., and Vohralík, M. Adaptive regularization, discretization, and linearization for nonsmooth problems based on primal-dual gap estimators. *Comput. Methods Appl. Mech. Engrg. 418* (2024), Paper No. 116558, 33.

[19] Forsyth, P. A., Wu, Y. S., and Pruess, K. Robust numerical methods for saturated-unsaturated flow with dry initial conditions in heterogeneous media. *Adv. Water Resour. 18*, 1 (1995), 25–38.

[20] Illiano, D., Pop, I. S., and Radu, F. A. Iterative schemes for surfactant transport in porous media. *Comput. Geosci. 25*, 2 (2021), 805–822.

[21] Jäger, W., and Kačur, J. Solution of porous medium type systems by linear approximation schemes. *Numer. Math. 60*, 1 (1991), 407–427.

[22] Jäger, W., and Kačur, J. Solution of doubly nonlinear and degenerate parabolic problems by relaxation schemes. *ESAIM: M2AN 29*, 5 (1995), 605–627.

[23] Jenny, P., Tchelepi, H. A., and Lee, S. H. Unconditionally convergent nonlinear solver for hyperbolic conservation laws with S-shaped flux functions. *J. Comput. Phys. 228*, 20 (2009), 7497–7512.

[24] Kirkland, M. R., Hills, R. G., and Wierenga, P. J. Algorithms for solving Richards' equation for variably saturated soils. *Water Resources Research 28*, 8 (1992), 2049–2058.

[25] Kräutle, S. The semismooth Newton method for multicomponent reactive transport with minerals. *Water Res. 34*, 1 (2011), 137–151.

[26] Ladevèze, P., and Leguillon, D. Error estimate procedure in the finite element method and applications. *SIAM J. Numer. Anal. 20*, 3 (1983), 485–509.

[27] Lehmann, F., and Ackerer, Ph. Comparison of iterative methods for improved solutions of the fluid flow equation in partially saturated porous media. *Transport in Porous Media 31*, 3 (1998), 275–292.

[28] List, F., and Radu, F. A. A study on iterative methods for solving Richards' equation. *Comput. Geosci. 20*, 2 (2016), 341–353.

[29] Mitra, K., and Pop, I. S. A modified L-scheme to solve nonlinear diffusion problems. *Comp. & Math. Appl. 77*, 6 (2019), 1722–1738.

[30] Mitra, K., and Vohralík, M. A posteriori error estimates for the Richards equation. *Math. Comp. 93*, 347 (2024), 1053–1096.

[31] Pop, I. S., Radu, F., and Knabner, P. Mixed finite elements for the Richards' equation: Linearization procedure. *J. Comput. Appl. Math. 168*, 1-2 (2004), 365–373.

[32] Pop, I. S., and Schweizer, B. Regularization schemes for degenerate Richards equations and outflow conditions. *Math. Models Methods Appl. Sci. 21*, 08 (2011), 1685–1712.

[33] Prager, W., and Synge, J. L. Approximations in elasticity based on the concept of function space. *Quart. Appl. Math. 5*, 3 (1947), 241–269.

[34] Qi, L., and Sun, J. A nonsmooth version of Newton's method. *Math. Prog. 58*, 1 (1993), 353–367.

[35] Radu, F. A., Pop, I. S., and Knabner, P. Newton-type methods for the mixed finite element discretization of some degenerate parabolic equations. In *Numer. Math. Adv. Appl.* (Berlin, Heidelberg, 2006), A. B. de Castro, D. Gómez, P. Quintela, and P. Salgado, Eds., Springer, pp. 1192–1200.

[36] Slodička, M. A robust and efficient linearization scheme for doubly nonlinear and degenerate parabolic problems arising in flow in porous media. *SIAM J. Sci. Comput. 23*, 5 (2002), 1593–1614.

[37] Stokke, J. S., Mitra, K., Storvik, E., Both, J. W., and Radu, F. A. An adaptive solution strategy for richards' equation. *Computers & Mathematics with Applications 152* (2023), 155–167.

[38] van Genuchten, M. T. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Sci. Soc. Am. J. 44*, 5 (1980), 892–898.

[39] Verdugo, F., and Badia, S. The software design of Gridap: A Finite Element package based on the Julia JIT compiler. *Comp. Phys. Commun. 276* (2022), 108341.

[40] Vlasák, M. On polynomial robustness of flux reconstructions. *Appl. Math. 65*, 2 (2020), 153–172.

[41] Wang, X., and Tchelepi, H. A. Trust-region based solver for nonlinear transport in heterogeneous porous media. *J. Comput. Phys. 253* (2013), 114–137.