An adaptive regularization strategy for efficiently solving the Richards equation

François Févotte[1] Ari Rappaport[4] Martin Vohralík[2,3]

[1]Triscale Innov, 7 rue de la Croix Martre, 91120 Palaiseau, France

[2]Inria, 48 rue Barrault, Paris 75647, France

[3]Université Paris-Est, CERMICS (ENPC), 77455 Marne-la-Vallée, France

[4]ENSTA Paris, 828, Boulevard des Maréchaux, 91762 Palaiseau Cedex

Journée Inria/IFPEN, December 9, 2024

Outline

Richards equation: flow in unsaturated porous media

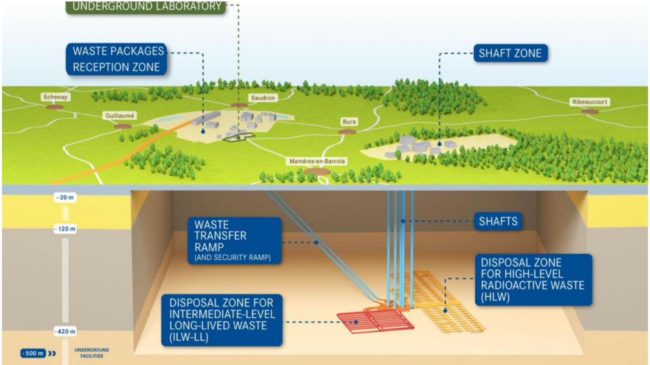


Figure: CIGÉO facility[1]

- Porous medium is a material containing **pores** (small regular voids)
- Safety certification of nuclear waste storage (flow of contaminants)

Motivation: PDE models are **highly nonlinear** and **nonsmooth** $\Longrightarrow$ difficult to solve numerically

[1] Image courtesy of andra.fr

Richards equation: derivation and data

Conservation of Mass (water)

$$\phi \partial_t s + \nabla \cdot \boldsymbol{q} = f(\mathbf{x}, t)$$

- Water saturation s
- ϕ porosity, f external source
- $\boldsymbol{q}$ so-called Darcy flux

Darcy's Law for Flow

$$\boldsymbol{q} = -\boldsymbol{K}\kappa(s)(\nabla p + \boldsymbol{g})$$

- Fluid pressure p
- $\boldsymbol{K}$ absolute permeability tensor, κ relative permeability, $\boldsymbol{g}$ gravity

Putting it all together

Find a pressure p and saturation s such that

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

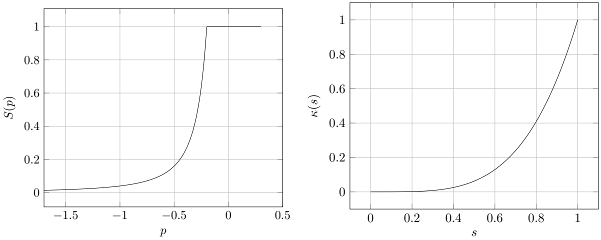


Figure: Brooks–Corey constituitive laws

- Capillary pressure relation: $s = S(p)$
- Choose "pressure formulation" p: always defined

Putting it all together

Find a pressure p and saturation s such that

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

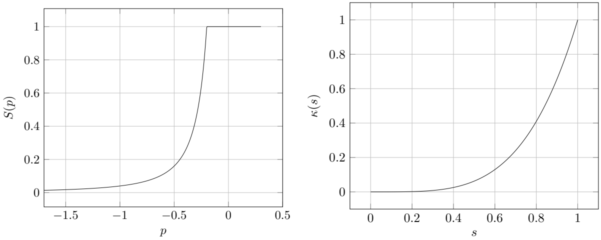


Figure: Brooks–Corey constituitive laws

- Capillary pressure relation: $s = S(p)$
- Choose “pressure formulation” p: always defined

Degenerancies and low differentiability

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

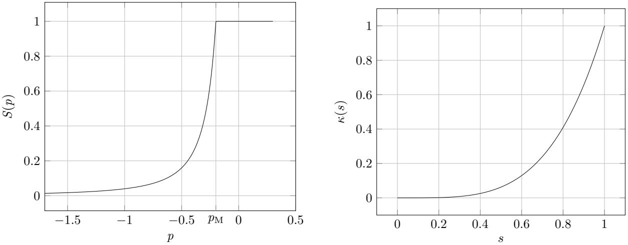


Figure: Brooks–Corey constitutive laws

- Elliptic: $\partial_t s = 0$
- Hyperbolic (ODE): $\kappa(s) = 0$
- Kink at $p = p_M$ for Brooks–Corey constitutive law

Degeneracies and low differentiability

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

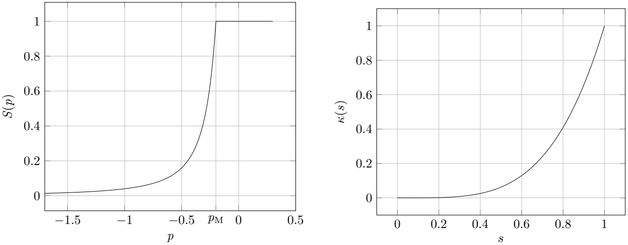


Figure: Brooks–Corey constitutive laws

- Elliptic: $\partial_t s = 0$
- Hyperbolic (ODE): $\kappa(s) = 0$
- Kink at $p = p_{\mathrm{M}}$ for Brooks–Corey constitutive law

Degeneracies and low differentiability

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

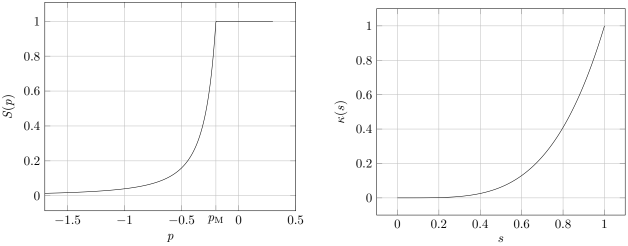
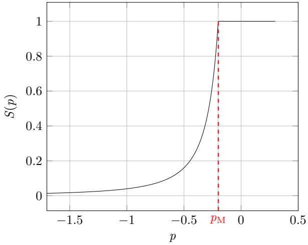


Figure: Brooks–Corey constitutive laws

- Elliptic: $\partial_t s = 0$
- Hyperbolic (ODE): $\kappa(s) = 0$
- Kink at $p = p_{\mathrm{M}}$ for Brooks–Corey constitutive law

Degeneracies and low differentiability

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



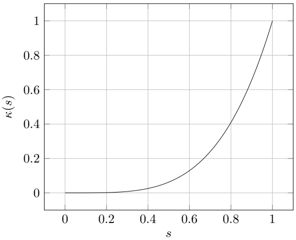


Figure: Brooks–Corey constitutive laws

- Elliptic: $\partial_t s = 0$
- Hyperbolic (ODE): $\kappa(s) = 0$
- Kink at $p = p_{\mathrm{M}}$ for Brooks–Corey constitutive law

State of the art

Variable switching

- Forsyth, P. A., Y. S. Wu, and K. Pruess (1995). "Robust Numerical Methods for Saturated-Unsaturated Flow with Dry Initial Conditions in Heterogeneous Media". In: *Adv. Water Resour.* 18.1, pp. 25–38.
- Diersch, H. J. G. and P. Perrochet (1999). "On the Primary Variable Switching Technique for Simulating Unsaturated–Saturated Flows". In: *Adv. Water Res.* 23.3, pp. 271–301.
- Brenner, K. and C. Cancès (2017). "Improving Newton's method performance by parametrization: the case of the Richards equation". In: *SIAM J. Numer. Anal.* 55.4, pp. 1760–1785.
- Bassetto, S., C. Cancès, G. Enchéry, and Q. H. Tran (2020). "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. Springer Proc. Math. Stat. Springer, Cham, pp. 385–393.

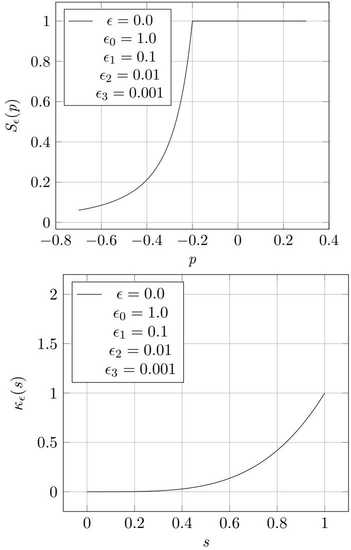
Subquadratic schemes

- Celia, M. A., E. T. Bouloutas, and R. L. Zarba (1990). "A General Mass-Conservative Numerical Solution for the Unsaturated Flow Equation". In: *Water Resour. Res.* 26.7, pp. 1483–1496.
- Pop, I. S., F. Radu, and P. Knabner (2004). "Mixed Finite Elements for the Richards' Equation: Linearization Procedure". In: *J. Comput. Appl. Math.* 168.1-2, pp. 365–373.
- Mitra, K. and I. S. Pop (2019). "A Modified L-scheme to Solve Nonlinear Diffusion Problems". In: *Comp. & Math. Appl.* 7th International Conference on Advanced Computational Methods in Engineering (ACOMEN 2017) 77.6, pp. 1722–1738.
- Stokke, J. S., K. Mitra, E. Storvik, J. W. Both, and F. A. Radu (2023). "An adaptive solution strategy for Richards' equation". In: *Comput. Math. Appl.* 152, pp. 155–167.

Outline

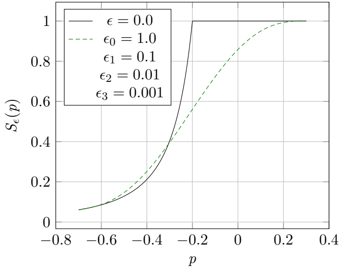
Regularized sequence

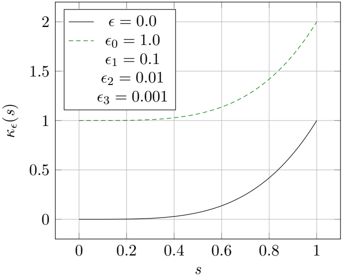
Use solution with ϵ_j as initial guess for solving ϵ_{j+1}



Regularized sequence

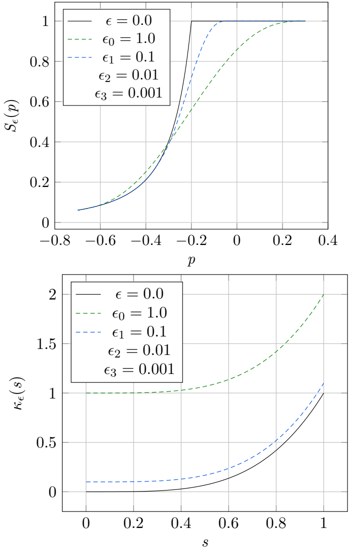
Use solution with ϵ_j as initial guess for solving ϵ_{j+1}





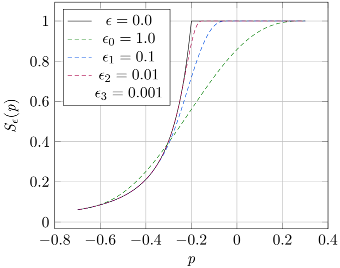
Regularized sequence

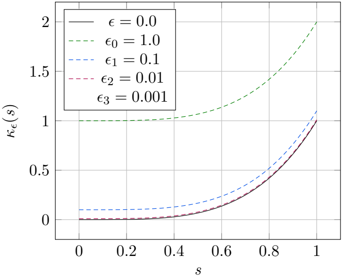
Use solution with ϵ_j as initial guess for solving ϵ_{j+1}



Regularized sequence

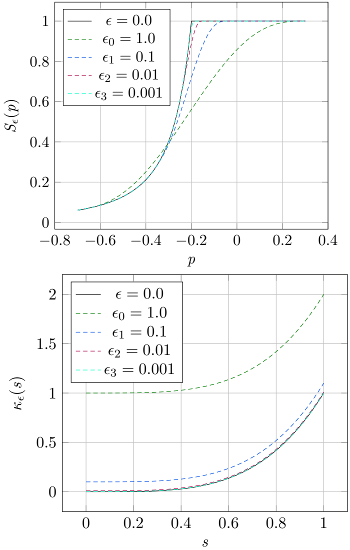
Use solution with ϵ_j as initial guess for solving ϵ_{j+1}





Regularized sequence

Use solution with ϵ_j as initial guess for solving ϵ_{j+1}



Discretization

Method of lines

- Mesh $\mathcal{T}_h$ of Ω, fixed conforming $\mathcal{P}^1$-FEM in space

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

- backward Euler in time: uniform time step $\tau = 1/NT$, for each $n \in \{1, \ldots, N\}$ and a given $p_{n-1,h} \in V_h$, $p_{n,h} \in V_h$ satisfying

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

The *flux function* is defined as

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

Regularized and linearized problems

Regularized problem (index j)

$$\frac{1}{\tau}(S_{\epsilon_j}(p^j_{n,h}) - S_{\epsilon_j}(p^{\bar{j}}_{n-1,h}), \varphi_h) + (\boldsymbol{F}_{\epsilon_j}(p^j_{n,h}), \nabla\varphi_h) = (f(\cdot, t_n), \varphi_h) + (q_{\mathrm{N}}, \varphi_h)_{\Gamma_{\mathrm{N}}} \quad \forall \varphi_h \in V_h,$$

- $\boldsymbol{F}_{\epsilon_j}(q) := \boldsymbol{K}\kappa_{\epsilon_j}(S_{\epsilon_j}(q))[\nabla q + \boldsymbol{g}]$.

Regularized/linearized problem (index k)

$$\frac{1}{\tau}(\phi S_{\epsilon_j}(p^{j,k-1}_{n,h}) - S_{\epsilon_j}(p^{\bar{j},\bar{k}}_{n-1,h}), \varphi_h) + \frac{1}{\tau}(\phi L(p^{j,k}_{n,h} - p^{j,k-1}_{n,h}), \varphi_h) + (\boldsymbol{F}^k_{\epsilon_j}, \nabla\varphi_h) + (q_{\mathrm{N}}, \varphi_h)_{\Gamma_{\mathrm{N}}} = (f(\cdot, t_n), \varphi_h) \quad \forall \varphi_h \in V_h,$$

- $\boldsymbol{F}^k_{\epsilon_j} := \boldsymbol{K}\kappa_{\epsilon_j}(S_{\epsilon_j}(p^{j,k-1}_{n,h}))[\nabla p^{j,k}_{n,h} + \boldsymbol{g}] + \boldsymbol{\xi}(p^{j,k}_{n,h} - p^{j,k-1}_{n,h})$
- $(L, \boldsymbol{\xi}) \in \mathbf{L}^\infty(\Omega; \mathbb{R}^{d+1})$ depend on the specific linearization used.

A posteriori error estimators

Averaging in $\boldsymbol{H}(\mathrm{div}, \Omega)$

- Lowest order Raviart-Thomas space
  $\boldsymbol{RT}_0(\mathcal{T}_h) := \{\boldsymbol{v}_h \in [L^2(\Omega)]^d : \boldsymbol{v}_h|_K \in [\mathcal{P}(K)]^d + \boldsymbol{x}\mathcal{P}_0(K), \forall K \in \mathcal{T}_h\}$
- Reconstruction $\boldsymbol{\sigma}_{n,h}^{j,k} \in \boldsymbol{RT}_0(\mathcal{T}_h) \cap \boldsymbol{H}(\mathrm{div}, \Omega)$ of $-\boldsymbol{F}_{\epsilon_j}^k$ based on averaging with connection to *equilibrated* flux [Vlasák 2020; Ern, Nicaise, and Vohralík 2007]

Component estimators

For an approximate solution $p_{n,h}^{j,k}$,

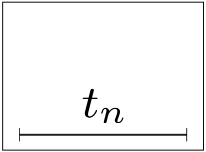
$$\eta_{\mathrm{dis}}^{\ell,j,k} := \|\boldsymbol{F}_{\epsilon_j}^k + \boldsymbol{\sigma}_{n,h}^{j,k}\| \qquad \text{(discretization)}$$

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

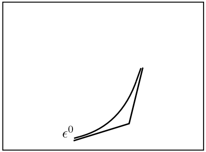
$$\eta_{\mathrm{reg}}^{\ell,j,k} := \|\boldsymbol{F}(p_{n,h}^{j,k}) - \boldsymbol{F}_{\epsilon_j}(p_{n,h}^{j,k})\| \qquad \text{(regularization)}$$

Adaptive algorithm for Richards based on a posteriori error estimators

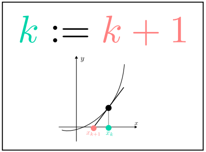
Timestepping



Regularization

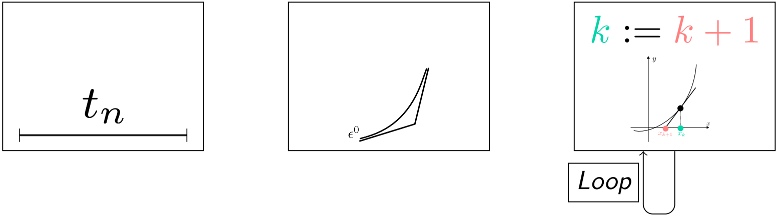


Linearization



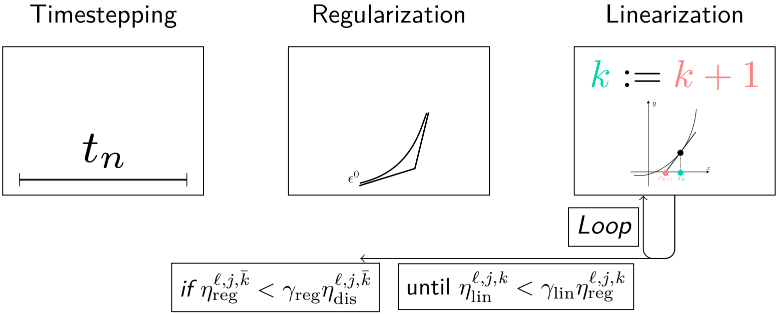
Adaptive algorithm for Richards based on a posteriori error estimators

Timestepping | Regularization | Linearization

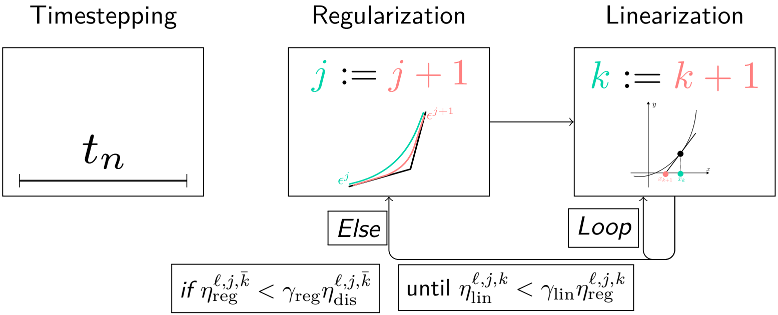


until $\eta_{\text{lin}}^{\ell,j,k} < \gamma_{\text{lin}} \eta_{\text{reg}}^{\ell,j,k}$

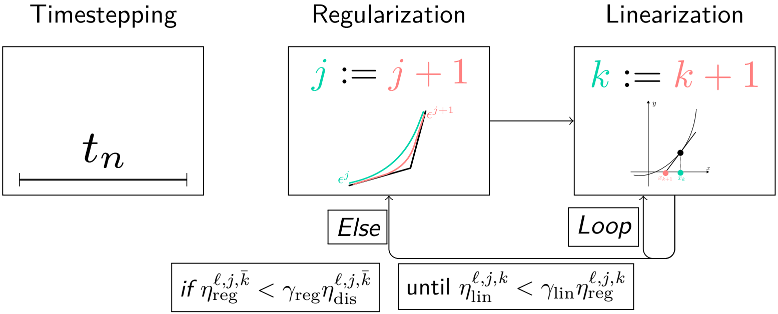
Adaptive algorithm for Richards based on a posteriori error estimators



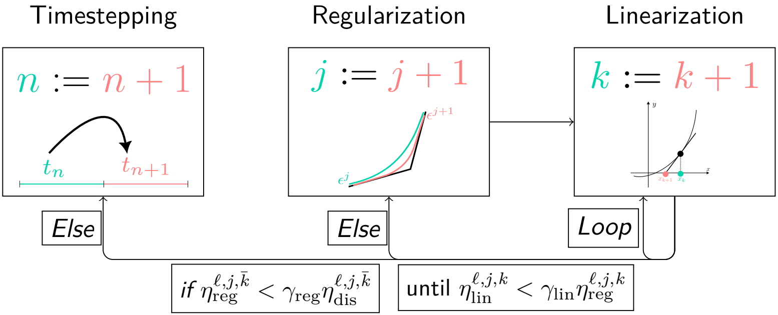
Adaptive algorithm for Richards based on a posteriori error estimators



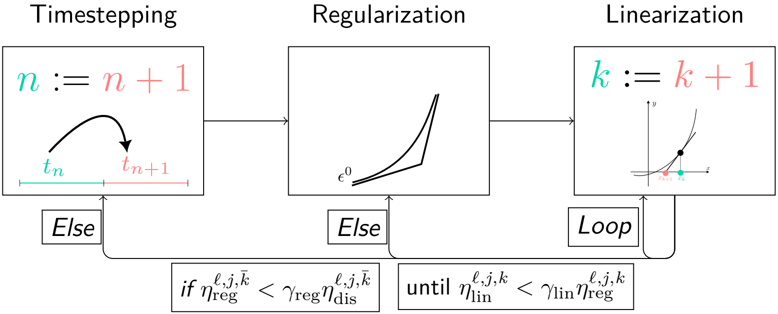
Adaptive algorithm for Richards based on a posteriori error estimators



Adaptive algorithm for Richards based on a posteriori error estimators

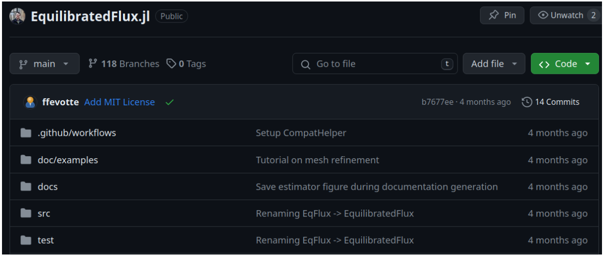


Adaptive algorithm for Richards based on a posteriori error estimators



Outline

Free and open source library




```julia
𝐅 = 𝐊 ⋅ ((κ0 ∘ Sₙ0) * ∇(pₙᵏ⁻¹) + (κ0 ∘ Sₙ0) * g)
𝐅ⱼ = 𝐊 ⋅ ((κ ∘ Sₙ) * ∇(pₙᵏ⁻¹) + (κ ∘ Sₙ) * g)
(ξ, _) = get_L_and_ξ(linearization, data)
𝐅ⱼᵏ⁻¹ = 𝐊 ⋅ ((κ ∘ Sₙ) * ∇(pₙᵏ) + (κ ∘ Sₙ) * g) + ξ*(pₙᵏ - pₙᵏ⁻¹)
σₙ = EquilibratedFlux.build_averaged_flux(-𝐅ⱼᵏ⁻¹, model)
```

Methods tested for comparision

Linearizations

- Newton's method:
  $L := S'_{\epsilon_j}(p_{n,h}^{j,k-1}), \quad \boldsymbol{\xi} := \boldsymbol{K}(\kappa_{\epsilon_j} \circ S_{\epsilon_j})'(p_{n,h}^{j,k-1})[\nabla p_{n,h}^{j,k-1} + \boldsymbol{g}]$
- modified Picard [Celia, Bouloutas, and Zarba 1990]:
  $L := S'_{\epsilon_j}(p_{n,h}^{j,k-1}), \quad \boldsymbol{\xi} := \boldsymbol{0}$

Timestepping/regularization

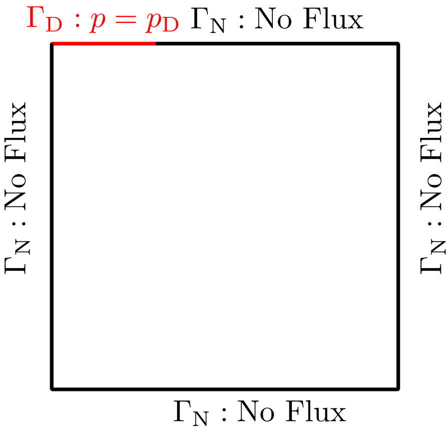
For Newton's method we consider

- No regularization and simple timestep cutting algorithm
- No regularization and uniform timestepping
- With regularization and uniform timestepping

For modified Picard only uniform timestepping and no regularization

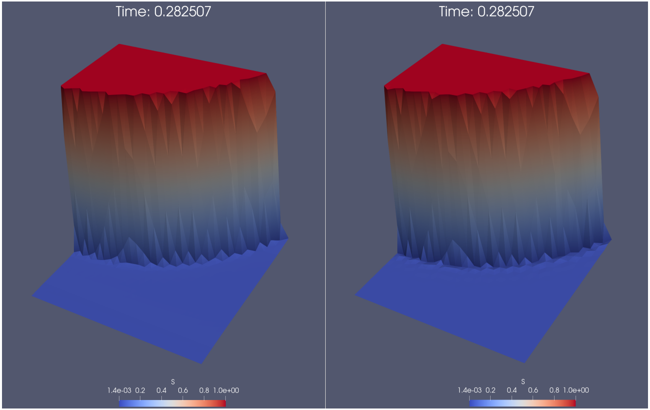
Injection test: setup

- $\Omega = (0,1)^2$
- $T = 1.0$
- $\tau = 2.82 \cdot 10^{-2}$
- Quasi uniform mesh with $h = 2.82 \cdot 10^{-2}$
- $\Gamma_{\mathrm{D}} = \{(x_1, x_2) | x_1 \in (0, 0.3), x_2 = 1\}$
- $\Gamma_{\mathrm{N}} = \partial\Omega \setminus \Gamma_{\mathrm{D}}$
- $\boldsymbol{g} = (0, -1)^T$
- $f = 0$
- $p_0 = -1, \quad s_0 = S(p_0)$
- $p_{\mathrm{D}} = 1$



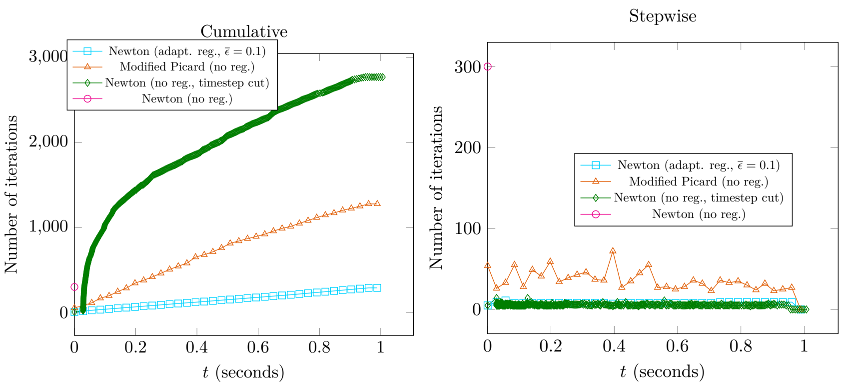
Inspired by test case in [Brenner and Cancès 2017]

Injection test: saturation comparision



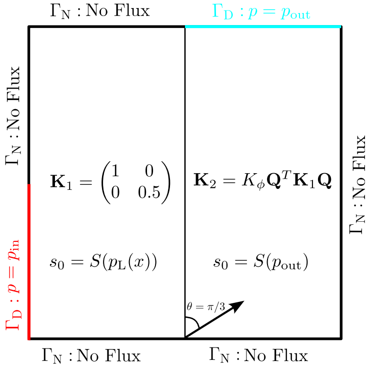
- With (left) and without (right) regularization

Injection test: performance



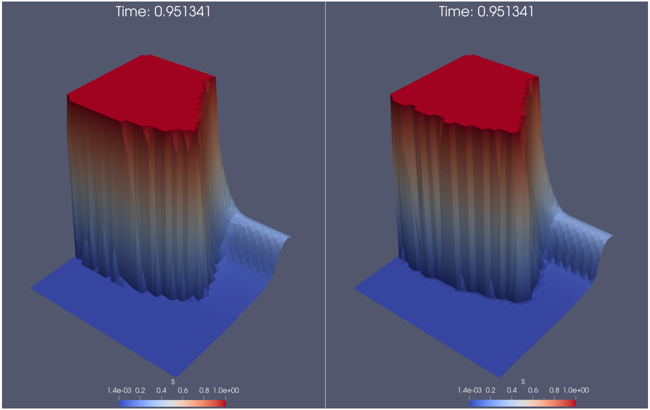
Realistic test: setup

- $\Omega = (0,1)^2$
- quasi uniform mesh with $h = 2.02 \cdot 10^{-2}$
- $T = 1$
- $\tau_0 = 2.02 \cdot 10^{-2}$
- $\boldsymbol{g} = (-1, 0)^T$
- $\mathbf{Q} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$
- $K_\phi = 0.1$
- $p_{\mathrm{L}}(\boldsymbol{x}) = \left(\frac{p_{\mathrm{out}} - p_{\mathrm{in}}}{0.5}\right) \boldsymbol{x}$
- $p_{\mathrm{out}} = -2.0$
- $p_{\mathrm{in}} = -0.2$
- $\phi = 1$



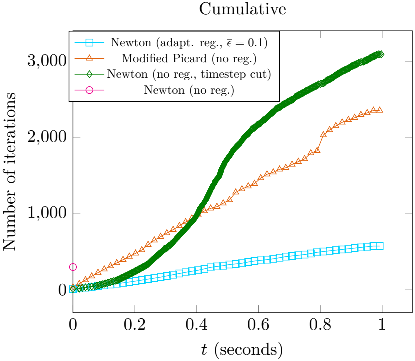
Inspired by test case in [Mitra and Vohralík 2024]

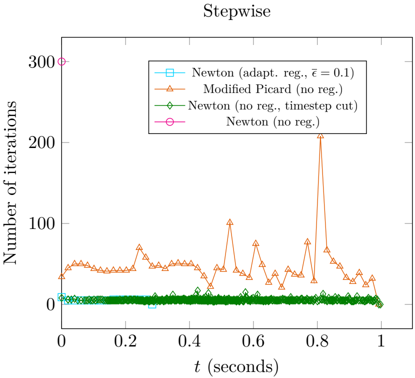
Realistic test: saturation comparison



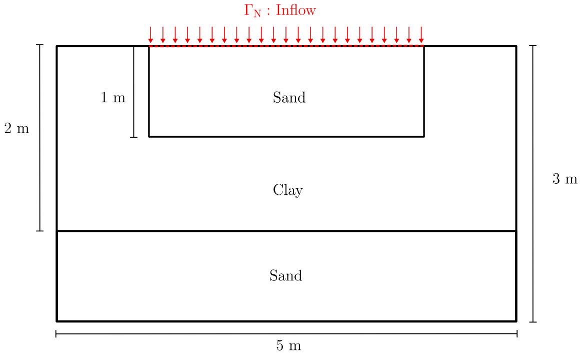
- With (left) and without (right) regularization

Realistic test: performance





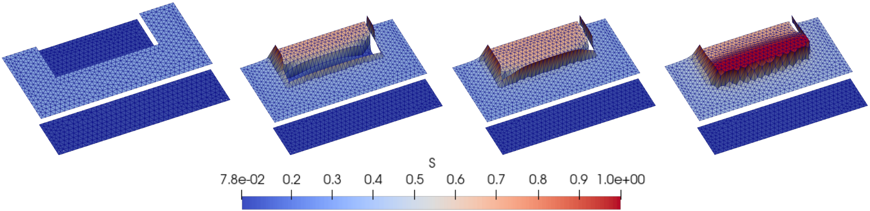
Perched water table: setup



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

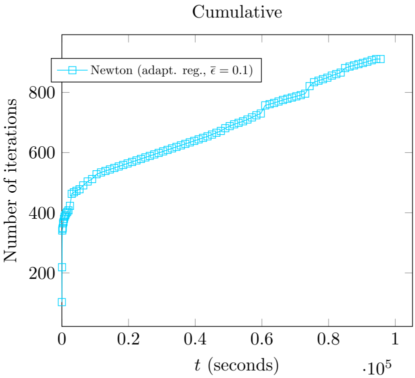
- Adapted from [Kirkland, Hills, and Wierenga 1992]

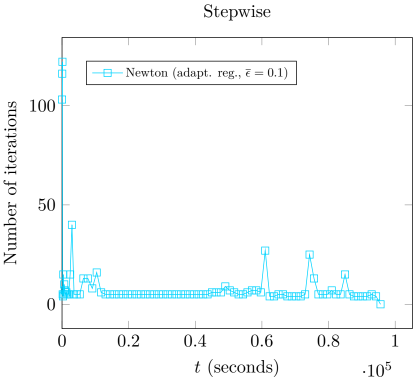
Perched water table: saturation profile



- Evolution of the saturation at time $t \in \{0\ \text{s}, 21 \cdot 10^3\ \text{s}, 41 \cdot 10^3\ \text{s}, 86.1 \cdot 10^3\ \text{s} = 1\ \text{day}\}$.

Perched water table: performance





Outline

Summary

- Introduce regularization of common constitutive laws for Richards equation
- Error estimation based on flux reconstruction
- Adaptive algorithm based on balancing error components
- Tested on benchmark problems from the literature

Févotte, F., Rappaport, A., and Vohralík, M. *Adaptive regularization for the Richards equation*. Comput. Geosci. (2024).

Perspectives

- Combine with existing techniques like variable switching
- Extension to two phase flow (variational inequalities)

Thank you for your attention!

Summary

- Introduce regularization of common constitutive laws for Richards equation
- Error estimation based on flux reconstruction
- Adaptive algorithm based on balancing error components
- Tested on benchmark problems from the literature

Févotte, F., Rappaport, A., and Vohralík, M. *Adaptive regularization for the Richards equation*. Comput. Geosci. (2024).

Perspectives

- Combine with existing techniques like variable switching
- Extension to two phase flow (variational inequalities)

Thank you for your attention!

Summary

- Introduce regularization of common constitutive laws for Richards equation
- Error estimation based on flux reconstruction
- Adaptive algorithm based on balancing error components
- Tested on benchmark problems from the literature

Févotte, F., Rappaport, A., and Vohralík, M. *Adaptive regularization for the Richards equation*. Comput. Geosci. (2024).

Perspectives

- Combine with existing techniques like variable switching
- Extension to two phase flow (variational inequalities)

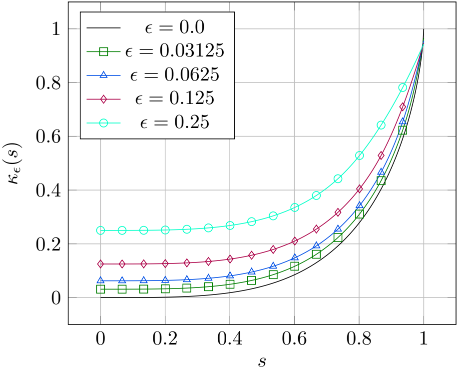
Thank you for your attention!

Regularization of the relative permeability for Van-Genuchten

Follow the approach in [Bassetto, Cancès, Enchéry, and Tran 2020] where the relative permeability κ is replaced by a second degree polynomial near the critical point $s = 1$:

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

$$\tilde{\kappa}(s) = \frac{\kappa''(1-\epsilon)}{2}(s - (1-\epsilon))^2 + \kappa'(1-\epsilon)(s - (1-\epsilon)) + \kappa(1-\epsilon),$$



Hermite interpolation for the Brooks–Corey Saturation

$$
\begin{aligned}
S_\epsilon(p_{\mathrm{M}} - \epsilon) &= S(p_{\mathrm{M}} - \epsilon), & S_\epsilon(p_{\mathrm{M}} + \epsilon) &= S(p_{\mathrm{M}} + \epsilon) \\
S'_\epsilon(p_{\mathrm{M}} - \epsilon) &= S'(p_{\mathrm{M}} - \epsilon), & S'_\epsilon(p_{\mathrm{M}} + \epsilon) &= S'(p_{\mathrm{M}} + \epsilon), \\
&\vdots & &\vdots \\
S^{(r)}_\epsilon(p_{\mathrm{M}} - \epsilon) &= S^{(r)}(p_{\mathrm{M}} - \epsilon), & S^{(r)}_\epsilon(p_{\mathrm{M}} + \epsilon) &= S^{(r)}(p_{\mathrm{M}} + \epsilon).
\end{aligned}
$$