Robust Numerical Solution of Richards' Equation Under Dry Conditions

Konstantin Brenner, **Roland Masson**

J.A. Dieudonné laboratory   Inria & Univ. Côte d'Azur

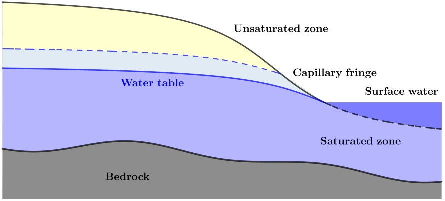
CMWR 2022

Gdańsk, June 20, 2022

Introduction

Unsaturated/saturated groundwater flow

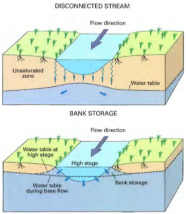
$$\phi \partial_t s - \operatorname{div}\left(\frac{\mathbb{K}k(s)}{\mu}(\nabla p - \rho \mathbf{g})\right) = 0, \qquad s = S(p)$$



Michel et al. '19

Applications:

- Water resource estimation
- Irrigation
- Contaminant transport
- Interaction with surface water
- ...

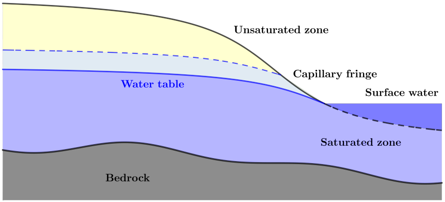


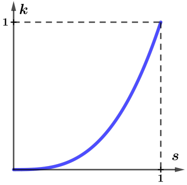
Reilly '01

Introduction

Unsaturated/saturated groundwater flow

$$\phi\partial_t s - \operatorname{div}\left(\frac{\mathbb{K}k(s)}{\mu}(\nabla p - \rho \mathbf{g})\right) = 0, \qquad s = S(p)$$

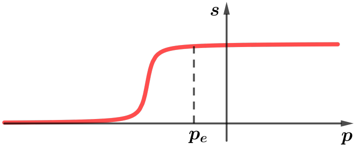




Relative permeability k as a function of saturation

Applications:

- Water resource estimation
- Irrigation
- Contaminant transport
- Interaction with surface water
- ...

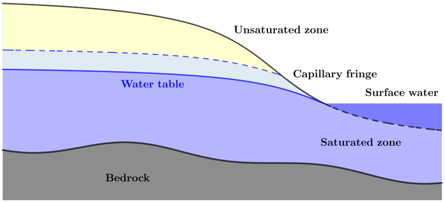


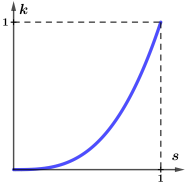
Retention curve $s = S(p)$

Introduction

Unsaturated/saturated groundwater flow

$$\phi \partial_t s - \operatorname{div}\left(\frac{\mathbb{K}k(s)}{\mu}(\nabla p - \rho \mathbf{g})\right) = 0, \qquad s = S(p)$$

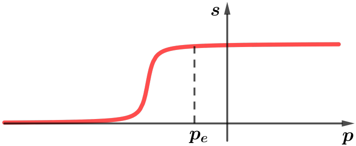




Relative permeability k as a function of saturation

Substantially nonlinear PDE

- Multiple "regimes"
- Numerically challenging: dry-wet transition; heterogeneous soils; **data-dependent performance**

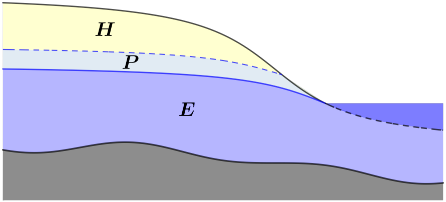


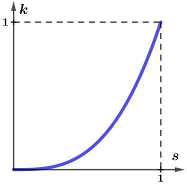
Retention curve $s = S(p)$

Introduction

Unsaturated/saturated groundwater flow

$$\phi \partial_t s - \operatorname{div}\left(\frac{\mathbb{K} k(s)}{\mu} (\nabla p - \rho \mathbf{g})\right) = 0, \qquad s = S(p)$$

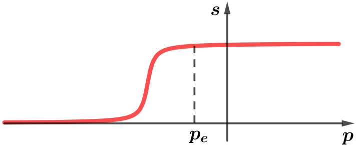




Relative permeability k as a function of saturation

Substantially nonlinear PDE

- Multiple “regimes”
- Numerically challenging: dry-wet transition; heterogeneous soils; **data-dependent performance**

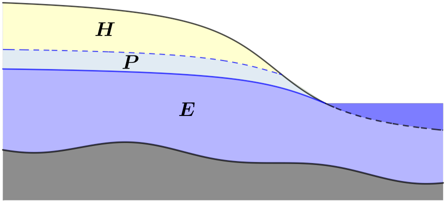


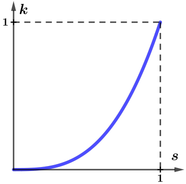
Retention curve $s = S(p)$

Introduction

Unsaturated/saturated groundwater flow

$$\phi \partial_t s - \operatorname{div}\left(\frac{\mathbb{K} k(s)}{\mu}(\nabla p - \rho \mathbf{g})\right) = 0, \qquad s = S(p)$$

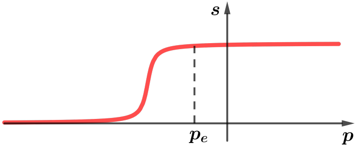




Relative permeability k as a function of saturation

Substantially nonlinear PDE

- Multiple "regimes"
- Numerically challenging:
  dry-wet transition;
  heterogeneous soils;
  data-dependent performance

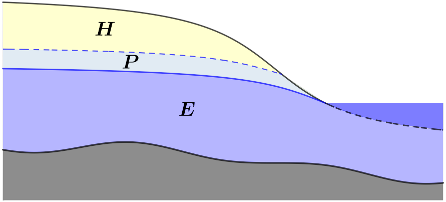


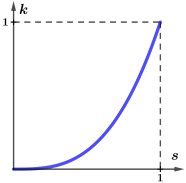
Retention curve $s = S(p)$

Introduction

Unsaturated/saturated groundwater flow

$$\partial_t s - \operatorname{div}\left(k(s)\nabla p\right) = 0, \qquad s = S(p)$$

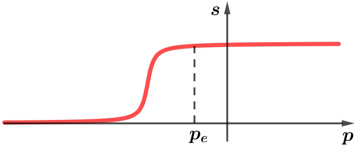




Relative permeability k as a function of saturation

Substantially nonlinear PDE

- Multiple "regimes"
- Numerically challenging:
  dry-wet transition;
  heterogeneous soils;
  data-dependent performance



Retention curve $s = S(p)$

Nonlinear convergence

PDE	
↓	mesh, discretization
Nonlinear algebraic system	
↓	linearization scheme
Linear system	
↓	direct or iterative solver
Discrete solution	

Nonlinear convergence

PDE	
↓	mesh, discretization
Nonlinear algebraic system	
↓	linearization scheme
Linear system	
↓	direct or iterative solver
Discrete solution	

Lack of robustness/efficiency of the nonlinear solver

- Unacceptably slow convergence
- Failure

Nonlinear convergence

PDE
$\downarrow$ mesh, discretization
Nonlinear algebraic system
$\downarrow$ linearization scheme
Linear system
$\downarrow$ direct or iterative solver
Discrete solution

Lack of robustness/efficiency of the nonlinear solver

- Unacceptably slow convergence
- Failure

Factors affecting the nonlinear convergence

- PDE formulation
- Mesh and discretization
- Linearization scheme: Newton, Picard, L-schemes, Quasi-Newton

  Globalization methods: backtracking, trust region, ...
- Nonlinear Preconditioning:

  Point-wise: variable substitution

  Domain Decomposition: Schwarz, Schur, Substructuring

Nonlinear convergence

PDE	
↓	mesh, discretization
Nonlinear algebraic system	
↓	linearization scheme
Linear system	
↓	direct or iterative solver
Discrete solution	

Lack of robustness/efficiency of the nonlinear solver

- Unacceptably slow convergence
- Failure

Factors affecting the nonlinear convergence

- **PDE formulation**
- Mesh and discretization
- Linearization scheme: **Newton**, Picard, L-schemes, Quasi-Newton

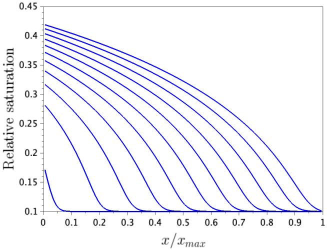
  Globalization methods: backtracking, trust region, . . .
- **Nonlinear Preconditioning**:

  Point-wise: variable substitution

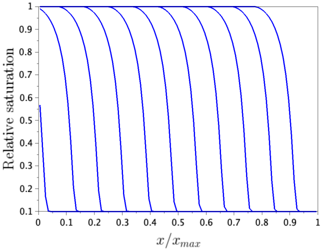
  Domain Decomposition: Schwarz, Schur, Substructuring

Test case: 1D water injection into a dry soil

Saturation profiles



Unsaturated flow

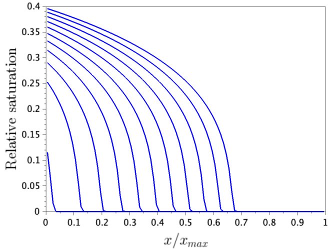


Pressurized flow

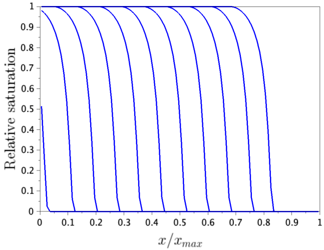
- Soil properties: Loam with Van Genuchten *kr*-*Pc* relation
- Input: Dry initial conditions $s_0 \in [10^{-6}, \mathbf{10^{-1}}]$
- Output: Newton iterations over time

Test case: 1D water injection into a dry soil

Saturation profiles



Unsaturated flow



Pressurized flow

- Soil properties: Loam with Van Genuchten *kr*-*Pc* relation
- Input: Dry initial conditions $s_0 \in [\mathbf{10^{-6}}, 10^{-1}]$
- Output: Newton iterations over time

Outline

Outline

Primary variable selection and (conservative) formulations

Original problem

$$\partial_t s - \operatorname{div}\left(k(s)\nabla p\right) = 0, \qquad s = S(p)$$

Major options: eliminate, reformulate.

Primary variable selection and (conservative) formulations

Original problem

$$\partial_t s - \mathrm{div}\left(k(s)\nabla p\right) = 0, \qquad \mathbf{s = S(p)}$$

Major options: **eliminate**, reformulate.

Nonlinear *p*-formulation

$$\partial_t S(p) - \mathrm{div}\left(k(S(p))\nabla p\right) = 0$$

Primary variable selection and (conservative) formulations

Original problem

$$\partial_t s - \mathrm{div}\,(\mathbf{k(s)}\nabla \mathbf{p}) = 0, \qquad s = S(p)$$

Major options: **eliminate**, reformulate.

Nonlinear p-formulation

$$\partial_t S(p) - \mathrm{div}\,(k(S(p))\nabla p) = 0$$

Nonlinear s-formulation

$$\partial_t s - \mathrm{div}\,(k(s)\nabla P(s)) = 0$$

Primary variable selection and (conservative) formulations

Original problem

$$\partial_t s - \mathrm{div}\,(\mathbf{k(s)}\nabla \mathbf{p}) = 0, \qquad s = S(p)$$

Major options: eliminate, **reformulate**.

Nonlinear p-formulation

$$\partial_t S(p) - \mathrm{div}\,(k(S(p))\nabla p) = 0$$

Nonlinear s-formulation

$$\partial_t s - \mathrm{div}\,(k(s)\nabla P(s)) = 0$$

Linearized s-formulation

$$\partial_t s - \mathrm{div}\,\left(k(s)P'(s)\nabla s\right) = 0$$

Primary variable selection and (conservative) formulations

Original problem

$$\partial_t s - \operatorname{div}\left(\mathbf{k}(\mathbf{s})\nabla \mathbf{p}\right) = 0, \qquad s = S(p)$$

Major options: eliminate, **reformulate**.

Nonlinear p-formulation

$$\partial_t S(p) - \operatorname{div}\left(k(S(p))\nabla p\right) = 0$$

Nonlinear s-formulation

$$\partial_t s - \operatorname{div}\left(k(s)\nabla P(s)\right) = 0$$

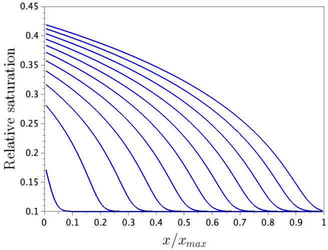
Linearized s-formulation

$$\partial_t s - \operatorname{div}\left(k(s)P'(s)\nabla s\right) = 0$$

	Variably sat.	Heterogeneities	Performance
Nonlin. p	Okay	Okay	
Nonlin. s	Unsat	Okay	
Lin. s	Unsat	Not trivial	

Test case: 1D water injection into a dry soil

Saturation profiles

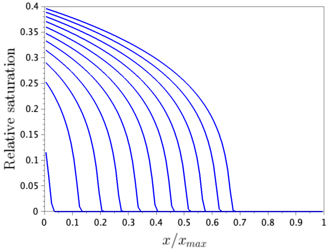


Unsaturated flow

- Soil properties: Loam with Van Genuchten *kr*-*Pc* relation
- Input: Dry initial conditions $s_0 \in [10^{-6}, \mathbf{10^{-1}}]$
- Output: Newton iterations over time

Test case: 1D water injection into a dry soil

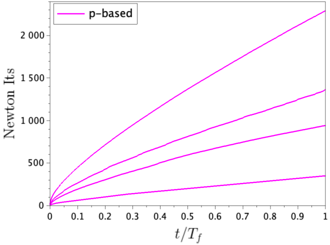
Saturation profiles



Unsaturated flow

- Soil properties: Loam with Van Genuchten *kr*-*Pc* relation
- Input: Dry initial conditions $s_0 \in [\mathbf{10^{-6}}, 10^{-1}]$
- Output: Newton iterations over time

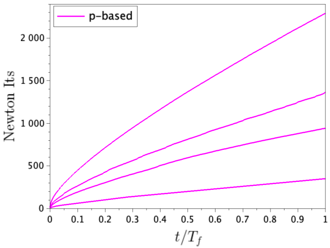
Performance: p-, nonlin. and lin. s-formulations



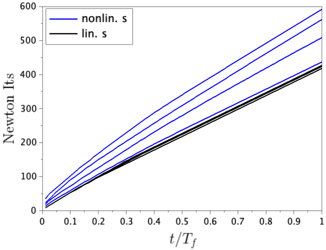
$s_0 \in \{0.1, 0.02, 0.01, 0.001\}$

- Space disc.: $Nx = 100$
- Time disc.: $\Delta t_{max} = T_f/\mathbf{100}$

Performance: p-, nonlin. and lin. s-formulations



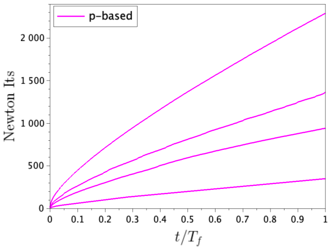
$s_0 \in \{0.1, 0.02, 0.01, 0.001\}$



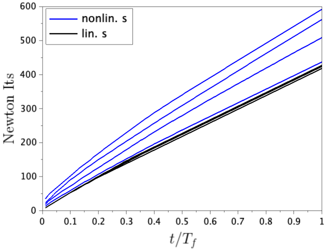
$s_0 \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$

- Space disc.: $Nx = 100$
- Time disc.: $\Delta t_{max} = T_f / \mathbf{100}$

Performance: p-, nonlin. and lin. s-formulations



$s_0 \in \{0.1, 0.02, 0.01, 0.001\}$



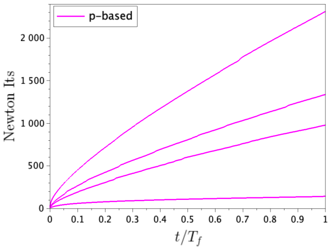
$s_0 \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$

- Space disc.: $Nx = 100$
- Time disc.: $\Delta t_{max} = T_f/\mathbf{100}$

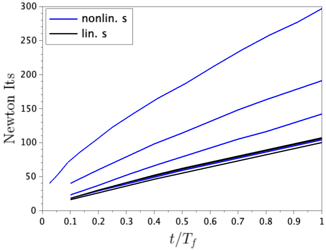
Observations:

- $p-$formulation: convergence is local, not robust
- linear $s-$formulation: fully robust, seemingly unconditional convergence
- nonlin. $s-$formulation: in between

Performance: p-, nonlin. and lin. s-formulations



$s_0 \in \{0.1, 0.02, 0.01, 0.001\}$



$s_0 \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$

- Space disc.: $Nx = 100$
- Time disc.: $\Delta t_{max} = T_f/\mathbf{10}$

Observations:

- $p-$formulation: convergence is local, not robust
- linear $s-$formulation: fully robust, convergence seems global
- nonlin. $s-$formulation: in between

Primary variable selection and (conservative) formulations

Nonlinear p-formulation

$$\partial_t S(p) - \mathrm{div}\,(k(S(p))\nabla p) = 0$$

Nonlinear s-formulation

$$\partial_t s - \mathrm{div}\,(k(s)\nabla P(s)) = 0$$

Linearized s-formulation

$$\partial_t s - \mathrm{div}\,\big(k(s)P'(s)\nabla s\big) = 0$$

	Variably sat.	Heterogeneities	Performance
Nonlin. p	Okay	Okay	Poor
Nonlin. s	Unsat	Okay	Okayish
Lin. s	Unsat	Not trivial	Good

Primary variable selection and (conservative) formulations

Nonlinear p-formulation

$$\partial_t S(p) - \mathrm{div}\,(k(S(p))\nabla p) = 0$$

Nonlinear s-formulation

$$\partial_t s - \mathrm{div}\,(k(s)\nabla P(s)) = 0$$

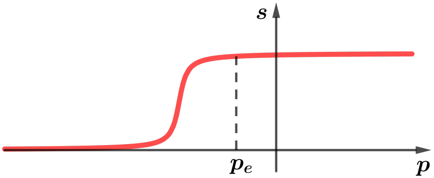
Linearized s-formulation

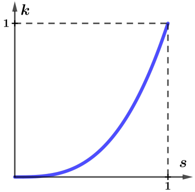
$$\partial_t s - \mathrm{div}\,\big(k(s)P'(s)\nabla s\big) = 0$$

	Variably sat.	Heterogeneities	Performance
Nonlin. p	Okay	Okay	Poor
Nonlin. s	Unsat	Okay	Okayish
Lin. s	Unsat	Not trivial	Good

Bibliography: *Haverkamp et al. '77, Vauclin '79, Van Genuchten '82, Hills et al. '89, Celia et al. '90, Kirkland, Hills, & Wierenga '92*

What is so terribly wrong with the nonlinear p−formulation?

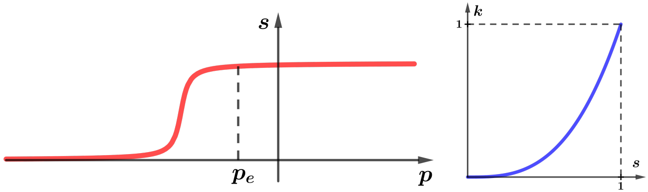




Think of a discretized system corresponding to

$$\partial_t S(p) - \operatorname{div}\left(k(S(p))\nabla p\right) = 0$$

What is so terribly wrong with the nonlinear p−formulation?



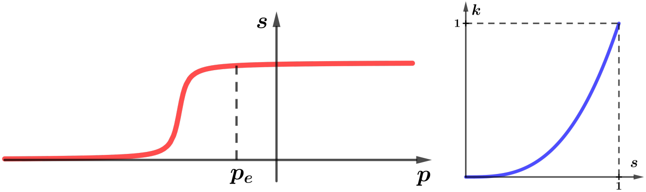
Think of a discretized system corresponding to

$$\partial_t S(p) - \mathrm{div}\,(k(S(p))\nabla p) = 0$$

Newton's iterations will result in a linearized system

$$S'(p)\partial_t p - \mathrm{div}\,(k(S(p))\nabla p) = 0$$

What is so terribly wrong with the nonlinear p−formulation?



Think of a discretized system corresponding to

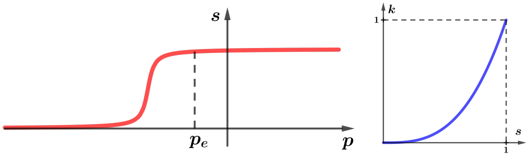
$$\partial_t S(p) - \mathrm{div}\,(k(S(p))\nabla p) = 0$$

Newton's iterations will result in a linearized system

$$S'(p)\partial_t p - \mathrm{div}\,(k(S(p))\nabla p) = 0$$

For small s gives $0 \approx 0$.

Nonlinear vs linear s−formulation

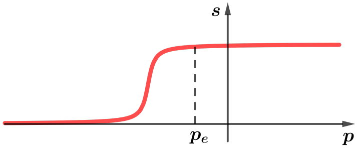


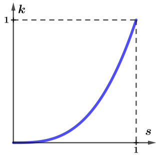
Upwind discretization of $\mathbf{k(s)P'(s)\nabla s}$

$$F_{KL} \propto k(s_{KL})P'(s_{KL})(s_K - s_L)$$

Compensation: small $\times$ large $=$ bounded

Nonlinear vs linear $s-$formulation





Upwind discretization of $\mathbf{k(s)P'(s)\nabla s}$

$$F_{KL} \propto k(s_{KL})P'(s_{KL})(s_K - s_L)$$

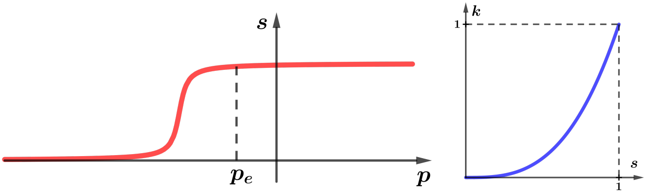
Compensation: small $\times$ large $=$ bounded

Upwind discretization of $\mathbf{k(s)\nabla P(s)}$

$$F_{KL} \propto k(s_{KL})(p_K - p_L) = k(s_{KL})P'(s^{\star}_{KL})(s_K - s_L)$$

(!) No compensation

Nonlinear vs linear $s-$formulation



Other issues of the nonlinear s-formulation

$$\partial_t s - \mathrm{div}\,(k(s)\nabla P(s)) = 0$$

- Floating point overflows $\Rightarrow$ need for regularization
- Does not combines well with nonlinear preconditioning

Primary variable selection and (conservative) formulations

Nonlinear p-formulation

$$\partial_t S(p) - \operatorname{div}\left(k(S(p))\nabla p\right) = 0$$

Nonlinear s-formulation

$$\partial_t s - \operatorname{div}\left(k(s)\nabla P(s)\right) = 0$$

Linearized s-formulation

$$\partial_t s - \operatorname{div}\left(k(s)P'(s)\nabla s\right) = 0$$

	Variably sat.	Heterogeneities	Performance
Nonlin. p	Okay	Okay	Poor
Nonlin. s	Unsat	Okay	Okayish
Lin. s	Unsat	Not trivial	Good

Objective: get best of the available formulations

- First step: nonlin. p + nonlin. s
- Second step: nonlin. p + lin. s

Variable switching

Original problem

$$\partial_t s - \operatorname{div}(k(s)\nabla p) = 0, \qquad s = S(p)$$

Variable substitution

$$\tau \mapsto \overline{s}(\tau), \overline{p}(\tau) \qquad \text{s.t.} \qquad \overline{s}(\tau) = S(\overline{p}(\tau)) \text{ for all } \tau$$

Variable switching

$$\tau \propto \begin{cases} \overline{s}(\tau) & \tau \leq \tau_{sw} \\ \overline{p}(\tau) & \tau > \tau_{sw} \end{cases}$$

Variable switching

Original problem

$$\partial_t s - \operatorname{div}\left(k(s)\nabla p\right) = 0, \qquad s = S(p)$$

Variable substitution

$$\tau \mapsto \overline{s}(\tau), \overline{p}(\tau) \qquad \text{s.t.} \qquad \overline{s}(\tau) = S\left(\overline{p}(\tau)\right) \text{ for all } \tau$$

Variable switching

$$\tau \propto \begin{cases} \overline{s}(\tau) & \tau \leq \tau_{sw} \\ \overline{p}(\tau) & \tau > \tau_{sw} \end{cases}$$

Reformulated problem (nonlinear)

$$\partial_t \overline{s}(\tau) - \operatorname{div}\left(k(\overline{s}(\tau))\nabla \overline{p}(\tau)\right) = 0$$

Variable switching

Original problem

$$\partial_t s - \operatorname{div}(k(s)\nabla p) = 0, \qquad s = S(p)$$

Variable substitution

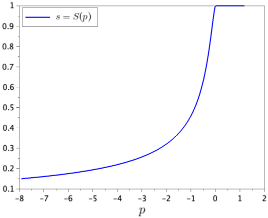
$$\tau \mapsto \overline{s}(\tau), \overline{p}(\tau) \qquad \text{s.t.} \qquad \overline{s}(\tau) = S(\overline{p}(\tau)) \;\; \text{for all } \tau$$

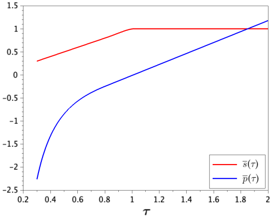
Variable switching

$$\tau \propto \begin{cases} \overline{s}(\tau) & \tau \leq \tau_{sw} \\ \overline{p}(\tau) & \tau > \tau_{sw} \end{cases}$$

Reformulated problem (nonlinear)

$$\partial_t \overline{s}(\tau) - \operatorname{div}(k(\overline{s}(\tau))\nabla \overline{p}(\tau)) = 0$$





Variable switching

Original problem

$$\partial_t s - \operatorname{div}\left(k(s)\nabla p\right) = 0, \qquad s = S(p)$$

Variable substitution

$$\tau \mapsto \overline{s}(\tau), \overline{p}(\tau) \qquad \text{s.t.} \qquad \overline{s}(\tau) = S\left(\overline{p}(\tau)\right) \text{ for all } \tau$$

Variable switching

$$\tau \propto \begin{cases} \overline{s}(\tau) & \tau \leq \tau_{sw} \\ \overline{p}(\tau) & \tau > \tau_{sw} \end{cases}$$

Reformulated problem (nonlinear)

$$\partial_t \overline{s}(\tau) - \operatorname{div}\left(k(\overline{s}(\tau))\nabla \overline{p}(\tau)\right) = 0$$

	Variably sat.	Heterogeneities	Performance
Nonlin. p	Okay	Okay	Poor
Nonlin. s	Unsat	Okay	Okayish
Nonlin. τ	**Okay**	**Okay**	**Okayish**

Variable switching

Original problem

$$\partial_t s - \operatorname{div}\left(k(s)\nabla p\right) = 0, \qquad s = S(p)$$

Variable substitution

$$\tau \mapsto \bar{s}(\tau), \bar{p}(\tau) \qquad \text{s.t.} \qquad \bar{s}(\tau) = S\left(\bar{p}(\tau)\right) \text{ for all } \tau$$

Variable switching

$$\tau \propto \begin{cases} \bar{s}(\tau) & \tau \leq \tau_{sw} \\ \bar{p}(\tau) & \tau > \tau_{sw} \end{cases}$$

Reformulated problem (nonlinear)

$$\partial_t \bar{s}(\tau) - \operatorname{div}\left(k(\bar{s}(\tau))\nabla \bar{p}(\tau)\right) = 0$$

	Variably sat.	Heterogeneities	Performance
Nonlin. p	Okay	Okay	Poor
Nonlin. s	Unsat	Okay	Okayish
Nonlin. τ	**Okay**	**Okay**	**Okayish**

Bibliography:

- *Kirkland et al. '92, Forsyth et al. '95, Diersch '98, Diersch & Perrochet '99, Krabbenhøft '07*
- *B. & Cancès '17, B. Groza, et al '17, Bassetto et al '20*

Variable switching

Original problem

$$\partial_t s - \operatorname{div}(k(s)\nabla p) = 0, \qquad s = S(p)$$

Variable substitution

$$\tau \mapsto \overline{s}(\tau), \overline{p}(\tau) \qquad \text{s.t.} \qquad \overline{s}(\tau) = S(\overline{p}(\tau)) \text{ for all } \tau$$

Variable switching

$$\tau \propto \begin{cases} \overline{s}(\tau) & \tau \leq \tau_{sw} \\ \overline{p}(\tau) & \tau > \tau_{sw} \end{cases}$$

Reformulated problem (partly linearized)

$$\partial_t \overline{s}(\tau) - \operatorname{div}\left(k(\overline{s}(\tau))\overline{p}'(\tau)\nabla\tau\right) = 0$$

	Variably sat.	Heterogeneities	Performance
Nonlin. p	Okay	Okay	Poor
Lin. s	Unsat	Not trivial	Good
Lin. τ	**Okay**	**???**	**Good**

Bibliography:

- *Kirkland et al. '92, Forsyth et al. '95, Diersch '98, Diersch & Perrochet '99, Krabbenhøft '07*
- *B. & Cancès '17, B. Groza, et al '17, Bassetto et al '20*

Variable switching

Original problem

$$\partial_t s - \mathrm{div}\,(k(s)\nabla p) = 0, \qquad s = S(p)$$

Variable substitution

$$\tau \mapsto \overline{s}(\tau), \overline{p}(\tau) \qquad \text{s.t.} \qquad \overline{s}(\tau) = S\left(\overline{p}(\tau)\right) \text{ for all } \tau$$

Variable switching

$$\tau \propto \begin{cases} \overline{s}(\tau) & \tau \leq \tau_{sw} \\ \overline{p}(\tau) & \tau > \tau_{sw} \end{cases}$$

Reformulated problem (partly linearized)

$$\partial_t \overline{s}(\tau) - \mathrm{div}\,\left(k(\overline{s}(\tau))\overline{p}'(\tau)\nabla\tau\right) = 0$$

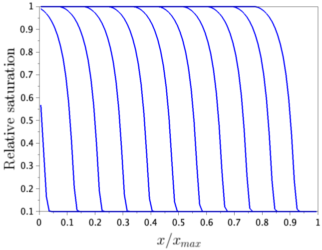
	Variably sat.	Heterogeneities	Performance
Nonlin. p	Okay	Okay	Poor
Lin. s	Unsat	Not trivial	Good
Lin. τ	**Okay**	**Okay**	**Good**

Bibliography:

- *Kirkland et al. '92, Forsyth et al. '95, Diersch '98, Diersch & Perrochet '99, Krabbenhøft '07*
- *B. & Cancès '17, B. Groza, et al '17, Bassetto et al '20*

Test case: 1D water injection into a dry soil

Saturation profiles

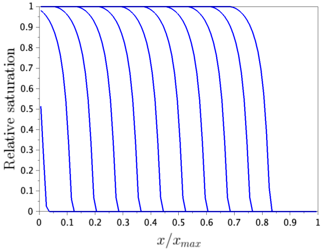


Pressurized flow

- Soil properties: Loam with Van Genuchten *kr*-*Pc* relation
- Input: Dry initial conditions $s_0 \in [10^{-6}, \mathbf{10^{-1}}]$
- Output: Newton iterations over time

Test case: 1D water injection into a dry soil

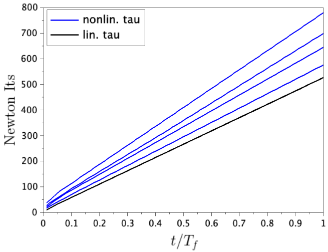
Saturation profiles



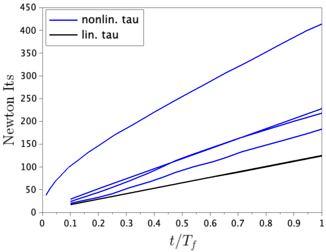
Pressurized flow

- Soil properties: Loam with Van Genuchten *kr*-*Pc* relation
- Input: Dry initial conditions $s_0 \in [\mathbf{10^{-6}}, 10^{-1}]$
- Output: Newton iterations over time

Performance: nonlinear vs. linear variable switching formulation



$\Delta t_{max} = T_f/\mathbf{100}$



$\Delta t_{max} = T_f/\mathbf{10}$

- Initial condition: $s_0 \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$
- Space disc.: $Nx = 100$

Outline

Introduction

Primary variables and formulations

Preconditioned Newton method

Conclusion

Preconditioned Newton method

Illustration over a toy problem and a couple of toy methods

Find $\boldsymbol{u} \in \mathbb{R}^N$ s.t.

$$f(\boldsymbol{u}) + A\boldsymbol{u} = \boldsymbol{b}, \qquad \text{with} \qquad f(\boldsymbol{u})_i = f_i(u_i)$$

Splitting $A = P - Q$:

$$f(\boldsymbol{u}) + P\boldsymbol{u} = Q\boldsymbol{u} + \boldsymbol{b}$$

Preconditioned Newton method

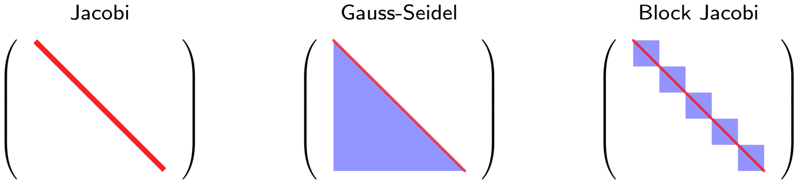
Illustration over a toy problem and a couple of toy methods

Find $\boldsymbol{u} \in \mathbb{R}^N$ s.t.

$$f(\boldsymbol{u}) + A\boldsymbol{u} = \boldsymbol{b}, \qquad \text{with} \qquad f(\boldsymbol{u})_i = f_i(u_i)$$

Splitting $A = P - Q$:

$$\underbrace{f(\boldsymbol{u}) + P\boldsymbol{u}}_{M(u)} = \underbrace{Q\boldsymbol{u} + \boldsymbol{b}}_{N(u)}$$



Preconditioned Newton method

Illustration over a toy problem and a couple of toy methods

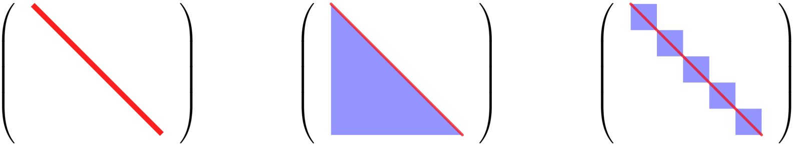
Find $\boldsymbol{u} \in \mathbb{R}^N$ s.t.

$$f(\boldsymbol{u}) + A\boldsymbol{u} = \boldsymbol{b}, \qquad \text{with} \qquad f(\boldsymbol{u})_i = f_i(u_i)$$

Splitting $A = P - Q$:

$$\underbrace{f(\boldsymbol{u}) + P\boldsymbol{u}}_{M(u)} = \underbrace{Q\boldsymbol{u} + \boldsymbol{b}}_{N(u)}$$

Jacobi — Gauss-Seidel — Block Jacobi



Preconditioned Newton method

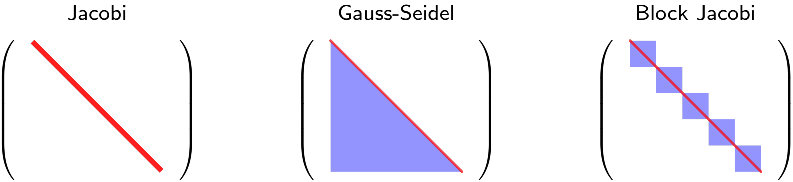
Illustration over a toy problem and a couple of toy methods

Find $\boldsymbol{u} \in \mathbb{R}^N$ s.t.

$$f(\boldsymbol{u}) + A\boldsymbol{u} = \boldsymbol{b}, \qquad \text{with} \qquad f(\boldsymbol{u})_i = f_i(u_i)$$

Splitting $A = P - Q$:

$$\underbrace{f(\boldsymbol{u}) + P\boldsymbol{u}}_{M(u)} = \underbrace{Q\boldsymbol{u} + \boldsymbol{b}}_{N(u)}$$



Stationary iterations

$$\boldsymbol{u}_{k+1} = M^{-1}\left(N(\boldsymbol{u}_k)\right)$$

Newton's method applied to

$$\boldsymbol{u} - M^{-1}\left(N(\boldsymbol{u})\right) = 0$$

Preconditioned Newton method

Illustration over a toy problem and a couple of toy methods

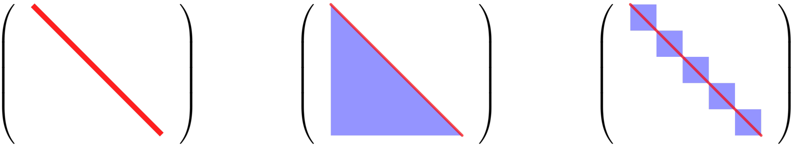
Find $\boldsymbol{u} \in \mathbb{R}^N$ s.t.

$$f(\boldsymbol{u}) + A\boldsymbol{u} = \boldsymbol{b}, \qquad \text{with} \qquad f(\boldsymbol{u})_i = f_i(u_i)$$

Splitting $A = P - Q$:

$$\underbrace{f(\boldsymbol{u}) + P\boldsymbol{u}}_{M(u)} = \underbrace{Q\boldsymbol{u} + \boldsymbol{b}}_{N(u)}$$

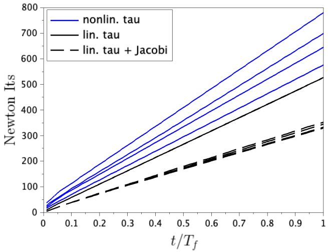
Jacobi   Gauss-Seidel   Block Jacobi



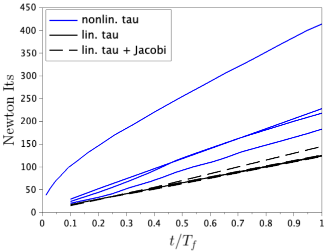
Extensions

- General systems $F(\boldsymbol{u}) = 0$
- General DD methods (e.g. ASPIN, RASPEN)

Performance: Jacobi-Newton method



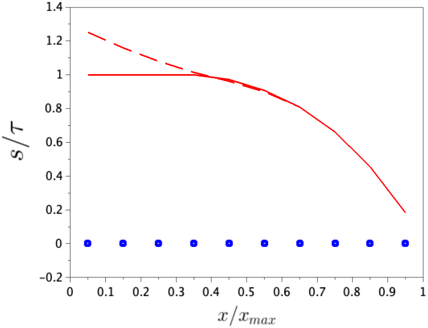
$\Delta t_{max} = T_f / \mathbf{100}$



$\Delta t_{max} = T_f / \mathbf{10}$

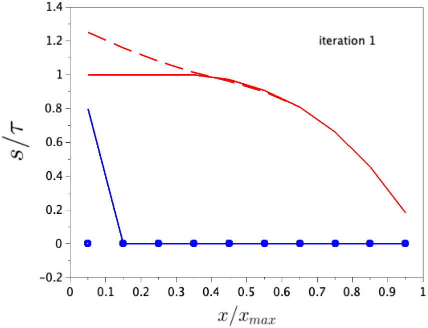
- Initial condition: $s_0 \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$
- Space disc.: $Nx = 100$

Jacobi-Newton iterates



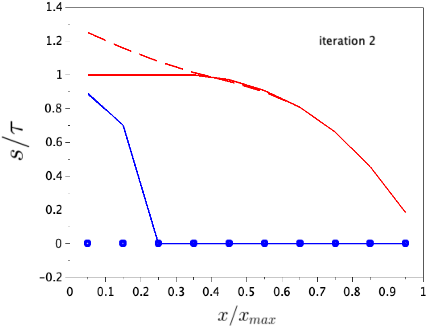
#unk = 10

Jacobi-Newton iterates



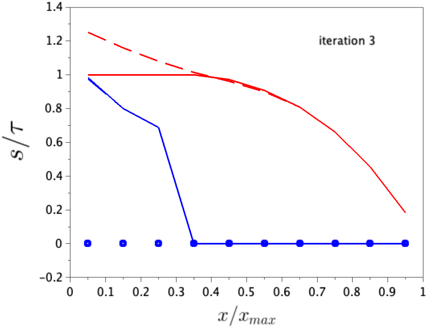
#unk = 10

Jacobi-Newton iterates



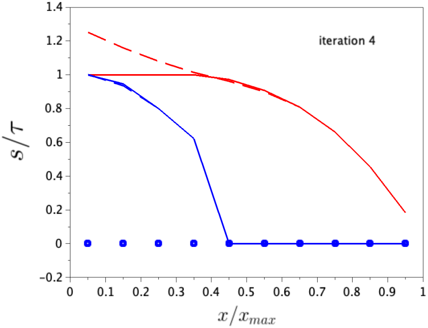
#unk = 10

Jacobi-Newton iterates



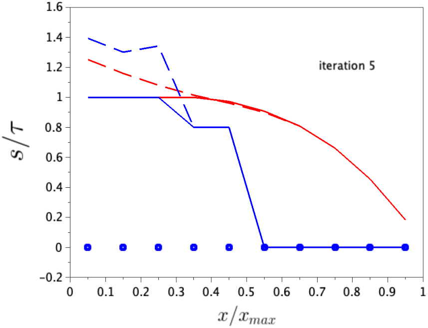
#unk = 10

Jacobi-Newton iterates



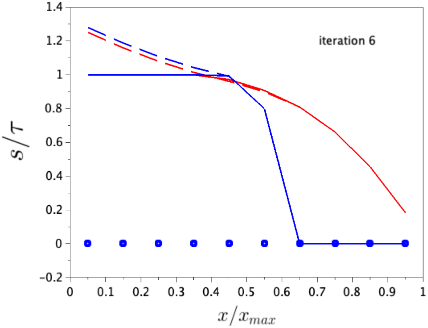
#unk = 10

Jacobi-Newton iterates



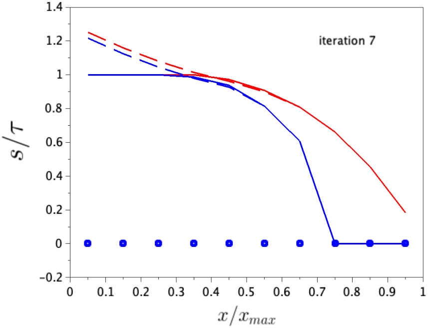
#unk = 10

Jacobi-Newton iterates



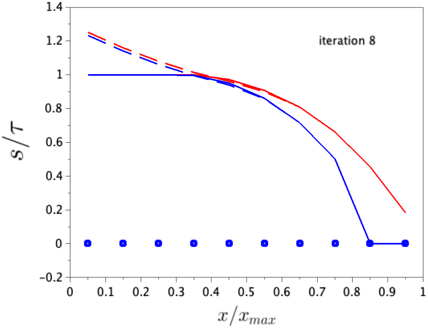
#unk = 10

Jacobi-Newton iterates



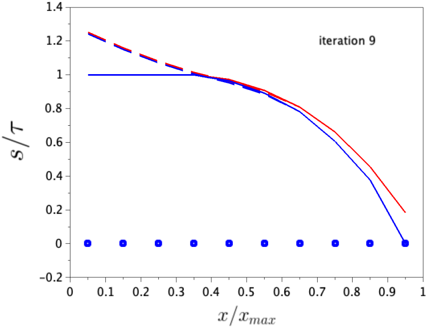
#unk $= 10$

Jacobi-Newton iterates



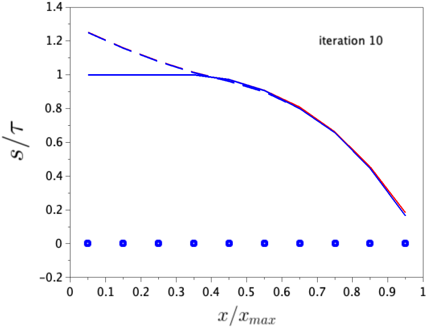
#unk = 10

Jacobi-Newton iterates



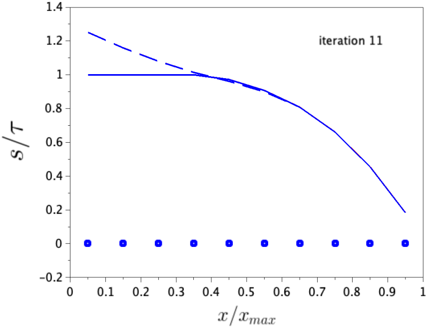
#unk = 10

Jacobi-Newton iterates



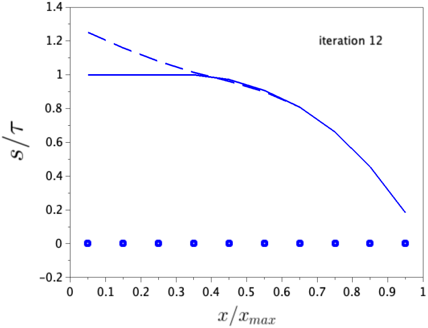
#unk = 10

Jacobi-Newton iterates



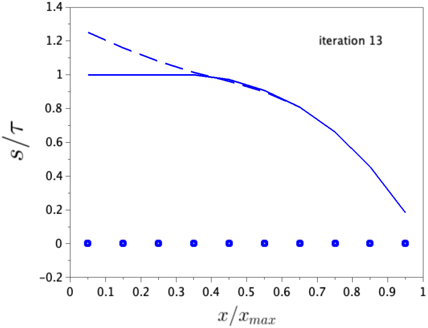
#unk = 10

Jacobi-Newton iterates



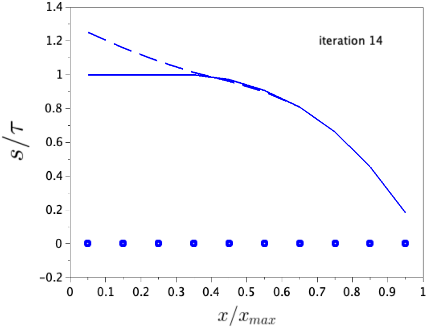
#unk = 10

Jacobi-Newton iterates



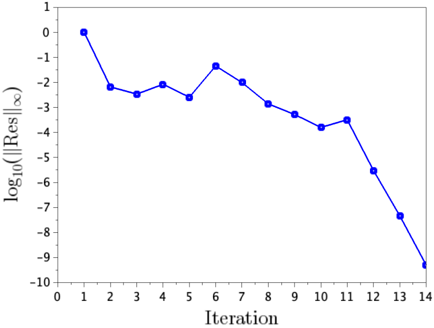
#unk = 10

Jacobi-Newton iterates



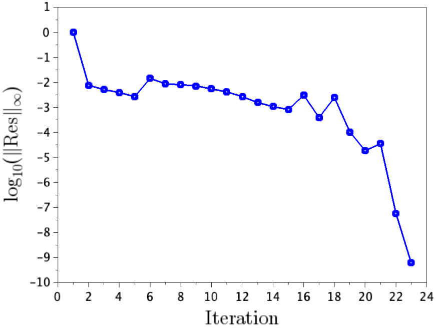
#unk = 10

Jacobi-Newton iterates



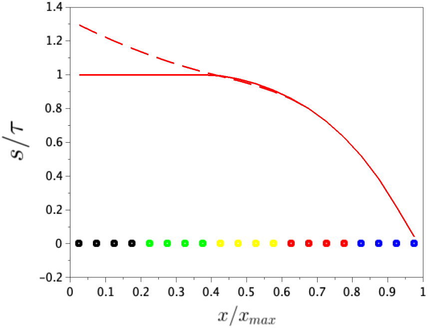
$\#\text{unk} = 10$

Jacobi-Newton iterates



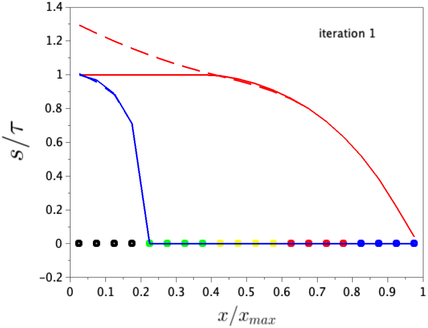
#unk = 20

Block Jacobi-Newton iterates



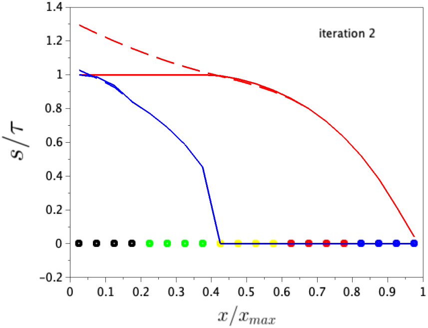
#unk = 20

Block Jacobi-Newton's iterates



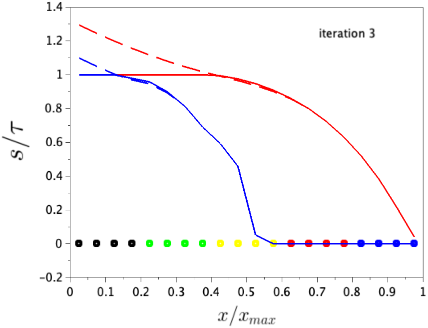
#unk = 20

Block Jacobi-Newton's iterates



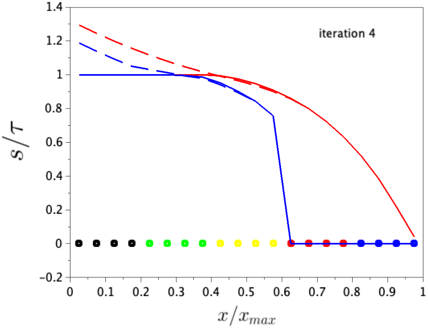
#unk = 20

Block Jacobi-Newton's iterates



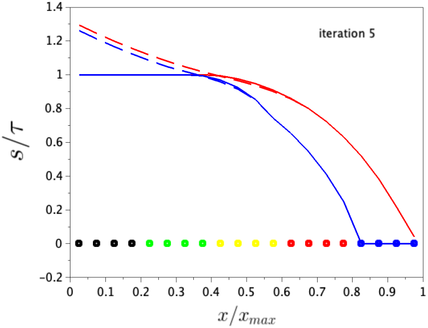
#unk = 20

Block Jacobi-Newton's iterates



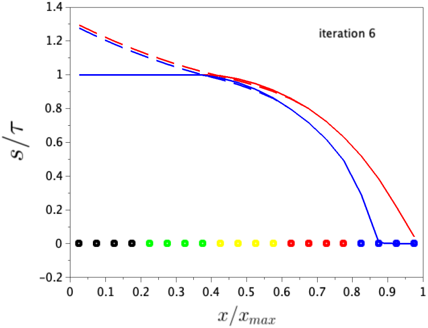
#unk = 20

Block Jacobi-Newton's iterates



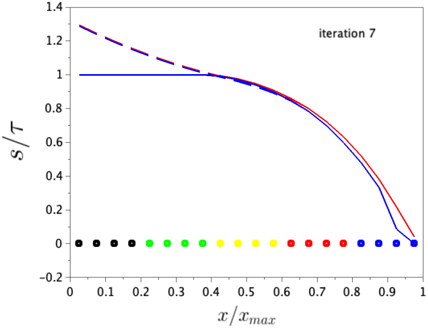
#unk = 20

Block Jacobi-Newton's iterates



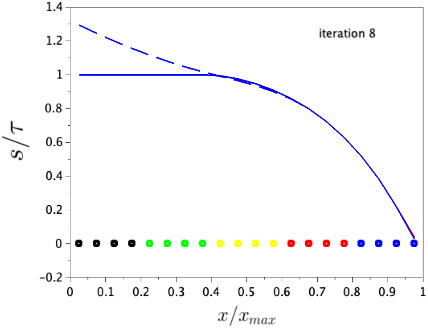
#unk = 20

Block Jacobi-Newton's iterates



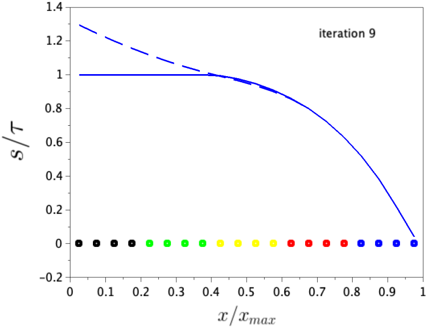
#unk = 20

Block Jacobi-Newton's iterates



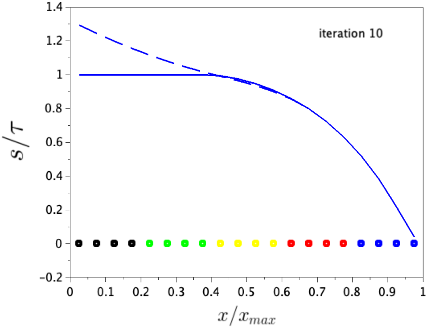
#unk = 20

Block Jacobi-Newton's iterates



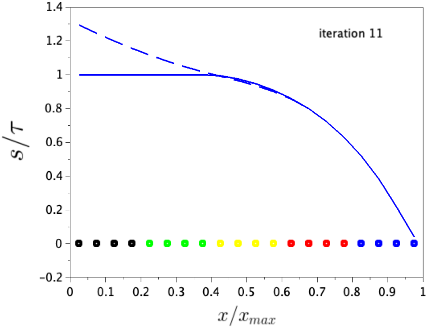
$\#\text{unk} = 20$

Block Jacobi-Newton's iterates



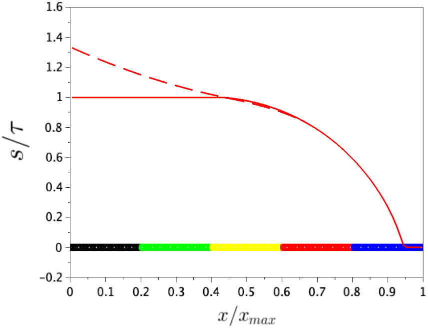
#unk = 20

Block Jacobi-Newton's iterates



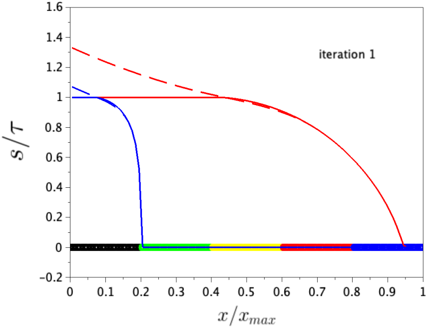
#unk = 20

Block Jacobi-Newton's iterates



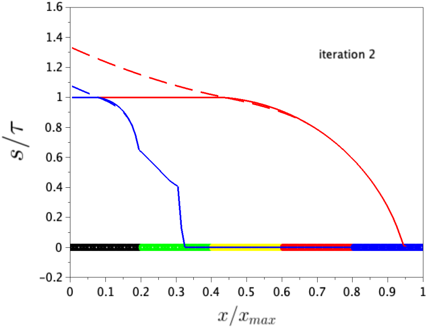
#unk = 100

Block Jacobi-Newton's iterates



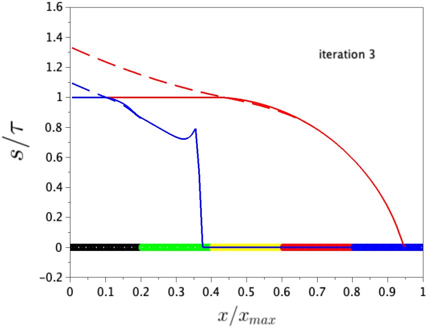
#unk = 100

Block Jacobi-Newton's iterates



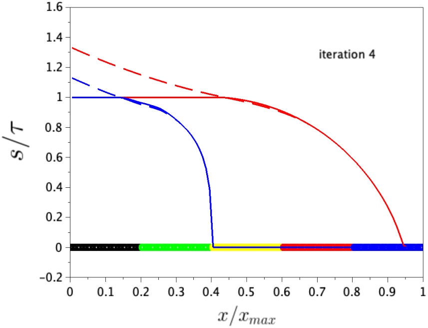
#unk = 100

Block Jacobi-Newton's iterates



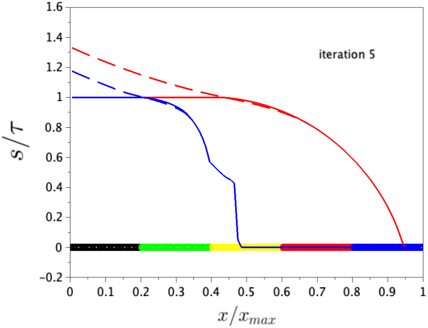
#unk = 100

Block Jacobi-Newton's iterates



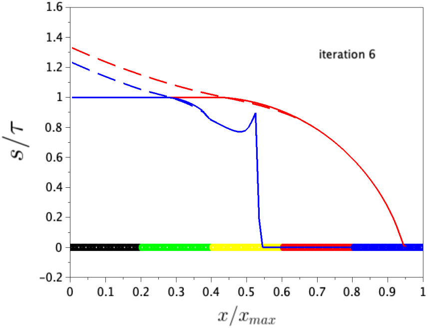
#unk = 100

Block Jacobi-Newton's iterates



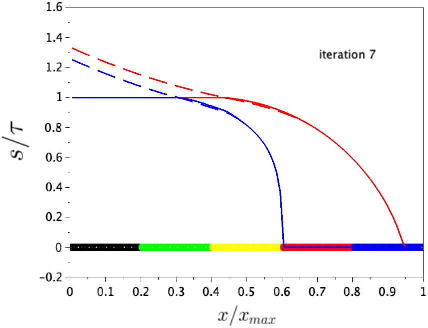
#unk = 100

Block Jacobi-Newton's iterates



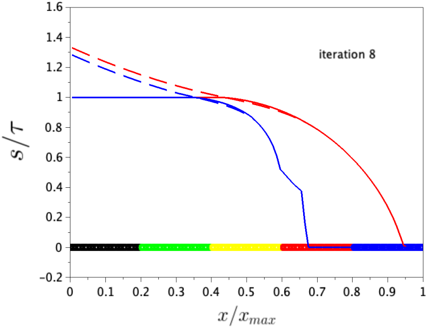
$\#$unk $= 100$

Block Jacobi-Newton's iterates



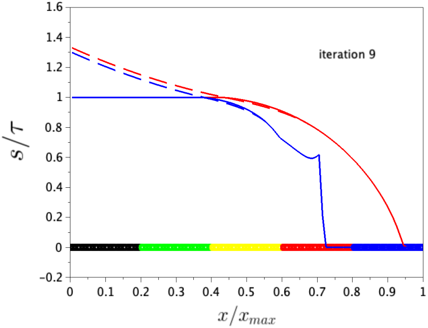
#unk = 100

Block Jacobi-Newton's iterates



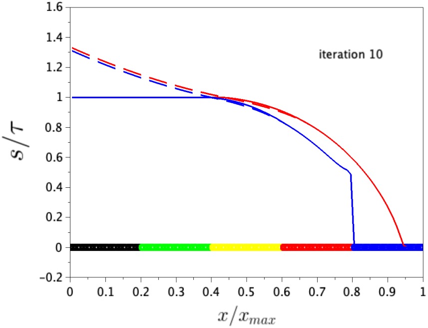
#unk = 100

Block Jacobi-Newton's iterates



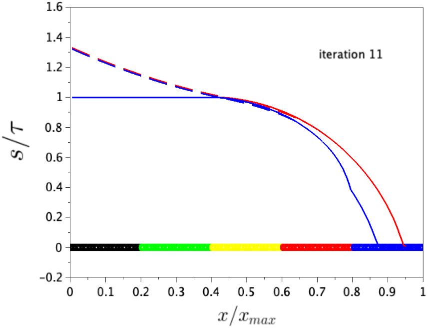
$\#\text{unk} = 100$

Block Jacobi-Newton's iterates



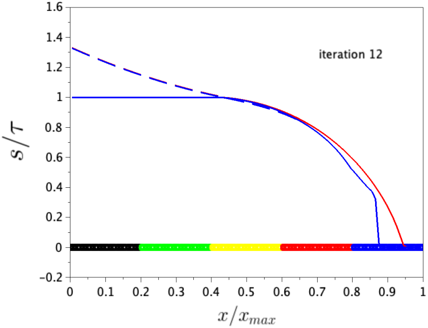
#unk = 100

Block Jacobi-Newton's iterates



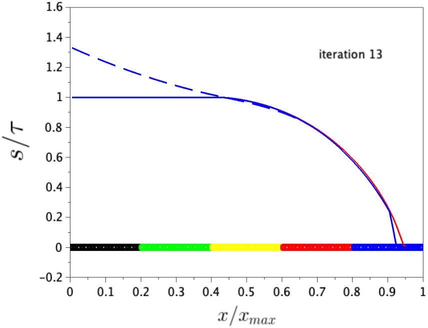
#unk = 100

Block Jacobi-Newton's iterates



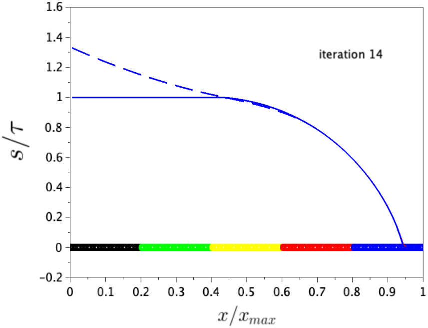
#unk = 100

Block Jacobi-Newton's iterates



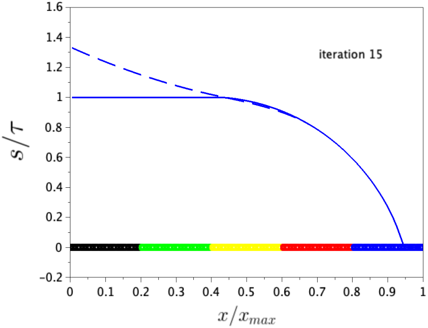
$\#\text{unk} = 100$

Block Jacobi-Newton's iterates



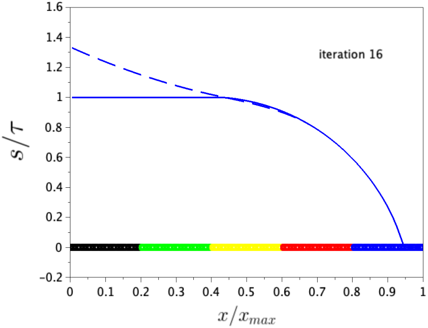
#unk = 100

Block Jacobi-Newton's iterates



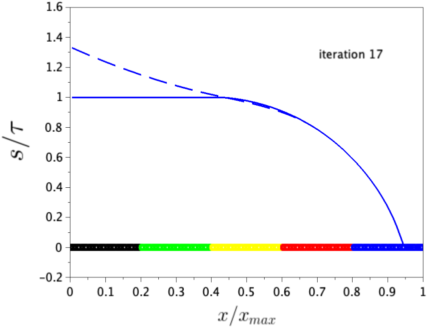
#unk = 100

Block Jacobi-Newton's iterates



#unk = 100

Block Jacobi-Newton's iterates



#unk = 100

Outline

Conclusions

Ways of improving robustness

- Heuristic variable substitution and reformulation
- Nonlinear preconditioning
- Large time steps

Analysis

- Global monotone convergence using
  Kirchhoff formulation and TPFA

Perspectives & ongoing work

- Heterogeneous problems
- Nonlinear two-level RAS methods, front tracking
- Connections to space-time DD?

Conclusions

Ways of improving robustness

- Heuristic variable substitution and reformulation
- Nonlinear preconditioning
- **Large time steps**

Analysis

- **Global** monotone convergence using Kirchhoff formulation and TPFA

Perspectives & ongoing work

- Heterogeneous problems
- Nonlinear two-level RAS methods, front tracking
- Connections to **space-time** DD?