جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Nonlinear Preconditioning for Implicit Solution of Discretized PDEs

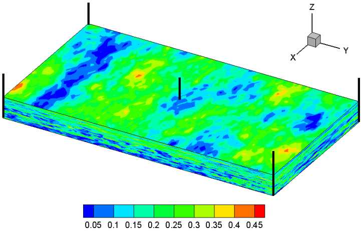
David Keyes, KAUST

Preconditioning 2024

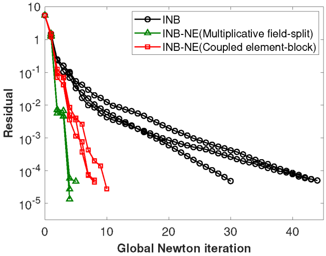
11.06.2024

Where we're going...

- Implicit time-stepping of multi-rate nonlinear problems
- Using Newton iteration on each time step
- With various forms of **nonlinear preconditioning**



(a) Porosity for SPE10 model with injector in the middle and producers in the corners



(b) Residual history at three early time steps

Along the way, we'll meet several nonlinear preconditioners

Why we're going there ...

- While the convergence rate of Newton's method is asymptotically independent of the spatial refinement of discretized PDEs, the embedded linear solves are *not*.
- In many implicit PDE applications – reservoir modeling, aerodynamics, combustion, etc. – the linear solves on each time step consume *80–90%* of the total execution time.
- The linear solves also make up the *most* memory-demanding and *least* parallel scalable phase of the code → hence the importance of Preconditioning'XY :-)
- One means of reducing their consumption is to do fewer *global* Newton solves. Nonlinear preconditioning replaces some global Newton solves by smaller and better linearly conditioned *local* Newton solves.

Goal: address "nonlinear stiffness"

What is stiffness?

- Formally, time-dependent systems are said to be "stiff" if the ratio of the largest to smallest eigenvalues of the principal modes of locally linearized systems is wide.
  - i.e., different modes have widely varying decay rates or oscillation frequencies
- Practically, a system is "stiff" if the stability requirements force a smaller timestep to resolve the phenomena of interest than is required by accuracy alone.

Analogously, we call a nonlinear algebraic system "stiff" if a some components of the Newton correction force a small Newton step, while most components could take the full step – if the progress of the whole is constrained by a part.

Origin of "nonlinear stiffness"

Consider a nonlinear problem $F(x) = 0$ of n nonlinear equations in n unknowns.

The multivariate Taylor expansion at any point x_k gives

$$\boxed{F(x) = F(x_k) + F'(x_k)(x - x_k)} + O(\|x - x_k\|^2).$$

where $F'(x_k) \equiv J^k$ is the $n \times n$ Jacobian matrix evaluated at x_k.

Truncating to the boxed linear terms and setting $F(x)$ to zero gives Newton's method:

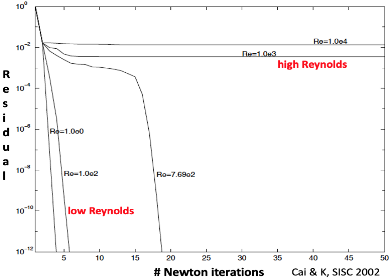
$$J^k s^k = -F(x^k); \quad x^{k+1} \leftarrow x^k + s^k$$

Each Newton step requires the inverse action of a typically sparse and ill-conditioned Jacobian, typically handled by linearly preconditioned Krylov solvers with Jacobian-vector products approximated by directional derivatives of $F(x)$ – the Jacobian-free Newton Krylov (JFNK) method (Knoll & K, JCP, 2004).

Origin of "nonlinear stiffness"

When high-order terms dominate, the linear model is not a suitable approximation to $F(x)$.

- Strong nonlinearities result in a long plateau period of the residual $\|F(x_k)\|$.
- Only a small number of the n components of the solution may undergo significant updates in Newton corrections that are highly damped by linesearch backtracking or trust region globalization.



Newton methods may thus waste considerable resources solving global linear systems in problems that are "nonlinearly stiff" until they find the convergence domain. Instances include material fronts, reaction fronts, shocks, recirculation zones, etc.

Example: nonlinear stiffness in reservoir problems

Possible sources:

- High contrast heterogeneous permeability or porosity
- Strong nonlinearity of relative permeability functions
- Spatially varying capillary pressure
- Faults, channels, and voids

These may cause Newton to damp to death, or require small timesteps to robustify Newton's method.

The latter erodes the intrinsic advantages of a fully implicit method, namely:

- Choosing the timestep adaptively based on temporal truncation error accuracy requirements alone
- Going higher than first-order in time

Quoting the reservoir modeling experts ...

- "The standard approach in reservoir simulation (and in simulation of carbon storage) is to use a fully implicit discretization and iteratively solve for all the primary unknowns at once using Newton's method. This requires repeated solves of large, ill-conditioned linearized systems of equations." – **Knut-Andreas Lie (SINTEF)** *et al.*, Comp. Geosci., 2021
- "The standard way to handle nonlinearity is by some variation of Newton iterations, in which a system of linear equations must be solved for each iteration. This consumes a considerable amount of the total simulation time, and methods have been developed ... to reduce the number of Newton iterations needed." – **Jan Nordbotten (Bergen)** *et al.*, JCP, 2013
- "Multiphase flow introduces strong nonlinearities, and a naive implementation of Newton's method may fail to converge when large timesteps are taken... Smart strategies to address the intrinsic nonlinearities in these systems remain an active area of research." – **Hamdi Tchelepi (Stanford)** *et al.*, CMAME, 2019

Outline

Enter nonlinear preconditioning

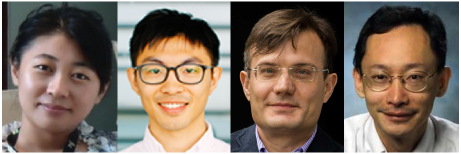
- A nonlinear "preconditioner" performs nonlinear relaxation within one or more subsets of equations and unknowns, inside the context of an outer Newton "accelerator."
  - Analogous to linear preconditioners, such as domain decomposition or multigrid, inside a Krylov accelerator
  - Such *linearly* preconditioned Krylov methods are often used inside *both* the nonlinear subproblems and the global problem
- A prime consideration in selecting an *inner* nonlinear preconditioner is whether the resulting *outer* nonlinear problem is amenable to linear preconditioning of the Jacobian.
- Early domain-decomposed left nonlinear preconditioners **ASPIN** and **MSPIN** complicated outer linear preconditioning by replacing a sparse outer Jacobian with a dense one.
- Right preconditioners like **INB-NE** retain the original (generally sparse) Jacobian.
- Left preconditioner **NEPIN** (introduced herein) can also employ the original Jacobian.

Origins of nonlinear preconditioning: nonlinear "projections" in subspaces

- Decomposed by domain (nonlinear block Jacobi)
  - **1997**, *On the nonlinear domain decomposition method*, M. Dryja & W. Hackbusch, BIT Numerical Mathematics **37**:296–311.
- Decomposed by field (nonlinear block Gauss-Seidel)
  - **2009**, R. Ernst, B. Flemisch & B. Wohlmuth, *A multiplicative Schwarz method and its application to nonlinear acoustic-structure interaction*, ESAIM Math. Modeling and Numerical Analysis, **43**:487–506.

These contributions lacked an outer Newton accelerator.

Background for this talk, with mostly KAUST co-authors



2015	Liu & K	Field-split preconditioned inexact Newton algorithms	SISC
2016	Liu & K	Convergence analysis for the multiplicative Schwarz preconditioned inexact Newton algorithm	SINUM
2018	Liu, K & Krause	A note on adaptive nonlinear preconditioning techniques	SISC
2020	Luo, Cai & K	Nonlinear preconditioning strategies for two-phase flows in porous media	SISC
2020	Luo, Cai, Yan, Xu & K	A multilayer nonlinear elimination preconditioned inexact Newton method for steady-state flows	SISC
2021	Liu & K	Approximate error bounds on solutions of nonlinearly preconditioned PDEs	SISC
2022	Liu, Hwang, Luo, Cai & K	A nonlinear elimination preconditioned inexact Newton algorithm	SISC
2024	Liu, Gao, Yu & K	Overlapping and nonoverlapping Schwarz preconditioning for linear and nonlinear systems	JCP

Selected bibliography: school of Xiao-Chuan Cai

2002	C & K	Nonlinearly preconditioned inexact Newton algorithms	SISC
2011	C & Li	Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity	SISC
2015	Huang, Su & C	A parallel adaptive nonlinear elimination preconditioned inexact Newton method for transonic full potential equation	Comp Fluids
2016	Yang, Huang & C	A nonlinearly preconditioned inexact Newton algorithm for steady state lattice Boltzmann equations	SISC
2016	Yang, Huang & C	Nonlinear preconditioning techniques for full-space Lagrange-Newton solution of PDE-constrained optimization problems	SISC
2019	Gong & C	A nonlinear elimination preconditioned inexact Newton method for heterogeneous hyperelasticity	SISC
2019	Luo, Shiu, Chen & C	A nonlinear elimination preconditioned inexact Newton method for blood flow problems in human artery with stenosis	JCP
2023	Luo & C	Preconditioned inexact Newton with learning capability for nonlinear system of equations	SISC

Selected bibliography: school of Axel Klawonn

2014	K, Lanser & Rheinbach	Nonlinear FETI-DP and BDDC methods	SISC
2015	K, Lanser & Rheinbach	Toward extremely scalable nonlinear domain decomposition methods for elliptic partial differential equations	SINUM
2017	K, Lanser, Rheinbach & Uran	Nonlinear FETI-DP and BDDC methods: a unified framework and parallel results	SISC
2018	K, Lanser & Rheinbach	Nonlinear BDDC methods with approximate solvers	ETNA
2020	Heinlein & Lanser	Additive and hybrid nonlinear two-level Schwarz methods and energy minimizing coarse spaces for unstructured grids	SISC
2022	K, Lanser & Uran	Adaptive nonlinear elimination in nonlinear FETI-DP methods	Proc of DD26
2023	Heinlein, K & Lanser	Adaptive nonlinear domain decomposition methods with an application to the p-Laplacian	SISC
2024	K & Lanser	Efficient adaptive elimination strategies in nonlinear FETI-DP methods in combination with adaptive spectral coarse spaces	Proc of DD27

Selected bibliography: school of Martin Gander

2016	Dolean, G, Kheriji, Kwok & Masson	Nonlinear preconditioning: how to use a nonlinear Schwarz method to precondition Newton's method	SISC
2016	G	On the origins of linear and nonlinear preconditioning	Proc of DD23
2021	Chaouqui, G, Kumbhar & Vanzan	On the nonlinear Dirichlet-Neumann method and preconditioner for Newton's method	Proc of DD26
2021	McCoid & G	Cycles in Newton-Raphson preconditioned by Schwarz (ASPIN and its cousins)	Proc of DD26
2022	Chaouqui, G, Kumbhar & Vanzan	Linear and nonlinear substructured Restricted Additive Schwarz iterations and preconditioning	Numer Algs
2023	McCoid & G	Cyclic and chaotic examples in Schwarz-preconditioned Newton methods	Proc of DD27

$2 \times 2 \times 2 \times 2^{+} \times n$ categories of Schwarz preconditioning

- Linear and nonlinear
- Left and right
- Nonoverlapping and overlapping
- Additive and multiplicative (and hybrids of the two)
- Selection of partitions (by subdomains, by fields, by high residual points, by high Mach points, etc.)

Our 2024 JCP paper, in a special issue dedicated to the late Roland Glowinski, closes some gaps in the theory for overlapping multiplicative methods.

In Roland Glowinski special issue of J. Comp. Phys.

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Overlapping multiplicative Schwarz preconditioning for linear and nonlinear systems

Lulu Liu [a,*], Weiguo Gao [b], Han Yu [c], David E. Keyes [d]

[a] School of Mathematics and Statistics, Nanjing University of Science and Technology, Nanjing, 210094, China
[b] School of Mathematical Sciences and School of Data Science, Fudan University, Shanghai, 200433, China
[c] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, 210046, China
[d] Program in Applied Mathematics and Computational Science, King Abdullah University of Science and Technology, Thuwal, 23955-6900, Saudi Arabia

ARTICLE INFO

Keywords:
Nonlinear preconditioning
Schwarz methods
Newton methods for nonlinear algebraic systems

ABSTRACT

For linear and nonlinear systems arising from the discretization of PDEs, multiplicative Schwarz preconditioners can be defined based on subsets of the unknowns that derive from domain decomposition, field splitting, or other collections of conveniently solved subproblems, and are well established theoretically for nonoverlapping subsets. For overlapping subsets, establishing the equivalence of the preconditioned and original iterations is less trivial. We derive herein an explicit formulation for a variety of multiplicative Schwarz preconditioners including overlaps representative of interfacial and bulk coupling in multiphysics systems, thus extending theoretical support for the nonlinear multiplicative Schwarz preconditioned inexact Newton (MSPIN) algorithm to these classes. For nonlinear multiplicative Schwarz preconditioners with overlaps, we illustrate the performance through numerical experiments involving applications such as a shocked duct flow and a natural convection cavity flow. We begin with a broad introduction to nonlinear preconditioning to set the context for those new to the technique.

Scope for today

- A two-component algebraic problem to illustrate
- Simple PDE flow models to warm up
  - 1D and 2D transonic potential flow over an airfoil
  - 2D velocity-vorticity incompressible Navier-Stokes in a cavity
- 1D, 2D, and 3D two-phase porous media flow
- We blaze through derivations and problem posings
  - Please see references

Inexact Newton with Backtracking (INB)

We aim to find a solution x^*, such that $F(x^*) = 0$, starting from an initial guess x^0, where $x = (x_1, \ldots, x_n)^T$, $F = (F_1, \ldots, F_n)^T$, and $F_i = F_i(x)$. A new x^{k+1} can be computed via

$$x^{k+1} \leftarrow x^k + \lambda^k s^k, \tag{1}$$

where λ^k is the scalar step length, and step s^k satisfies

$$\|J^k s^k + F(x^k)\| \leq \eta^k \|F(x^k)\|. \tag{2}$$

Here $J^k = F'(x^k)$ is the Jacobian matrix, $\eta^k \in [0, 1)$ is a tolerance that determines how accurately the Jacobian system needs to be solved, and $\lambda^k \in [0, 1]$ is determined by line search on the merit function $\|F(x^k + \lambda^k s^k)\|^2$.

Nonlinear preconditioning

- **Left preconditioning:** solve "equivalent system" (same root) with better balanced nonlinearities

$$\mathcal{F}(x) = G(F(x)) = 0$$

  - refs: Additive Schwarz Preconditioned Inexact Newton (**ASPIN**), Cai & K (2002); Multiplicative Schwarz Preconditioned Inexact Newton (**MSPIN**), Liu & K (2015); Restricted Additive Schwarz Preconditioned Exact Newton (**RASPEN**), Dolean *et al.* (2016); Nonlinear Elimination Preconditioning Inexact Newton (**NEPIN**), Liu *et al.* (2022)
- **Right preconditioning:** start from a better initial guess by first correcting within a subset of equations:

$$F(y) = 0, y = G(x)$$

  - refs: Nonlinear **FETI-DP** and **BDDC**, Klawonn *et al.* (2014); Nonlinear Elimination (**INB-NE**), Hwang *et al.* (2015) & Luo *et al.* (2020)

Preconditioning: Linear/Nonlinear, Left/Right

Linear preconditioning

$$F(x) = Ax - b = 0$$

Left:

$$G(F(x)) = M^{-1}(Ax - b) = 0$$

- $\kappa\left(M^{-1}A\right)$ smaller than $\kappa(A)$
- $F(x) = 0$ and $G(F(x)) = 0$ have same solution
- Linear system is changed

Right:

$$AM^{-1}y = b,$$
$$y = Mx.$$

- Linear system is unchanged

Nonlinear preconditioning

$$F(x) = 0$$

Left:

$$G(F(x)) = 0$$

- $G(F(x))$ less nonlinearly stiff than $F(x)$
- $F(x) = 0$ and $G(F(x)) = 0$ have same solution
- Nonlinear system is changed

Right:

$$F(y) = 0,$$
$$y = G(x).$$

- Nonlinear system is unchanged

Caution, with apologies to Tolstoy*

"All linear problems are alike;
each nonlinear problem is nonlinear in its own way."

* see first line of *Anna Karenina* (1878)

Outline

Pedagogical example with two equations in two unknowns

$$F(x_1, x_2) = \begin{bmatrix} F_1(x_1, x_2) \\ F_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} (x_1 - x_2^3 + 1)^5 - x_2^5 \\ x_1 + 2x_2 - 3 \end{bmatrix} = 0 \quad (3)$$

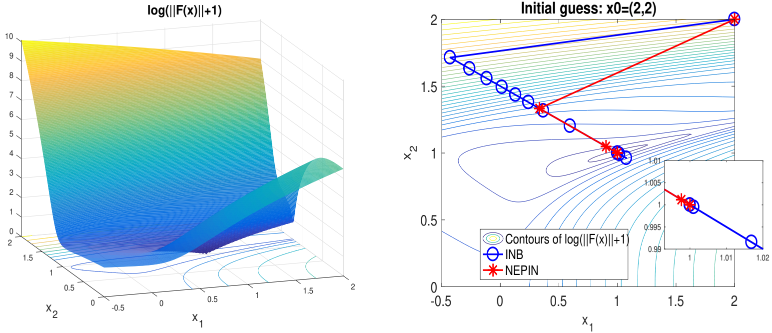


Figure: Contours of $\log(\|F(x)\| + 1)$ and path using Inexact Newton with Backtracking (INB) (blue circles, 13 steps) and Nonlinear Elimination Preconditioning Inexact Newton (NEPIN) (red stars, 4 steps) from same starting point $x^0 = [2, 2]^T$.

What happened?

Original system:

$$F^{\text{orig}}(x_1, x_2) = \begin{bmatrix} F_1(x_1, x_2) \\ F_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} (x_1 - x_2^3 + 1)^5 - x_2^5 \\ x_1 + 2x_2 - 3 \end{bmatrix} = 0 \qquad (4)$$

NEPIN-preconditioned system (analytically derived in this case):

$$F^{\text{NEPIN}}(x_1, x_2) = \begin{bmatrix} \tilde{F}_1(x_1, x_2) \\ F_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} x_1 - x_2^3 + 1 - x_2 \\ x_1 + 2x_2 - 3 \end{bmatrix} = 0. \qquad (5)$$

Table: Number of nonlinear iterations starting from four corners. Relative tolerance for the global Newton iterations is 10^{-8}.

	INB	NEPIN
$x_0 = [0, 0]^T$	10	6
$x_0 = [0, 2]^T$	12	4
$x_0 = [2, 0]^T$	7	6
$x_0 = [2, 2]^T$	13	4

Partitioning for preconditioning

Generalize from two equations to two subsets. The components of the nonlinear system $F(x) = 0$ are partitioned heuristically into two groups, "bad" and "good," labeled as F_b and F_g, resp., according to the degree of nonlinear "stiffness."

(This is often associated with the components whose residual exceeds some threshold, or at which some physical feature exceeds some threshold. Often the "bad" components are relatively few.)

The unknowns principally associated with each equation are split conformally into $x = [x_b, x_g]^T$:

$$F(x) = F(x_b, x_g) = \begin{bmatrix} F_b(x_b, x_g) \\ F_g(x_b, x_g) \end{bmatrix}, \tag{6}$$

where x_b and x_g are "bad" and "good" components, respectively.

NEPIN algorithm (2022): basic k^{th} step

1. Solve (inexactly) for the correction T_b to the "bad" unknowns in

$$F_b(x_b^{(k)} - T_b, x_g^{(k)}) = 0 \tag{7}$$

2. Form modified global residual by replacing "bad" component:

$$\tilde{F}^{(k)} = \begin{bmatrix} J_b(z^{(k)})T_b^{(k)} \\ F_g(x^{(k)}) \end{bmatrix}, \quad J_b(z^{(k)}) = R_b J(z^{(k)}) R_b^T. \tag{8}$$

where $z^{(k)} = (x_b^{(k)} - T_b, x_g^{(k)})^T$

3. Solve (inexactly) for the global Newton direction $s^{(k)}$ in

$$J(z^{(k)})s^{(k)} = -\tilde{F}^{(k)} \tag{9}$$

4. Update the global approximation:

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}. \tag{10}$$

How it works

For a given partitioning, the nonlinear elimination preconditioned function

$$\mathcal{F}(x) = \mathcal{F}(x_b, x_g) = \begin{bmatrix} T_b(x_b, x_g) \\ F_g(x_b, x_g) \end{bmatrix} = \begin{bmatrix} T_b(x) \\ F_g(x) \end{bmatrix}, \tag{11}$$

is obtained by solving the subsystem

$$F_b(x_b - T_b(x), x_g) = 0. \tag{12}$$

for $T_b(x)$. The Jacobian of $\mathcal{F}(x)$ can be written in the form of

$$\mathcal{J}(x) = \begin{bmatrix} \left(\frac{\partial F_b}{\partial u_b}\right)^{-1} & \\ & I_g \end{bmatrix} \begin{bmatrix} \frac{\partial F_b}{\partial u_b} & \frac{\partial F_b}{\partial x_g} \\ \frac{\partial F_g}{\partial x_b} & \frac{\partial F_g}{\partial x_g} \end{bmatrix}, \quad \text{where} \quad u_b = x_b - T_b(x). \tag{13}$$

Then the Newton correction step

$$\mathcal{J}(x)\hat{s} = \mathcal{F}(x) = \begin{bmatrix} T_b(x) \\ F_g(x) \end{bmatrix} \tag{14}$$

is equivalent, upon multiplying the upper block row through by $J_b = R_b J(u_b, x_g) R_b^T$, to

$$J(u_b, x_g)\hat{s} = \begin{bmatrix} J_b T_b(x) \\ F_g(x) \end{bmatrix} = \tilde{F}. \tag{15}$$

Example: 1D shocked duct flow*

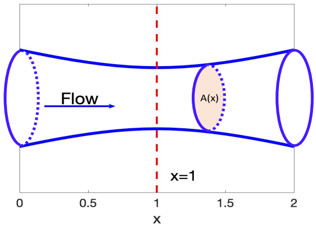
Consider inviscid, irrotational, compressible flow passing through a duct with variable cross-section area, a quasi-1D problem with model for velocity potential $\phi(x)$:

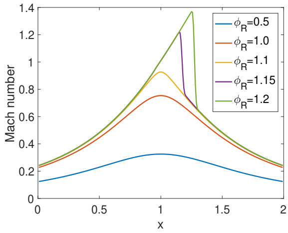
$$(A(x)\rho(\phi)\phi_x)_x = 0, \quad 0 < x < 2, \tag{16}$$

$$\phi(0) = 0, \ \phi(2) = \phi_R, \tag{17}$$

where the duct area and the density are given by

$$A(x) = 0.4 + 0.6(x-1)^2, \qquad \rho(\phi) = \left(1 + \frac{\gamma - 1}{2}(1 - \phi_x^2)\right)^{\frac{1}{\gamma - 1}}. \tag{18}$$





* Cai, K & Young, Boeing Technical Report (2001)

Example: 1D shocked duct flow ($\phi_R = 1.15$)

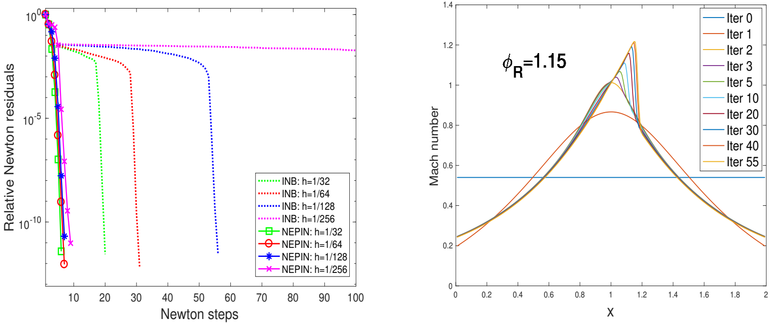


Figure: Left: history of the Newton residual using INB and NEPIN for increasing resolution with mesh size $h = \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}$. Right: history of Mach number profiles for the dotted blue residual plot (INB at $h = 1/128$). For $\phi_R = 1.15$.

Example: 1D shocked duct flow ($\phi_R = 1.15$)

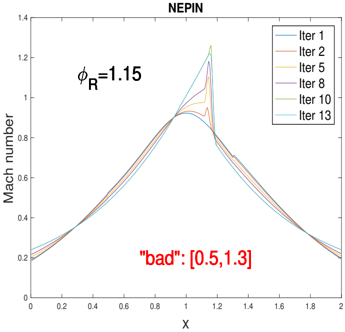


Figure: History of Mach number profiles for NEPIN. For $\phi_R = 1.15$, with "bad" unknowns in a fixed interval around the throat of the nozzle NEPIN requires 13 global Newton iterations, each of which requires solving approximately a nonlinear subproblem. INB without preconditioning requires 55 global Newton iterations.

Example: 2D transonic full potential flow

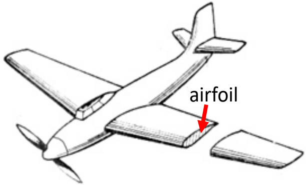
Consider transonic flow around an airfoil, which is described by the scalar full potential equation, derived for inviscid, irrotational, isentropic compressible flow as:

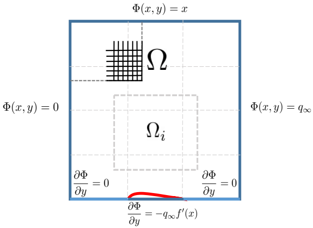
$$\nabla \cdot (\rho(\Phi)\nabla\Phi) = 0, \tag{19}$$

where Φ is the velocity potential, and $\nabla\Phi = [u, v]^T$ is the velocity field. The density function ρ is computed by

$$\rho(\Phi) = \rho_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty^2 \left(1 - \frac{\|\nabla\Phi\|_2^2}{q_\infty^2}\right)\right)^{\frac{1}{\gamma-1}} \tag{20}$$

with suitable upwinding, as in the Boeing TRANAIR code [Young *et al.*, JCP 1991]. The ∞ subscripts refer to the "freestream" far from the wing; here $\rho_\infty = q_\infty = 1.0$ and $M_\infty = 0.8$ (subsonic).





Example: 2D transonic full potential flow

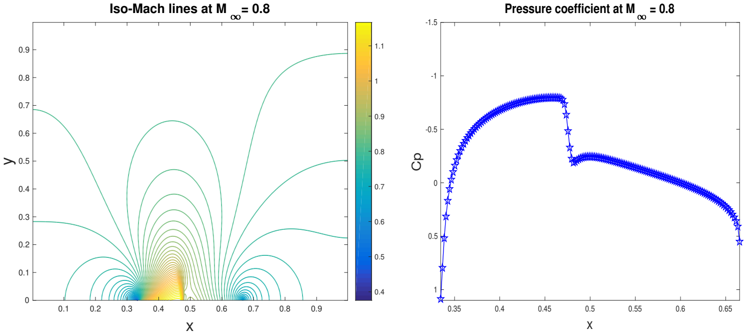


Figure: Left: Mach number countours. Right: pressure coefficient C_p curve (right) along the upper surface of the wing. For freestream transonic conditions $M_\infty = 0.8$ on a uniform 512×512 mesh.

Identification of the "bad" components

Define "bad" components as those where the local velocity exceeds a certain cut-off Mach number, $M(x, y) > M_c$.

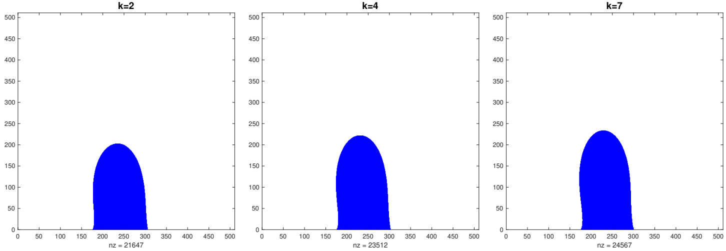
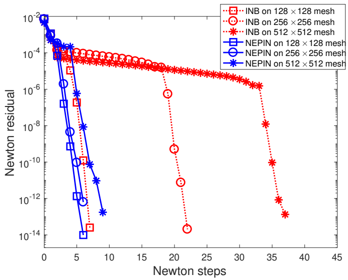
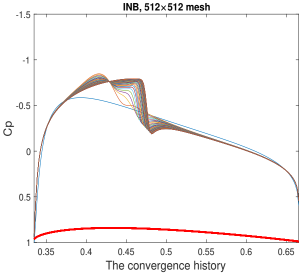
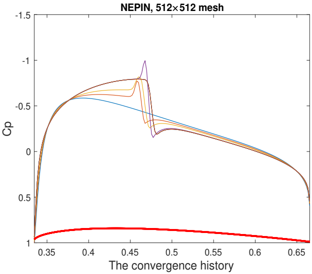


Figure: The evolution of the "bad" component region using NEPIN for $M_\infty = 0.8$ and $M_c = 0.82$, on a uniform 512×512 mesh, on the second, the fourth and the seventh global Newton iterations. Number of bad components: 21,647 at iteration 2; then 23,512 at iteration 4; then 24,567 at iteration 7.

Convergence history for varying mesh size ($M_c = 0.82$)







Outline

INB-NE algorithm (2011): basic k^{th} step

1. Solve (inexactly) for the correction T_b to the "bad" unknowns in

$$F_b(x_b^{(k)} - T_b, x_g^{(k)}) = 0 \tag{21}$$

2. Form a shifted starting point

$$z^{(k)} = (x_b^{(k)} - T_b, x_g^{(k)})^T \tag{22}$$

3. Solve (inexactly) for the global Newton direction $s^{(k)}$ in

$$J(z^{(k)})s^{(k)} = -F(z^{(k)}) \tag{23}$$

4. Update the global approximation:

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}. \tag{24}$$

Solving the "bad" equations for the "bad" components with the "good" fixed (Step 1) and updating the state vector (Step 4), are the same as in NEPIN.
The methods differ in modifying the residual function (NEPIN Step 2) versus the starting iterate (INB-NE Step 2) before the global correction (Step 3).

Example: two-phase flows in porous media

Darcy's law and saturation equations for incompressible two-phase flow:

$$\begin{cases} \mathbf{u}_\alpha = -\lambda_\alpha \mathbf{K}(\nabla p_\alpha - \rho_\alpha g \nabla D), & \alpha = w, n, \\ \phi \dfrac{\partial s_\alpha}{\partial t} + \nabla \cdot \mathbf{u}_\alpha = q_\alpha, & \alpha = w, n, \\ s_w + s_n = 1. \end{cases} \tag{25}$$

Subscripts w, n represent the wetting (water) and non-wetting (oil) phases.

Unknowns are velocities $\mathbf{u}_\alpha$ and saturations s_α for phases $\alpha = w, n$.

p_α, ρ_α, q_α are, respectively, pressure, density, and external source or sink for phases $\alpha = w, n$.

Constitutive assumptions, next slide...

Constitutive assumptions

- $\mathbf{K}$ and ϕ are the absolute permeability tensor and porosity of the porous media (possibly discontinuous and varying by orders of magnitude).
- The mobility function λ_α is given by $\lambda_\alpha(s_w) = k_{r\alpha}(s_w)/\mu_\alpha$, for relative permeability $k_{r\alpha}(s_w)$ and viscosity μ_α.
- The capillary pressure function $p_c(s_w)$ is the difference in the pressure of the two phases $p_c(s_w) = p_n - p_w$.
- D is the depth at position (x, y, z) and g is the magnitude of the gravitational acceleration.

The relative permeability and capillary pressure functions are given as (Van Duijn 1998, Hoteit 2008):

$$k_{rw}(s_w) = s_e^\beta, \quad k_{rn}(s_w) = (1 - s_e)^\beta,$$

$$p_c(s_w) = -\frac{B_c}{\sqrt{K}}\log(s_e), \quad s_e = \frac{s_w - s_{rw}}{1 - s_{rw} - s_{rn}}.$$

Initial and boundary conditions

Initial condition: a domain whose pores are saturated with oil

$$s_w|_{t=0} = s_w^0 \quad (= 0 \ \text{typically}), \qquad \text{in} \quad \Omega.$$

Boundary conditions:

Let $\partial\Omega = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_0$, where Γ_{in} denotes the inlet boundary, Γ_{out} denotes the outlet boundary, and $\Gamma_0 = \partial\Omega \setminus \{\Gamma_{in} \cup \Gamma_{out}\}$ is the no-flow boundary.

$$\begin{aligned} &\mathbf{u}_w \cdot \mathbf{n} = f_w^{in}, \ \mathbf{u}_n \cdot \mathbf{n} = f_n^{in}, && \text{on} \quad \Gamma_{in}, \\ &p_w = p_w^{out}, \ \lambda_n \mathbf{K} \nabla p_c \cdot \mathbf{n} = 0, && \text{on} \quad \Gamma_{out}, \\ &\mathbf{u}_w \cdot \mathbf{n} = 0, \ \mathbf{u}_n \cdot \mathbf{n} = 0, && \text{on} \quad \Gamma_0, \end{aligned}$$

where $\mathbf{n}$ is the unit outward normal vector, f_w^{in} and f_n^{in} are given flow rates at the inlet.

Discretization

The problem is discretized by a temporally implicit, spatially discontinuous Galerkin finite element method:

- Temporal discretization: backward Euler (first-order for now)
- Spatial discretization: Non-symmetric Interior Penalty Galerkin (NIPG) method (Epshteyn & Riviere, Appl. Numer. Math., 2007)

The fully implicit DG discretization results in a nonlinear algebraic system

$$F(x) = 0 \tag{26}$$

to be solved at each time step, where x is the vector of unknowns, which we may manipulate to be s_w and p_w. $F(x)$ is a highly nonlinear function, where the nonlinearities come from the relative permeability $k_{r\alpha}(s_w)$ and the capillary pressure function $p_c(s_w)$. Extra difficulties in solving (26) are induced by the heterogeneity of ϕ and K.

Why discontinuous Galerkin?

- Element-wise conservation
- Diagonal mass matrices
- Tends to have localized errors, allowing sharp *a posteriori* error indicators and effective adaptation
- Supports nonconforming spaces, incl. unstructured meshes, nonmatching structured meshes, variable degrees in adjacent elements, allowing general $h-$, $p-$, and $hp-$adaptivity
- Capable of exponential rates of convergence with appropriate meshing and element order
- Allows control of numerical diffusion
- Allows rough coefficients and captures discontinuities
- Robust and nonoscillatory in the presence of high gradients
- Low communication overhead in distributed memory ("halo" thickness does not depend upon order)

Software and Solution

LibMesh (**libmesh.github.io**) for the finite element construction

- piecewise quadratic for pressure
- piecewise linear for saturation

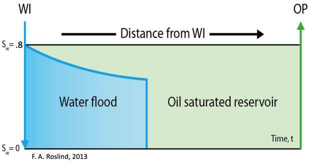
PETSc (**petsc.org**) for the algebraic solution

- relative tolerance for outer iteration global inexact Newton convergence on each timestep: 10^{-5}
- relative tolerance for inner inexact Newton convergence: 10^{-1}
- Jacobian and preconditioner evaluated only once per inner Newton iteration and reused
- GMRES restart dimension: 100
- overlap parameter for restricted additive Schwarz linear preconditioner: 1 cell
- subdomain preconditioner: ILU(2)

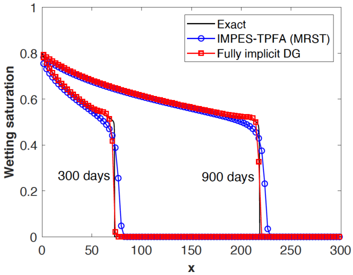
Shaheen-2 (**hpc.kaust.edu.sa**) for the scalable computation

- Cray XC-40 with dual socket Intel Haswell nodes and Aries dragonfly network
- ranked #7 in the Top 500 in 2015 (now #141)

Establishing the discretization: 1D Buckley-Leverett (1942)



Domain dimensions	300 m×1 m×1 m
Rock properties	$\phi = 0.2$, $K = 1$ mD
Fluid properties	$\mu_w = 2$ cP, $\mu_n = 3$ cP
	$\rho_w = \rho_n = 1000$ kg/m^3
Relative permeabilities	$\beta = 2$ in (2.8)
Capillary pressure	$\bar{B}_c = 0$ in (2.8)
Residual saturations	$s_{rw} = 0$, $s_{rn} = 0.2$
Injection rate	5×10^{-4} PV/day



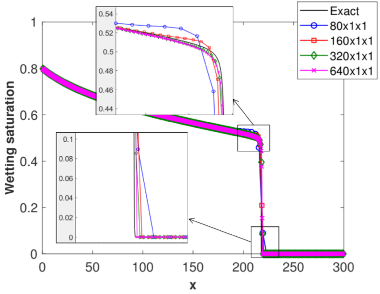


Figure: Left: Exact, IMPES (from MATLAB Reservoir Simulation Toolbox), and Fully Implicit DG at 300 and 900 days. Right: Exact and Fully Implicit DG at 900 days at three successive mesh doublings.

Elimination strategies for two-phase flow

- Multiplicative field-split approach. A strategy based on the field-splitting of pressure and saturation. Subspace correction is performed in two stages to eliminate the pressure components and the saturation components alternatively.
- Coupled element-block approach. A domain-splitting strategy that keeps pressure and saturation together within each domain. The strong nonlinearities of the system are often related to certain critical features that appear in certain local regions. Herein, when one component defined on a particular element is selected for pre-elimination, all other components associated with this element are also pre-eliminated.

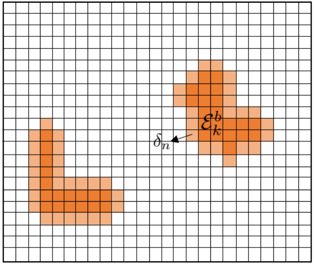


Figure: An example for the coupled element-block approach.

Example: 2D domain with obstacles

Domain dimensions	250 m×250 m×5 m
Rock properties	homogeneous: $\phi = 0.2$, $K = 100$ mD
	heterogeneous: $\phi \in [0.01, 0.48]$, $K \in [0.00177, 718.69]$ mD
Fluid properties	μ_w (cP)/μ_n (cP) = 1/1, 1/2, 1/4
	$\rho_w = 1025$ kg/m^3, $\rho_n = 849$ kg/m^3
Relative permeabilities	$\beta = 2$ in (2.8)
Capillary pressure	$\bar{B}_c = B_c/\sqrt{K}$ in (2.8), $B_c = 0 - 18$ bar·mD$^{1/2}$
Residual saturations	$s_{rw} = 0$, $s_{rn} = 0$
Injection rate	43.2 m^3/day
Production well	$r_w = 0.1$ m, $s_k = 0$, $p_{bh} = 1$ bar [9]

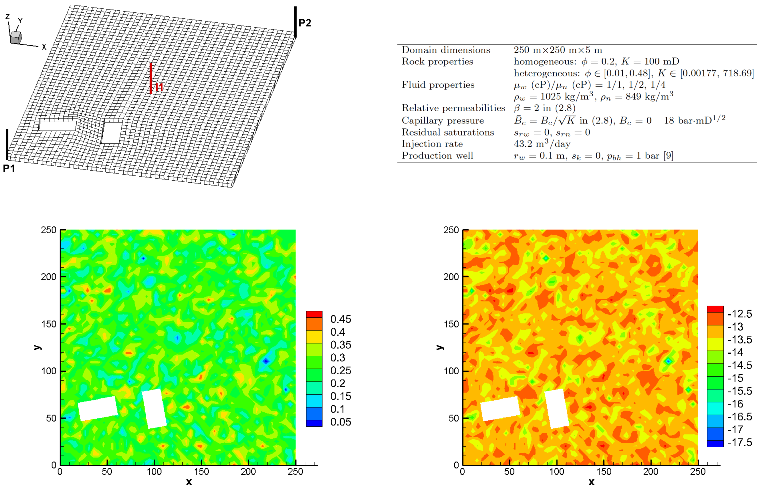
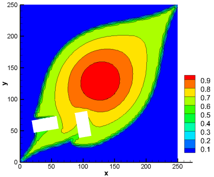
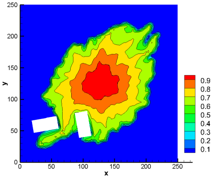


Figure: The mesh (top), porosity (bottom left), and $\log_{10} K$ permeability (bottom right) for a square horizontal domain with obstacles.

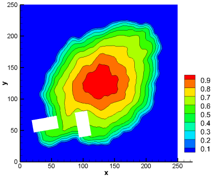
INB-NE solutions with various constitutive relationships



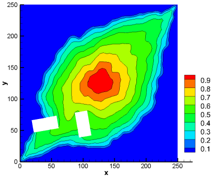
(a) Homo., $\mu_w/\mu_n = 1/2$, $B_c = 0$



(b) Hetero., $\mu_w/\mu_n = 1/2$, $B_c = 0$



(c) Hetero., $\mu_w/\mu_n = 1/2$, $B_c = 8$



(d) Hetero., $\mu_w/\mu_n = 1/4$, $B_c = 10$

Figure: Wetting saturation at 500 days for oil displacement.

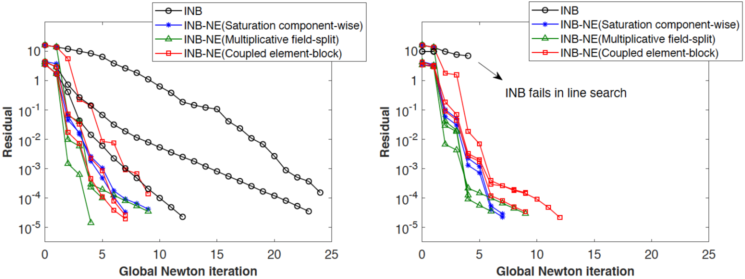
INB vs INB-NE for heterogeneous media

Table: The average numbers of iterations and compute times for heterogeneous case and $\mu_w/\mu_n = 1/2$, without and with capillary pressure, for middle two cases in previous plots.

Capillary effect	NI_g	LI_g	T_t	N_{NE}	NI_{NE}	LI_{NE}	T_{NE}
	INB						
No $(B_c = 0)$	23.6	8.8	4.47				
Yes $(B_c = 8)$	–	–	–				
	INB-NE (Multiplicative field-split)						
No $(B_c = 0)$	7.9	12.1	3.35	3.5	5.7	5.7	1.80
Yes $(B_c = 8)$	5.9	11.7	3.11	3.6	5.8	5.4	1.90
	INB-NE (Coupled element-block)						
No $(B_c = 0)$	8.9	13.4	3.24	3.0	4.6	2.3	1.39
Yes $(B_c = 8)$	10.4	12.2	3.58	3.1	5.4	2.8	1.49

'NI_g', avg # of global Newton iterations per time step
'LI_g', avg # of GMRES iterations per global Newton iteration
'T_t', total compute time in sec per time step
'N_{NE}', avg # of subspace correction steps in NE per time step
'NI_{NE}', avg # of Newton iterations per subspace correction in NE
'LI_{NE}', avg # of GMRES iterations per Newton iteration in NE
'T_{NE}' compute time in sec for all NE applications per time step.

Residual history



(a) Case 2 ($\mu_w/\mu_n = 1/2,\ B_c = 0$) (b) Case 3 ($\mu_w/\mu_n = 1/2,\ B_c = 8$)

Figure: Nonlinear residual history at the 1^{st}, 5^{th}, 10^{th} time steps, using INB and INB-NE for case $\mu_w/\mu_n = 1/2$. Left: $B_c = 0$. Right: $B_c = 8$.

Step length

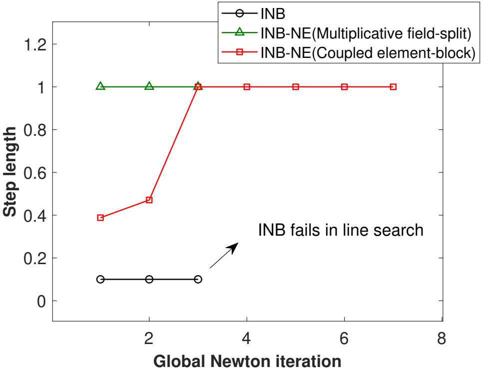
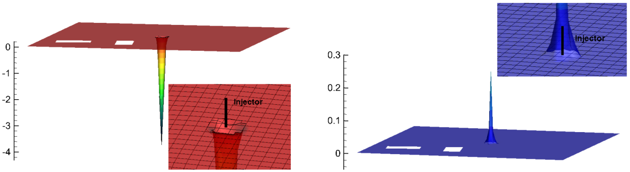
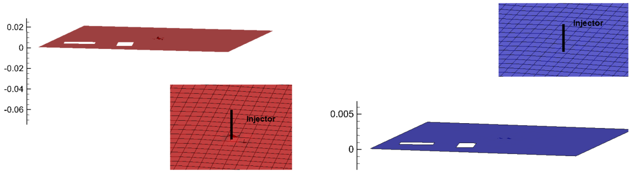


Figure: Step length λ^k at the 1^{st} time step, using INB and INB-NE for case $\mu_w/\mu_n = 1/2,\ B_c = 8$.

Before and after NE / multiplicative field-split



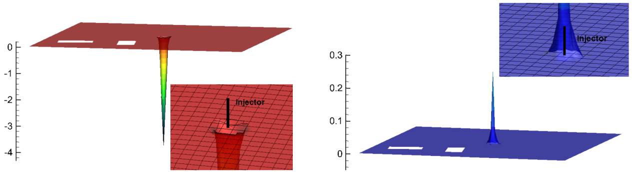
(a) before NE, 2^{nd} Newton, 1^{st} timestep



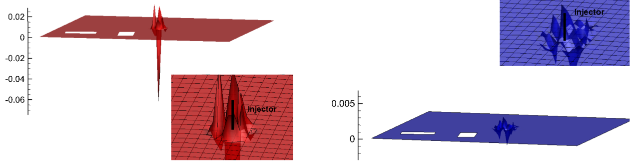
(b) after NE, 2^{nd} Newton, 1^{st} timestep – note scale change!

Figure: Left: saturation residual. Right: pressure residual.

Before and after NE / coupled element-block



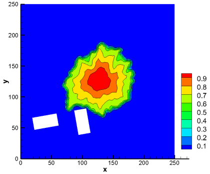
(a) before NE, 2^{nd} Newton, 1^{st} timestep



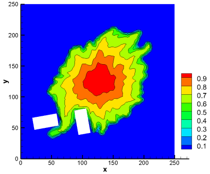
(b) after NE, 2^{nd} Newton, 1^{st} timestep – note scale change!

Figure: Left: saturation residual. Right: pressure residual.

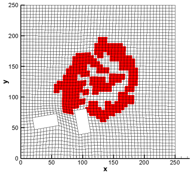
Element-block elimination



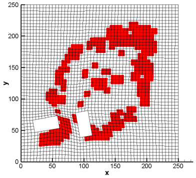
(a) Wetting saturation, 200^{th} step



(b) Wetting saturation, 400^{th} step



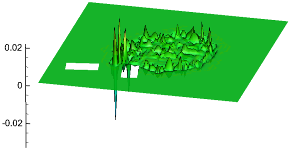
(c) Bad subset



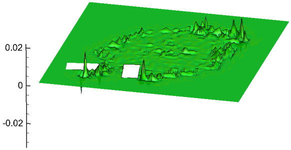
(d) Bad subset

Figure: Results at 2^{nd} global Newton iteration at the 200^{th} time step (left) and 400^{th} time step (right) for case $\mu_w/\mu_n = 1/2$, $B_c = 0$.

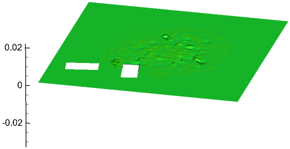
Residual of wetting saturation



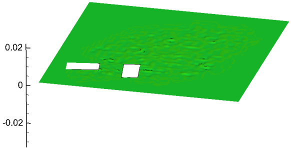
(a) before NE, 200^{th} step



(b) before NE, 400^{th} step



(c) after NE, 200^{th} step

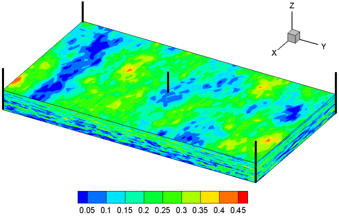


(d) after NE, 400^{th} step

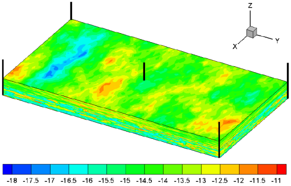
Figure: Results at the 2^{nd} global Newton iteration at the 200^{th} time step (left) and the 400^{th} time step (right), using INB-NE (Element-block) for case $\mu_w/\mu_n = 1/2,\ B_c = 0$.

Example: SPE10 setup

Domain dimensions	250 m$\times$250 m$\times$5 m
Rock properties	homogeneous: $\phi = 0.2$, $K = 100$ mD
	heterogeneous: $\phi \in [0.01, 0.48]$, $K \in [0.00177,\ 718.69]$ mD
Fluid properties	μ_w (cP)/μ_n (cP) = 1/1, 1/2, 1/4
	$\rho_w = 1025$ kg/m^3, $\rho_n = 849$ kg/m^3
Relative permeabilities	$\beta = 2$ in (2.8)
Capillary pressure	$\bar{B}_c = B_c/\sqrt{K}$ in (2.8), $B_c = 0 - 18$ bar$\cdot$mD$^{1/2}$
Residual saturations	$s_{rw} = 0$, $s_{rn} = 0$
Injection rate	43.2 m^3/day
Production well	$r_w = 0.1$ m, $s_k = 0$, $p_{bh} = 1$ bar [9]

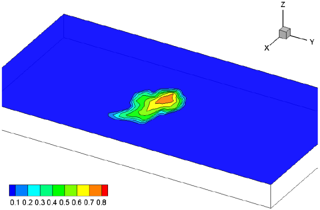


(a) Porosity

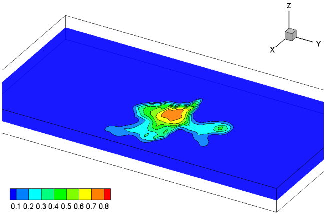


(b) Permeability ($\log_{10} K_{yy}$)

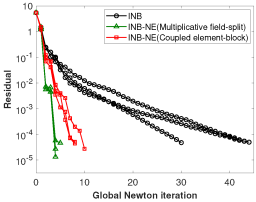
Example: SPE10 results



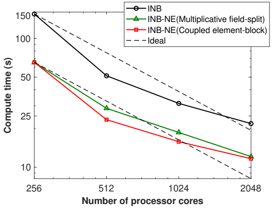
(c) Top layer sat at $t = 850$



(d) Middle layer sat at $t = 850$



(e) Residuals at the 6^{th}, 8^{th}, and 10^{th} time steps



(f) Strong scaling with 689,920 DOFs

Outline

Conclusions

- Nonlinear preconditioners can be effective in improving on the convergence of global Inexact Newton with Backtracking (INB) iterations.
    - They robustify Newton's method, often obviating the need for other globalization methods, by going to the core of the difficulty – removing a few components of the correction that require severe damping.
- Nonlinear preconditioning expends extra local computational cost for the solution of nonlinear subproblems to reduce the computation, communication, and synchronization costs of the global outer iterations, by reducing their number.

Future Work

- Better or more automated identification of "bad" components.
  - "Cascadic" ideas, wherein one starts with a small cluster of "bad" points and enlarges in a multistage nonlinear elimination process, have proved effective (Luo et al, SISC 2020).
  - Machine learning is proving effective (Luo et al, SISC 2023).
- Use of dynamic runtime systems to better allocate work in parallel implementations, in view of the imbalanced work of the inner nonlinear iterations.
  - Different preconditioning subproblems have different nonlinear difficulty.
  - Some outer iteration tasks (but not a synchronous full iteration!) can overlap with preconditioning subproblems.
- Fully asynchronous nonlinear iterations (within each time step for time-implicit problems or overall for steady-state problems).

New applications call for new ideas

"... at this very moment the search is on –
every numerical analyst has a favorite preconditioner,
and you have a perfect chance to find a better one."

– Gil Strang (1986), *Introduction to Applied Mathematics*

Your intuition is needed and your collaboration is invited!


Thank you!