Acceleration of Newton-like methods for nonlinear systems by preflattening techniques

Congrès National d'Analyse Numérique

June 2, 2026

Ngoc Do Quyen DANG

Supervisors: Quang Huy TRAN, Clément CANCÈS, Ibtihel BEN GHARBIA

Outline

Need for a faster nonlinear solver

- Challenge: solving large nonlinear algebraic systems derived from the numerical discretization of physical models
- Newton's method
  - Pros:
    - high convergence rate (superlinear or even quadratic) with the *sufficiently close* initials
  - Cons:
    - local convergence
    - slow convergence or failure: strong and unbalanced nonlinearities
- Goal: improving the effectiveness of Newton's method
- Existing techniques:
  - better starting points
  - globalization, line search, trust region methods
  - continuation and homotopy methods
  - nonlinear preconditioning

Nonlinear preconditioning

- Nonlinear system

$$F(\boldsymbol{u}) = 0$$

Left preconditioning	*Right preconditioning*
$G(F(\boldsymbol{u})) = G(0)$	$F(U(\boldsymbol{\tau})) = 0$
- Same unknown variables	- Change unknown variables
- Modified equations	- Same equations

- Pros:
  - ↑ robustness: Bassetto '21, Jonval '24
  - ↑ speed: Marelli '25
- Cons:
  - Linear systems: lower the *condition number* = faster iterative solvers: CG, GMRES, BiCGSTAB,...
  - Nonlinear systems: no equivalent quantity that governs convergence rate and that is decreased by preconditioning

New technique: Preflattening

- Affine function: Newton's method converges in a single iteration
- Paradigm: the closer F is to an affine function, the faster Newton's method converges
- Basic idea:
  - identifying new quantity: $\rho_F(\boldsymbol{u})$ called *flatness number*
  - designing an equivalent system at the current point s.t.

$$\rho_{F\circ U}(U^{-1}(\boldsymbol{u})) < \rho_F(\boldsymbol{u})$$

  $\Rightarrow$ preflattening technique
- Advantage: act exclusively at the nonlinear level of Newton's method
  - disentangling the two sources of difficulty
  - leaving the linear level to existing linear preconditioners

Outline

Flatness matrix - Flatness number

- Newton's *iteration function*

$$\Phi_F(\boldsymbol{u}) = \boldsymbol{u} - \nabla F(\boldsymbol{u})^{-1} F(\boldsymbol{u})$$

- The flatness matrix of F at $\boldsymbol{u} \in \Omega$ is the Jacobian matrix of the iteration function

$$\mathcal{J}_F(\boldsymbol{u}) := \nabla \Phi_F(\boldsymbol{u}) = \nabla F(\boldsymbol{u})^{-1} \nabla^2 F(\boldsymbol{u}) \nabla F(\boldsymbol{u})^{-1} F(\boldsymbol{u})$$

- Ostrowski's theorem: *spectral radius* of $\mathcal{J}_F$ at the solution $\boldsymbol{u}^*$ plays a crucial role for convergence
- The flatness number of F at $\boldsymbol{u} \in \Omega$ is spectral radius of the flatness matrix

$$\rho_F(\boldsymbol{u}) := \rho(\mathcal{J}_F(\boldsymbol{u}))$$

- $\rho_F(\boldsymbol{u})$: dimensionless and affine-invariant

Connection with Newton's rate of convergence

- Kantorovich '48
  - semi-local convergence of Newton's method
  - quadratic convergence rate governed by the product of 2 constants

$$L_{\mathrm{K}} = \sup_{\boldsymbol{u}} \|\nabla F(\boldsymbol{u}^0)^{-1}\nabla^2 F(\boldsymbol{u})\|, \quad \eta = \|\nabla F(\boldsymbol{u}^0)^{-1}F(\boldsymbol{u}^0)\|$$

  - $\rho_F(\boldsymbol{u}^0)$: lower bound of this product
- Hernández and Salanova '94
  - similar statements by a single constant $M = \sup_{\boldsymbol{u}} \|\mathcal{J}_F(\boldsymbol{u})\|$
  - $\rho_F(\boldsymbol{u}^0)$: lower bound of M
- Halley's method
  - high-order Newton's scheme with cubic convergence rate

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - \left[I - \tfrac{1}{2}\mathcal{J}_F(\boldsymbol{u}^k)\right]^{-1}\nabla F(\boldsymbol{u}^k)^{-1}F(\boldsymbol{u}^k)$$

  - the smaller the correction term $\frac{1}{2}\mathcal{J}_F(\boldsymbol{u}^k)$ is, the closer Halley is to Newton

A new approach: right-preflattening

- $\{U_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha}\in\mathcal{A}}$: a family of parametrization $\boldsymbol{u} = U_{\boldsymbol{\alpha}}(\boldsymbol{\tau})$
- $\mathcal{A}$ can be discrete or continuous
- Given the current iterate $\boldsymbol{u}^k$, we successively compute

$$
\begin{aligned}
\boldsymbol{\alpha}^k &= \arg\min_{\boldsymbol{\alpha}\in\mathcal{A}} \left\{\rho_{F\circ U_{\boldsymbol{\alpha}}}(U_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{u}^k))\right\}, && \text{[optimal preflattening]}\\
\boldsymbol{\tau}^k &= U_{\boldsymbol{\alpha}^k}^{-1}(\boldsymbol{u}^k), && \text{[current variable]}\\
\boldsymbol{\tau}^{k+1} &= \boldsymbol{\tau}^k - [\nabla_{\boldsymbol{\tau}}(F\circ U_{\boldsymbol{\alpha}^k})(\boldsymbol{\tau}^k)]^{-1}(F\circ U_{\boldsymbol{\alpha}^k})(\boldsymbol{\tau}^k)\\
&= \boldsymbol{\tau}^k - [\nabla_{\boldsymbol{\tau}} U_{\boldsymbol{\alpha}^k}(\boldsymbol{\tau}^k)]^{-1}[\nabla_{\boldsymbol{u}} F(\boldsymbol{u}^k)]^{-1}F(\boldsymbol{u}^k), && \text{[updated variable]}\\
\boldsymbol{u}^{k+1} &= U_{\boldsymbol{\alpha}^k}(\boldsymbol{\tau}^{k+1}), && \text{[primary unknown]}
\end{aligned}
$$

- Halley's method = right-preflattener in this framework

Outline

Chemical equilibrium problem

- Speciation problem: find the number of moles n_j of species C_j s.t.

$$\mathbf{A}\mathbf{n} = \mathsf{f}, \qquad \text{[element conservation]}$$
$$S^T \boldsymbol{\mu}(\mathbf{n}) = 0, \qquad \text{[equality of chemical potentials]}$$

- Expanded reformulation [Jonval, 2024]

$$\begin{aligned}
f_1 &:= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1\mathfrak{n}}x_{\mathfrak{n}} - b_1 = 0,\\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots\\
f_{\mathfrak{e}} &:= a_{\mathfrak{e}1}x_1 + a_{\mathfrak{e}2}x_2 + \ldots + a_{\mathfrak{e}\mathfrak{n}}x_{\mathfrak{n}} - b_{\mathfrak{e}} = 0,\\
f_{\mathfrak{e}+1} &:= s_{11}\ln x_1 + s_{12}\ln x_2 + \ldots + s_{1\mathfrak{n}}\ln x_{\mathfrak{n}} - t_1 = 0,\\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots\\
f_{\mathfrak{n}} &:= s_{\mathfrak{r}1}\ln x_1 + s_{\mathfrak{r}2}\ln x_2 + \ldots + s_{\mathfrak{r}\mathfrak{n}}\ln x_{\mathfrak{n}} - t_{\mathfrak{r}} = 0
\end{aligned}$$

- Difficulty: as $x_i \to 0^+$, the derivatives of $\ln x_i$ blow up

Equivalent formulations

- The log-trick model $G(\boldsymbol{y}) = 0$ is defined by replacing

$$\ln x_i \leftarrow y_i \quad \text{and} \quad x_i \leftarrow \exp y_i$$

  - Log-trick algorithm: Newton's method on G
  - Drawback: explosion of derivatives of $\exp y_i$ as $y_i \to +\infty$
- The parameterized model $H(\boldsymbol{\tau}) = 0$ is established by replacing

$$\ln x_i \leftarrow Y_i(\tau_i) \quad \text{and} \quad x_i \leftarrow X_i(\tau_i)$$

- Parametrization $(X_i(\tau_i), Y_i(\tau_i))$ depends on each method
  - Enhance robustness
  - Speed up convergence

Brenner-Cancès (BC) method

- Objective: guarantee robustness in all solution regimes
- BC parametrization

$$(X_i(\tau_i), Y_i(\tau_i)) = \begin{cases} (\exp \tau_i, \tau_i) & \text{if } \tau_i < 0 \\ (\tau_i + 1, \ln(\tau_i + 1)) & \text{if } \tau_i \geq 0 \end{cases}$$

- Procedure: Start from (x^0, y^0),
  - compute parameter $\boldsymbol{\tau}^0$
  - run Newton on H to update $\boldsymbol{\tau}$ until the stopping criteria hold
- Robust by construction

Continuous Dynamic (CD) method

- Motivation: Using a larger family of parametrizations to better minimize the flatness number
- CD parametrization

$$X_{\alpha_i}(\tau_i) = (1 + (1-\alpha_i)\tau_i)^{\frac{1}{1-\alpha_i}}$$

$$Y_{\alpha_i}(\tau_i) = \frac{1}{1-\alpha_i} \ln\left(1 + (1-\alpha_i)\tau_i\right)$$

- Minimize an upper bound $\widehat{\rho}_{H_\alpha} := \left\| P^{-1} \mathcal{J}_{H_\alpha}(\boldsymbol{\tau}) P \right\|_{\text{Frob}}$ to find optimal values for α_i
- Numerical observations:
  - BC is robust but sometimes slower than CD
  - CD is less robust but sometimes faster than BC

Hybrid method

- Aim: taking advantage of BC's robustness and CD's local rapidity
- Idea: use CD only when the upper bound of the flatness number $\widehat{\rho}_{H_\alpha}$ is *small enough*
- Procedure: At each iteration k,
  - Given $\left(x_i^k, y_i^k\right)$ with $y_i^k = \ln(x_i^k)$
  - Compute $\widehat{\rho}_{H_\alpha}^k$
  - Run BC or CD:
    - If $\widehat{\rho}_{H_\alpha}^k > \mathfrak{a}$, run BC,
      where $\mathfrak{a}$ is prescribed threshold, e.g., $\mathfrak{a} = 0.5$
    - Otherwise, run CD

Numerical data

- Test cases
  - H_2O: 3 species, 2 elements, 1 reaction
  - NaCl: 8 species, 4 elements, 4 reactions
  - Seawater: 37 species, 10 elements, 27 reactions
- Stopping conditions

$$\|H(\tau)\| < \varepsilon = 10^{-10}, \qquad \texttt{max\_iter} = 100$$

- High-precision stopping conditions

$$\|H(\tau)\| < \varepsilon = 10^{-150}, \qquad \texttt{max\_iter} = 100$$

- Methods
  - **Log-trick**: a state-of-the-art technique in industry
  - BC: a technique preventing the blow-up of derivatives
  - CD: a technique that is integrated with preflattening
  - Hybrid: a combination of BC and CD methods

Results of NaCl case

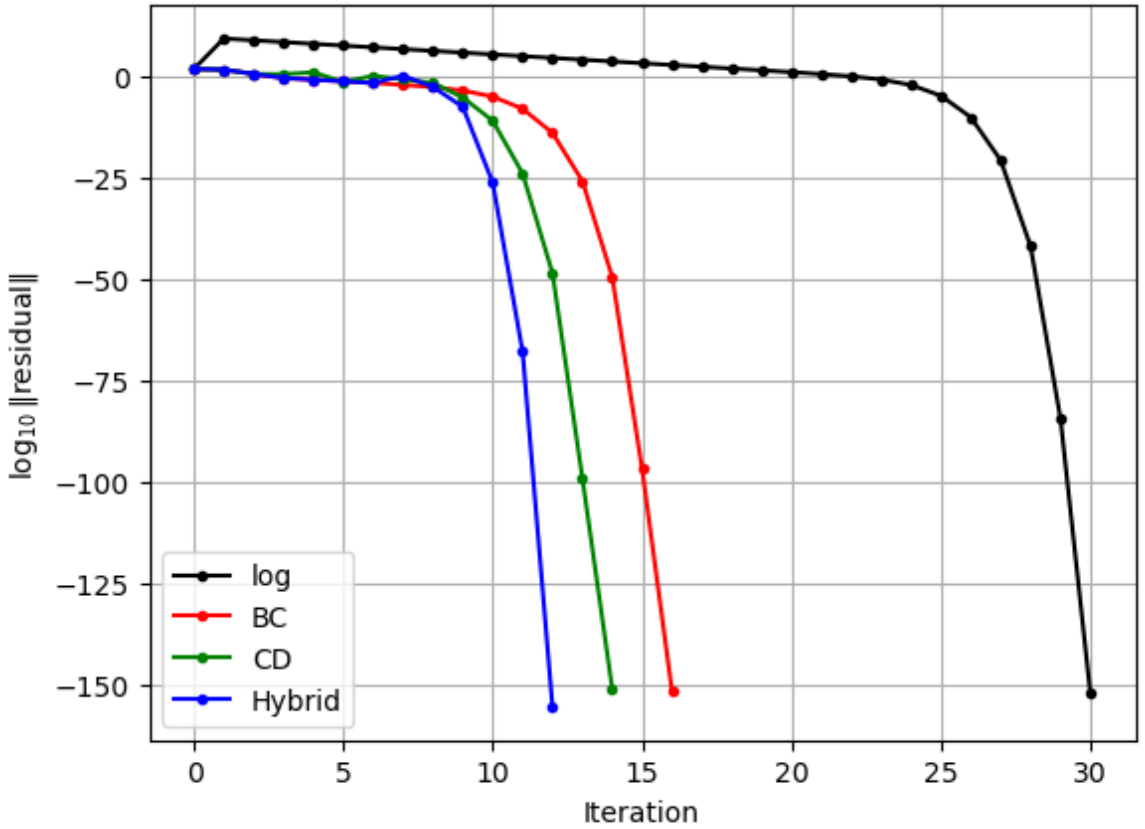
- Number of iterations

Initial guess		log	BC	CD	Hybrid
n_1^0	$n_{j\neq1}^0$				
55	55	26	12	13	10
55	1	14	12	11	10
55	0.1	11	11	11	10
55	10^{-2}	12	12	12	11
55	10^{-4}	x	30	x	27
55	ε_{32}	x	16	x	16

- Remarks
  - CD is less robust but sometimes faster than BC
  - Hybrid is the fastest algorithm

Evolution of residuals in NaCl case

- High-precision figure



- CD and Hybrid methods
  - Speed up Newton's convergence
  - Not produce results as impressive as those of the scalar counterpart

Outline

Elimination strategy

- Problem: unbalanced nonlinearities $\Rightarrow$ Newton's slowness
- Idea: eliminate "bad" unknowns from corresponding "bad" equations
- Splitting of good/bad variables: $\boldsymbol{\tau} = (\boldsymbol{\tau}_g, \boldsymbol{\tau}_b)$ and equations

$$H(\boldsymbol{\tau}) = 0 \quad \Leftrightarrow \quad \begin{array}{l} H_g(\boldsymbol{\tau}_g, \boldsymbol{\tau}_b) = 0 \\ H_b(\boldsymbol{\tau}_g, \boldsymbol{\tau}_b) = 0 \end{array}$$

- Elimination scheme
  - From bad equations $H_b = 0$, extract bad unknowns as functions of good ones, i.e., $\widetilde{\boldsymbol{\tau}}_b = \varphi_b(\boldsymbol{\tau}_g)$
  - Apply Newton to the reduced system $H_g(\boldsymbol{\tau}_g, \varphi_b(\boldsymbol{\tau}_g))$
- Difficulty: complicated calculation due to a Schur complement

Equivalent two-step procedure

- *2-step formulation*: at each iteration k
  - Projection: solve for $\widetilde{\boldsymbol{\tau}}_b^k = \varphi_b(\boldsymbol{\tau}_g)$ the *inner* system

$$H_b(\boldsymbol{\tau}_g^k, \widetilde{\boldsymbol{\tau}}_b^k) = 0$$

    and set the projected state as

$$\widetilde{\boldsymbol{\tau}}^k = \varphi(\boldsymbol{\tau}^k) := (\boldsymbol{\tau}_g^k, \widetilde{\boldsymbol{\tau}}_b^k) = (\boldsymbol{\tau}_g^k, \varphi_b(\boldsymbol{\tau}_g^k))$$

  - Linearization: perform one global *outer* Newton iteration starting from $\widetilde{\boldsymbol{\tau}}^k$, i.e.,

$$\boldsymbol{\tau}^{k+1} = \widetilde{\boldsymbol{\tau}}^k - \nabla H(\widetilde{\boldsymbol{\tau}}^k)^{-1} H(\widetilde{\boldsymbol{\tau}}^k)$$

- Advantage: dynamic partitioning into good/bad subset [Brenner et al. 2026]
- Challenge:
  - PDEs: obvious to associate a “bad” unknown with a “bad” function
  - Chemical systems: no notion of mesh

Connection between equations and unknowns

- Linear normalized residues

$$(\xi_1, \xi_2, \ldots, \xi_{\mathfrak{n}})^T = \Xi(\bar{\boldsymbol{y}}; \boldsymbol{\tau}) := \nabla G(\bar{\boldsymbol{y}})^{-1} H(\boldsymbol{\tau}),$$

  where Jacobian matrix of log-trick formulation ∇G is computed at the current state $\bar{\boldsymbol{y}}$

- Benefits of considering $\Delta(\boldsymbol{\tau}) = 0$
  - Equivalent to $H(\boldsymbol{\tau}) = 0$
  - Same orders of magnitude as those of unknowns
  - Easy to link with "bad" unknowns: choose the same indices for bad equations and bad unknowns

How to choose bad equations?

- Techniques inspired from statistics
  - Clustering: k-means, IQR, MAD...
  - Heuristic criteria: median, max distance...
- Novel approach based on flatness
  - Upper bound

$$\widehat{\rho}_{H_\alpha} = \left\| P^{-1} \mathcal{J}_{H_\alpha}(\boldsymbol{\tau}) P \right\|_{\text{Frob}} = \left(\sum_{j=1}^{\mathbf{n}} \Gamma_j \right)^{1/2}$$

where

$$\Gamma_j = \| [\mathcal{J}_{H_\alpha}]_{j\text{-th column}} \|_2^2$$

  - Idea: cancel those columns j that have largest norms Γ_j

$$\mathcal{B} = \{ j \mid \Gamma_j \text{ most significantly contributes to } \widehat{\rho}_{H_\alpha} \}$$

Numerical tests

- Test cases
  - H_2O: 3 species, 2 elements, 1 reaction
  - NaCl: 8 species, 4 elements, 4 reactions
  - Seawater: 37 species, 10 elements, 27 reactions
- Stopping conditions

$$\|H(\tau)\| < \varepsilon = 10^{-10}, \qquad \texttt{max\_iter} = 100$$

- Methods
  - BC: a technique proposed in Jonval's thesis to enhance the robustness
  - Hybrid: a combination of BC and CD methods
  - CDe: an integration of elimination into the CD method

The necessity of preflattening

- H_2O test case
- Algorithms integrated with elimination
  - CDe: elimination + CD method (with prefattening)
  - BCe: elimination + BC method (without prefattening)
- Number of iterations

Initial guess		BC	Hybrid	BCe	CDe
n_2^0	n_3^0				
10^{-2}	$2 \cdot 10^{-2}$	7	4	4	2
10^{-2}	$5 \cdot 10^{-2}$	7	4	4	2
10^{-4}	$2 \cdot 10^{-4}$	7	4	4	2
10^{-4}	$5 \cdot 10^{-4}$	7	4	4	2
ε_{32}	$2\varepsilon_{32}$	7	4	4	2
ε_{32}	$5\varepsilon_{32}$	7	4	4	2

- CDe is a better than BCe

  $\Rightarrow$ optimizing the flatness number is essential

Seawater test case

- Complexity

$$\mathfrak{m}^2 \times (\text{inner iterations}) + \mathfrak{n}^2 \times (\text{outer iterations})$$

- Number of iterations

Initial guess		BC	Hybrid	CDe
$\mathfrak{n}_1^0$	$\mathfrak{n}_{j\neq 1}^0$			
55	55	22	22	**7**
55	1	19	16	**6**
55	0.1	17	14	**5**
55	10^{-2}	16	14	**5**
55	10^{-4}	27	27	**5**
55	ε_{32}	29	27	**5**

Number of outer iterations

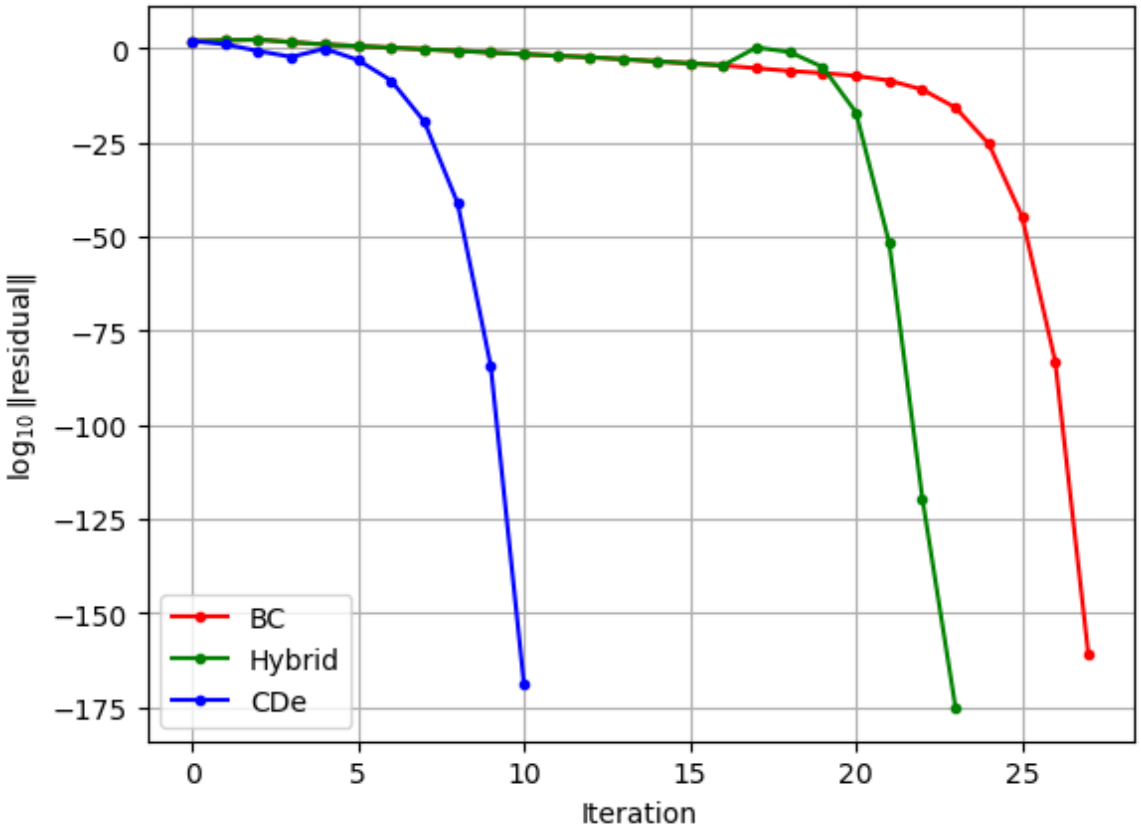
Initial guess		BC	Hybrid	CDe
$\mathfrak{n}_1^0$	$\mathfrak{n}_{j\neq 1}^0$			
55	55	31768	31768	**21921**
55	1	27436	23104	**19322**
55	0.1	24548	20216	**16206**
55	10^{-2}	23104	20216	**13787**
55	10^{-4}	38988	38988	**13743**
55	ε_{32}	41876	38988	**24693**

Complexity

- Remarks
  - CDe outperforms the others
  - CDe requires the fewest iterations, always less than 10

Evolution of residuals in Seawater case

- High-precision figure



- BC and Hybrid experience a quite long plateau before decreasing
- CDe is the fastest algorithm

Outline

Key results and future works

- Conclusions
  - Flatness number is a promising measure of the nonlinearity of a function
  - Preflattening is a technique used to reduce the flatness number
  - Elimination helps accelerate the convergence rate of Newton's method, while minimizing the flatness number is also indispensable
- Future works
  - Discretization of PDEs in geosciences: Richards equation
  - Theoretical framework for the new approach

Thank you for listening