TU Delft

Delft University of Technology

Nonlinear solver based on trust region approximation for CO_2 utilization and storage in subsurface reservoir

Pour, Kiarash Mansour; Voskov, Denis; Bruhn, David

DOI
10.1016/j.geoen.2023.211698

Publication date
2023

Document Version
Final published version

Published in
Geoenergy Science and Engineering

Citation (APA)
Pour, K. M., Voskov, D., & Bruhn, D. (2023). Nonlinear solver based on trust region approximation for CO_2 utilization and storage in subsurface reservoir. *Geoenergy Science and Engineering*, *225*, Article 211698. https://doi.org/10.1016/j.geoen.2023.211698

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

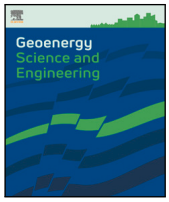
This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

225 (2023) 211698

Contents lists available at ScienceDirect

Geoenergy Science and Engineering

journal homepage: www.sciencedirect.com/journal/geoenergy-science-and-engineering



Nonlinear solver based on trust region approximation for CO_2 utilization and storage in subsurface reservoir

Kiarash Mansour Pour [a], Denis Voskov [a,b,*], David Bruhn [a,c]

[a] Department of Geoscience and Engineering, TU Delft, Delft, Netherlands
[b] Department of Energy Resources Engineering, Stanford University, CA, USA
[c] GFZ German Research Centre for Geosciences, Helmholtz Center Potsdam, Postdam, Germany

ARTICLE INFO

Keywords:
Compositional simulation
Trust-region nonlinear solver
Operator based linearization
Nonlinear analysis
Gas injection
CCUS

ABSTRACT

Simulation of CO_2 utilization and storage (CCUS) in subsurface reservoirs with complex heterogeneous structures requires a model that captures multiphase compositional flow and transport. Accurate simulation of these processes necessitates the use of stable numerical methods that are based on an implicit treatment of the flux term in the conservation equation. Due to the complicated thermodynamic phase behavior, including the appearance and disappearance of multiple phases, the discrete approximation of the governing equations is highly nonlinear. Consequently, robust and efficient techniques are needed to solve the resulting nonlinear system of algebraic equations. In this study, we present a powerful nonlinear solver based on a generalization of the trust-region technique for compositional multiphase flows. The approach is designed to embed a newly introduced Operator-Based Linearization technique and is grounded on the analysis of multi-dimensional tables related to parameterized convection operators. We split the parameter space of the nonlinear problem into a set of trust regions where the convection operators preserve the second-order behavior (i.e., they remain positive or negative definite). We approximate these trust regions in the solution process by detecting the boundary of convex regions via analysis of the directional derivative. This analysis is performed adaptively while tracking the nonlinear update trajectory in the parameter space. The proposed nonlinear solver locally constrains the update of the overall compositions across the boundaries of convex regions. We tested the performance of the proposed nonlinear solver for various scenarios. In many cases, our approach yields an improved behavior of the nonlinear solution in comparison to state-of-the-art solvers.

1. Introduction

Carbon emissions reduction has become a high priority as the world strives to mitigate global warming. Carbon dioxide capture, utilization, and storage (CCUS) is one of the promising technologies to minimize the amount of greenhouse gases entering the atmosphere. CCUS includes the capture of carbon dioxide and its associated compounds from producing sources, compression, transportation, and use of the captured CO_2 for operations such as permanent storage in deep underground geological formations and increased hydrocarbon recovery in existing oil fields. A model that captures multiphase compositional flow and transport is required to simulate CO_2 use and storage (CCUS) in subsurface reservoirs with complicated heterogeneous structures.

Compositional simulation is based on the solution of the discretized governing equations describing the mass, energy and momentum transfer in the reservoir. Explicit schemes have severe timestep size restrictions and are impractical for large-scale detailed reservoir models with Courant–Friedrichs–Lewy (CFL) numbers that vary by several orders of magnitude across the domain. As a result, in practice, the fully-implicit method (FIM) is preferred.

After the discretization of the governing Partial Differential Equations is complete, a nonlinear system needs to be linearized. The most frequently used sets of variables for linearization are based on natural (Coats, 1980) and molar formulations (Collins et al., 1992; Acs et al., 1985) which include phase-dependent or mass-dependent variables respectively. Typically, linearization is done using a version of the Newton-based method, which demands the assembly of the Jacobian and the residual for the combined system of equations. A previous timestep solution is used as an initial guess for the nonlinear solver. Due to the nonlinear nature of the equations and dependency on the initial guess, Newton's method is not guaranteed to converge for larger timesteps (Deuflhard, 2004). Once the solution of the linearized system is obtained, the nonlinear unknowns are updated and nonlinear iterations are repeated until convergence is achieved.

* Corresponding author.
E-mail address: d.v.voskov@tudelft.nl (D. Voskov).

https://doi.org/10.1016/j.geoen.2023.211698
Received 19 September 2022; Received in revised form 16 February 2023; Accepted 15 March 2023
Available online 17 March 2023
2949-8910/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Heuristic strategies are utilized to select timesteps in reservoir simulation practice (Aziz and Settari, 1979). The use of such heuristics often yields to timesteps that are either too conservative (i.e., small) or too large which in turn leads to wasted nonlinear iterations (Younis, 2011). The limitation of timestep selection can be overcome by applying an advanced nonlinear solution strategy.

There are several nonlinear solvers described in the literature for the compositional formulation. One of the promising ideas is the continuation method proposed by Younis (2011) that introduces a continuous parameter changing between 0 to 1 through the timestep. This approach controls the residual through continuous integration along the nonlinear trajectory in parameter space. Recently, Jiang and Pan (2022) introduced a Dissipation-Based Nonlinear Solver for compositional transport. By using numerical dissipation, the approach creates a homotopy of the discrete governing equations. A continuation parameter is included to limit dissipation and ensure that the converged solution's accuracy is not harmed.

Another approach is the flux-based trust region (TR) method for the natural formulation, proposed initially by Jenny et al. (2009) for two-phase immiscible flow with the S-shape fractional flow curves. Their work demonstrates that unconditional convergence is possible by limiting saturation updates based on the inflection point of the flux function. Later, Wang and Tchelepi (2013), Li and Tchelepi (2014) extended the flux-based trust region for two-phase immiscible flow and transport where buoyancy, capillary, and viscous forces are present. More recently, Møyner (2017) extended this work for black oil three-phase physics with a non-smooth flux function.

Even though different trust region nonlinear solvers were developed for the natural formulation, there is a lack of such strategies for the molar formulation. Voskov and Tchelepi (2008) demonstrated a compositional flow simulation employing tie-lines to parametrize the compositional space. A version of trust-region correction has been developed for molar formulation along the key tie-lines (Voskov and Tchelepi, 2011) but was not robust enough in comparison with techniques proposed for the natural formulation. Khebzegga et al. (2021) designed a nonlinear solver that detects phase boundaries. They focused their research on chopping at phase boundaries, ignoring the inflection line within the two-phase zone. They highlighted that detecting the inflection line for compositional problems required a second-order derivative and the Hessian analysis, which is more expensive and challenging to compute.

Recently, a new approach for the linearization of governing equations, called operator-based linearization (OBL), was proposed by Voskov (2017). In this approach, the exact physics of the simulation model was approximated using abstract algebraic operators. Later this technique was extended and implemented in the open-source Delft Advanced Research Terra Simulator (DARTS). DARTS is a scalable parallel modeling framework and aims to accelerate the simulation performance while capturing multi-physics processes in geo-engineering fields such as hydrocarbon (Khait and Voskov, 2017; Lyu et al., 2021a) geothermal (Wang et al., 2020) and CO_2 sequestration (Kala and Voskov, 2020; Lyu et al., 2021b). In the OBL approach, the parameterization is performed dependent on the conventional molar unknowns (pressure and overall composition). Using the OBL approach, the nonlinearity of the residual is translated into the operators. Consequently, by analyzing the nonlinearity of the operators, one can understand the major source of nonlinearity in the discrete residual equations. This greatly facilitates the design of a nonlinear solver for this framework.

In this work, we present an advanced nonlinear solver based on a generalization of the trust-region technique for compositional multiphase transport applied for CCUS. First, we investigate the nonlinearity of convective operators written in fractional flow form and detect boundaries of the trust region for the hyperbolic operator by assembling the directional approximation of the Hessian matrix. Next, we design the nonlinear solver in which we track the nonlinear trajectory for binary and ternary kernel in OBL parameter space and approximate these trust regions in the solution process via directional analysis of the derivative. By drawing some trial Newton trajectories on OBL parameter space, we observe that our directional analysis of derivatives predicts the boundaries of these trust regions correctly. Furthermore, it is less computationally expensive than computing the full Hessian matrix. Finally, we test the performance of the new nonlinear solver for several complex examples.

2. Modeling approach

In this section, we describe an operator form of governing equations used in Delft Advanced Research Terra Simulator (DARTS) framework (Khait, 2019).

2.1. Governing equations

We start from the description of the governing equations and nonlinear formulation for a general-purpose compositional simulation used in DARTS. The conservation equations for an isothermal multiphase compositional problem with n_p phases and n_c components can be written as:

$$\frac{\partial}{\partial t}(\phi \sum_{j=1}^{n_p} x_{cj}\rho_j s_j) + div \sum_{j=1}^{n_p} x_{cj}\rho_j \mathbf{v}_j + \sum_{j=1}^{n_p} x_{cj}\rho_j \tilde{q}_j = 0, \quad c = 1, \ldots, n_c. \tag{1}$$

Here, we introduce all variables in the equations as functions of spatial coordinate ξ and physical state ω :

- $\phi(\xi, \omega)$- porosity,
- $x_{cj}(\omega)$- the mole fraction of component c in phase j,
- $s_j(\omega)$ - phase saturations,
- $\rho_j(\omega)$- phase molar density,
- $\mathbf{v}_j(\xi, \omega)$- phase velocity,
- $\tilde{q}_j(\xi, \omega, u)$- phase rate per unit volume.

Darcy's law is applied to describe how each phase flows:

$$\mathbf{v}_j = -\left(\mathbf{K}\frac{k_{rj}}{\mu_j}(\nabla \mathbf{p_j} - \gamma_j \nabla \mathbf{d})\right), \tag{2}$$

where

- $\mathbf{K}(\xi)$ – permeability tensor,
- $k_{rj}(\omega)$ – relative permeability,
- $\mu_j(\omega)$ – phase viscosity,
- $\mathbf{p_j}(\omega)$ – vector of pressures in phase j,
- $\gamma_j(\omega)$ – phase gravity vector,
- $\mathbf{d}(\xi)$ – vector of depths (positive downwards).

By applying a finite-volume discretization on a general unstructured mesh and backward Euler approximation in time, we transform the conservation equations into

$$V\left((\phi \sum_j x_{cj}\rho_j s_j)^{n+1} - (\phi \sum_j x_{cj}\rho_j s_j)^n\right) - \Delta t \sum_j \left(\sum_{l \in L} x_{cj}^l \rho_j^l T_j^l \Delta \Psi^l\right) + \Delta t \sum_j \rho_p x_{cj} \overline{q_j} = 0, \tag{3}$$

where V is the volume of a control volume and $\overline{q_j} = \tilde{j}_j V$ the source of a phase. Here we assume Darcy's law neglecting capillarity and gravity and used a Two-Point Flux Approximation (TPFA) with upstream weighting introducing the summation over all interfaces L connecting the control volume with other grid blocks. Based on these simplifications, $\Delta \psi^l$ becomes a simple difference in pressures between blocks a and b, where T_j^l is phase transmissibility. These assumptions are not required by the proposed approaches, but help to simplify the further description.

2.2. Sources of nonlinearity

The main source of nonlinearity is related to the use of the Fully Implicit Method (FIM) for time approximation of the governing equations which requires the flux term in Eq. (3) to be defined based on of the nonlinear unknowns at a new timestep $(n+1)$. The closure assumption of instantaneous thermodynamic equilibrium further increases the nonlinearity. We used the overall molar formulation suggested by Collins et al. (1992). In this formulation, the following system must be solved at any grid block containing a multiphase (n_p) multi-component n_c mixture:

$$F_c = z_c - \sum_{j=1}^{n_p} v_j x_{cj} = 0, \tag{4}$$

$$F_{c+n_c} = f_{c1}(p,T,x_1) - f_{cj}(p,T,x_j) = 0, \tag{5}$$

$$F_{j+n_c*n_p} = \sum_{c=1}^{n_c} (x_{c_1} - x_{cj}) = 0, \tag{6}$$

$$F_{n_p+n_c*n_p} = \sum_{j=1}^{np} v_j - 1 = 0. \tag{7}$$

Here $z_c = \Sigma x_{cj}\rho_j s_j / \rho_j s_j$ is overall composition and $f_{cj}(p,T,x_{cj})$ is the fugacity of component c in phase j. The solution of this system is called a multiphase flash (Michelsen, 1982) and needs to be applied at every nonlinear iteration (Voskov and Tchelepi, 2012). The solution provides molar fractions for each component x_{cj} and phase fraction v_j.

2.3. Operator form

We can rewrite Eq. (3) as the component of a residual vector in general algebraic form. In this case, each term can be represented as a product of state-dependent and space-dependent operators. The resulting mass conservation equation, written for a control volume i in residual form, is

$$\begin{aligned} r_c(\omega) =& V(\xi)\phi_0(\xi)(\alpha_c(\omega) - \alpha_c(\omega^n)) \\ &- \sum_l \beta_c^l(\omega)\Lambda(\omega)\Delta t T^{ab}(\xi)(p^b - p^a) + \theta_c(\xi,\omega,u) = 0, \end{aligned} \tag{8}$$

where operators are defined as

$$\alpha_c(\omega) = (1 + c_r(p - p_{ref})) \sum_{j=1}^{n_p} x_{cj}\rho_j s_j, \tag{9}$$

$$\beta_c(\omega) = \frac{\sum_j x_{cj}\frac{k_{rj}}{\mu_j}\rho_j}{\Lambda} = \sum_j x_{cj} f_j \rho_j, \tag{10}$$

$$\Lambda(\omega) = \sum_j \frac{k_{rj}}{\mu_j}, \tag{11}$$

$$\theta_c(\xi,\omega,u) = \Delta t \sum_j \rho_j x_{cj} q_j(\xi,\omega,u). \tag{12}$$

Notice, that in this formulation, an additional operator Λ is introduced in comparison to one suggested in Voskov (2017). Here, c_r is the rock compressibility, T^{ab} is the geometric part of transmissibility (which involves permeability and the geometry of the control volume), and f_j is a fractional flow for phase j. The variables ω and ω^n are nonlinear unknowns at the current and the previous timestep respectively, and u is a vector of well control variables. The operator $\theta_c(\xi,\omega,u)$ is the influx/outflux term. In addition, ϕ_0, V_i, and p, are the initial porosity, volume, and pressure respectively.

The operator α_c is dependent on the properties of rock and fluid and independent of spatially distributed properties. Similarly, the divergence operator is present as a fluid-related operator β_c independent of spatially distributed properties. It should be noted that the Operator-based linearization framework is intended for molar formulation (based on an overall molar fraction, see Voskov (2017) for more details). Since the number of unknowns is changing in natural formulation due to the phase appearance and disappearance, the application of OBL based on interpolation in the parameter space with variable dimensionality is not straightforward.

3. Linearization and solution

In this section, we describe different types of linearization using the general algebraic form of governing equation.

3.1. Standard linearization approach

To solve nonlinear Eq. (8), we need to linearize them. The conventional approach in reservoir simulation is based on the application of the Newton–Raphson method. In each iteration of this method, we need to solve a linear system of equations of the following form:

$$\mathbf{J}(\omega^k)(\omega^{k+1} - \omega^k) = -\mathbf{r}(\omega^k), \tag{13}$$

where J is the Jacobian defined at nonlinear iteration step k.

The standard approach requires a sequential assembly of the residual and the Jacobian based on the numerical approximation of the analytic relations in Eqs. (9)–(12). This may demand a table interpolation (for standard PVT correlations or relative permeabilities), or a solution of the highly nonlinear equations (for EoS-based properties). Each property evaluation requires storage space for both values of the property and its derivatives with respect to the nonlinear unknowns. Most reservoir simulation software performs numerical (Pruess et al., 1997), analytic (Geoquest, 2011) or automatic differentiation (Garipov et al., 2018) of each property with respect to nonlinear unknowns.

3.2. Operator-based linearization

Operator-Based Linearization is a newly proposed strategy for the linearization of the reservoir simulation problem described by Eq. (8). As can be seen from the structure of each operator in Eqs. (9)–(12), this system is based on a complex combination of different nonlinear properties and relations. Since the space and time approximation is fixed, the discretization error depends on the variation of the timestep size Δt and the characteristic size of the mesh embedded in the T^{ab} term.

The operators α_c and β_c represent the physics-based terms. The accuracy of the nonlinear physics representation is controlled by these two operators (and a part of θ_c). In conventional linearization, we introduce all nonlinear properties and their derivatives into residual and Jacobian assemble. Next, the nonlinear solver tries to resolve all the details of the nonlinear description, struggling sometimes with unimportant features due to the numerical nature and some uncertainty in the property representations.

The Operator-Based Linearization (OBL) strategy, utilized in this work, is based on the simplified representation of the nonlinear operators α_c and β_c in the parameter-space of the simulation problem (Voskov, 2017). In this approach, we uniformly discretize the parameter space with a fixed number of points. Next, we apply multilinear interpolation in parameter space for the continuous representation of physics-based operators and the discrete representation of their derivatives. The number of points in the interpolation controls the accuracy of approximation of the nonlinear physics, which governs the process. This is similar to the accuracy of the approximation in space and time being controlled by the 3D grid and timestep size. The details of the OBL approach, test results and convergence analysis can be found in Khait and Voskov (2017) and Khait and Voskov (2018).

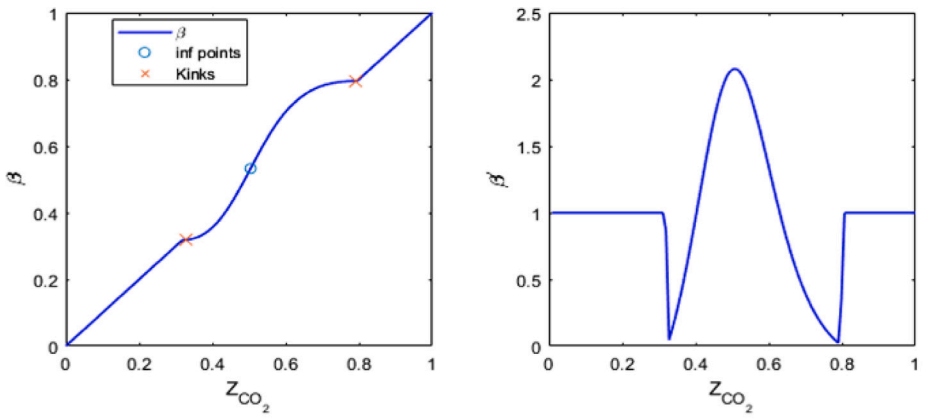


Fig. 1. Binary incompressible compositional kernel.

4. Nonlinear analysis of convective operator β

For simplicity, we assume in the following derivations that the system is incompressible, which limits the analysis to the convection operators β_c only as the main source of nonlinearity in the residual Eq. (8) written in total velocity formulation. As we can see in Fig. 1, for a binary compositional kernel, we have two kinks in addition to the inflection point of the fractional curve. These two points correspond to bubble and dew points compositions where the phase transition occurs. Kinks have different properties than inflection points and usually have a negative impact on nonlinear convergence (Li and Tchelepi, 2015). There is a discontinuity in derivative in the point of kinks and thus there is an abrupt change in concavity and residual.

Kinks and inflection points in parameter space dictate the boundaries of trust regions. For general multicomponent systems, we estimate the inflection point(s) based on the analysis of the Hessian of the convective operator. The Hessian matrix is a way of organizing all the second partial derivative information of a multivariable function. The general Hessian matrix for a convection operator can be written as:

$$\mathbf{H}(\omega) = \mathbf{J}\left(\nabla[\beta_c(\omega)]\right) = \begin{bmatrix} \frac{\partial^2\beta_1}{\partial\omega_1^2} & \frac{\partial^2\beta_1}{\partial\omega_1\partial\omega_2} & \cdots & \frac{\partial^2\beta_1}{\partial\omega_1\partial\omega_c} \\ \frac{\partial^2\beta_1}{\partial\omega_2\partial\omega_1} & \frac{\partial^2\beta_1}{\partial\omega_2^2} & \cdots & \frac{\partial^2\beta_1}{\partial\omega_2\omega_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\beta_1}{\partial\omega_c\partial\omega_1} & \frac{\partial^2\beta_1}{\partial\omega_c\partial\omega_2} & \cdots & \frac{\partial^2\beta_1}{\partial\omega_c^2} \end{bmatrix}. \tag{14}$$

In this work, we focus on the analysis of binary and ternary compositional problems and evaluate the Hessian matrix with respect to hyperbolic variables z_c using the finite difference method.

In our analysis of the ternary system, we are interested in the variation of convective operators with respect to z_1 and z_2. Accordingly, we construct the Hessian matrix for the fixed pressure as follows:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2\beta}{\partial z_1^2} & \frac{\partial^2\beta}{\partial z_1\partial z_2} \\ \frac{\partial^2\beta}{\partial z_2\partial z_1} & \frac{\partial^2\beta}{\partial z_2^2} \end{bmatrix}. \tag{15}$$

For ternary systems, we uniformly discretize the parameter space and compute the Hessian numerically as follows

$$\left(\frac{\partial^2\beta}{\partial z_1^2}\right)_{i,j} = \frac{(\frac{\partial\beta}{\partial z_1})_{i+1,j} - (\frac{\partial\beta}{\partial z_1})_{i,j}}{\Delta z}, \tag{16}$$

$$\left(\frac{\partial^2\beta}{\partial z_2^2}\right)_{i,j} = \frac{(\frac{\partial\beta}{\partial z_2})_{i,j+1} - (\frac{\partial\beta}{\partial z_2})_{i,j}}{\Delta z}, \tag{17}$$

$$\left(\frac{\partial^2\beta}{\partial z_2\partial z_1}\right)_{i,j} = \left(\frac{\partial^2\beta}{\partial z_1\partial z_2}\right)_{i,j} = \left(\frac{(\frac{\partial\beta}{\partial z_1})_{i,j+1} - (\frac{\partial\beta}{\partial z_1})_{i,j}}{\Delta z}\right), \tag{18}$$

where i and j corresponds to the coordinates of the hypercube centers for axes z_1 and z_2 respectively. Next, for each point in the centers at the interface of parameterized hypercubes, we define a quadratic form

$$Q = dz\mathbf{H}dz'. \tag{19}$$

After calculating Q for all points in the parameter space, we identify trust regions. If Q changes the sign from positive to negative, it indicates that our operator goes from positive definite to negative definite and changes its convex condition.

Fig. 2 shows the Hessian diagram for all three convection operators (β_1, β_2, β_3) and the phase diagram corresponding to that ternary kernel. In the phase diagram, the red color corresponds to the two-phase region and the blue color corresponds to the single-phase region. In the Hessian diagram, each color corresponds to different convex conditions of the flux operators. Comparing the Hessian diagram to the phase diagram, it is clear that there is an abrupt change in concavity (kink) on the boundaries between single-phase and two-phase regions. Moreover, there is an inflection line in the two-phase region for each component that segments the two-phase zone into a concave and a convex part.

All computations of special features in the phase diagram depends on the OBL resolution. Since all properties including convective operator β_i are defined based on interpolation among supporting points in the OBL mesh, any changes in convexity can be detected based on the numerical approximation of second derivatives. It is clear that these calculations as well as the performance of the nonlinear solver are dependent on the OBL resolution. In Appendix C, we show the sensitivity of inflection point definition and nonlinear convergence based on OBL resolution. More sensitivity analysis of the nonlinear convergence can be found in Voskov (2017) and Khait and Voskov (2018)

It is computationally expensive and time-consuming to compute the Hessian at each point in the parameter space. Checking the convexity condition only for the interfaces through which Newton's trajectory passes excludes the need to study the entire parameter space. However, the quadratic form of the operator must be evaluated using Eq. (19), which requires computation of the full Hessian matrix. The size of the Hessian matrix is $n_c \times n_c$, where n_c is the number of components. As a result, n_c^2 evaluations of first derivatives are required for hessian matrix assembly. To avoid calculating the entire hessian matrix, in the directional analysis of the second derivative, we only need to evaluate derivatives in the direction of the given interface that nonlinear solver passes. As an example, given the simple Newton trajectory in Fig. 3 that

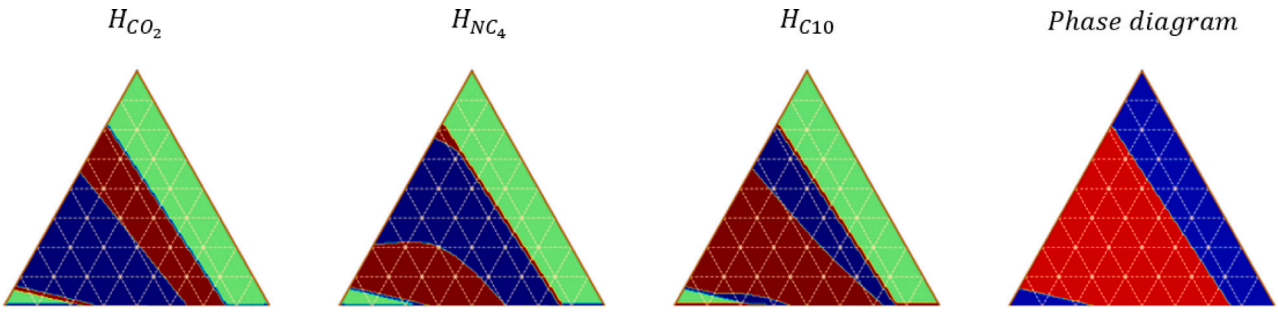


Fig. 2. Hessian for three convective operators and phase diagram for incompressible ternary kernel.

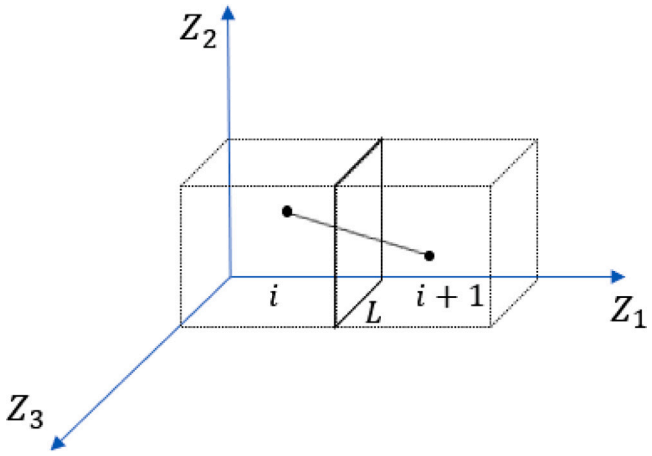


Fig. 3. Newton trajectory passing interface L between two cells in the parameter space.

passes interface L in the direction of Z_1, the second order directional second derivative is calculated as follows:

$$\partial^2_{Z_1}\beta(\omega) = \frac{(\frac{\partial\beta}{\partial z_1})_{i+1} - (\frac{\partial\beta}{\partial z_1})_i}{\Delta z_1}, \qquad \omega = [Z_1, \dots, Z_{nc}]. \tag{20}$$

The advantage of using Eq. (20) is that we only need to evaluate derivatives once to calculate the second-order directional derivative regardless of the dimension of the operator's space n_c.

5. Nonlinear solver for OBL framework

Before we present our Trust Region (TR) nonlinear Newton solver in detail, we provide an overview of the state-of-the-art damping strategies for Newton's solver. Newton update can be written at every nonlinear iteration in the general form as follows:

$$\Delta\omega = -\boldsymbol{\Phi}\mathbf{J}^{-1}\mathbf{r}, \tag{21}$$

Here $\boldsymbol{\Phi}$ is the diagonal matrix:

$$\boldsymbol{\Phi} = \mathrm{diag}(\phi_1, \phi_2, \dots, \phi_n), \phi_i \in [0, 1]. \tag{22}$$

Global and local nonlinear solvers can be seen as different methods to damp the Newton updates (Ortega and Rheinboldt, 1970) by specifying the diagonal matrix $\boldsymbol{\Phi}$:

- *STD* The standard Newton's solver where the updated matrix is the identity matrix.
- *Global chop* In the global-chop nonlinear strategy, all entries of the diagonal are identical, implying that the Newton direction is simply scaled by a constant factor.
- *Local chop* In the local-chop nonlinear solver, the diagonal scaling entries depend on a cell-by-cell basis to limit the local compositional update.

The benefit of global chop is that the nonlinear update direction is not changing. However, it could be quite restrictive due to a significant number of resulting nonlinear iterations. Local chop on the other hand is more effective due to the local adjustment of the update. However, it may lead to inconsistency in the nonlinear update due to the restrictions of the hyperbolic transport solution. Next, we will present a Trust Region (TR) solver which yields a consistent nonlinear update.

5.1. Trust region solver for OBL framework

In previous sections, the sources of nonlinear convergence failure, such as kinks and inflection lines encoded in the convective operator, were discussed. All of these features can be discovered by applying the Hessian analysis of the convective operator. With these kinks and inflection lines, the nonlinear space is divided by multiple subregions, which are referred to as trust regions (TR). The key idea is that any nonlinear update (in terms of composition) is not allowed to cross the boundary of any trust region too far. If an intersection is found, the nonlinear update's size is reduced until the updated composition barely crosses the boundary.

The full Hessian evaluation in the trust-region identification is a time-consuming procedure. To detect the trust regions more efficiently, we apply a directional analysis of the second derivative while tracking the nonlinear update and passing each interface in the OBL parametrization. The procedure is as follows:

1. detect OBL interfaces along the nonlinear update trajectory in the parameter space (see Appendix A for detailed algorithm detecting OBL interfaces based on tracking of nonlinear trajectory in parameter space),
2. computes directional second derivative for each convection operator at each crossing interface of OBL space with finite difference method,
3. detects inflection point(s) and kinks based on the second derivative information,
4. limits the local nonlinear update by the location inside the trust region.

Next, we illustrate our approach with examples and compare a calculation of special points detected by two types of analysis: reference results based on the construction of the full Hessian matrix for binary and ternary systems and directional analysis based on the second derivative only. In Fig. 4, we detect an inflection line position in the parameter space based on the convex condition approximated by the quadratic form of the Hessian matrix and compare it to the inflection point detected during tracking the nonlinear update. In this example, we use a two-phase immiscible physical kernel with viscosity dependent on pressure, which results in a pressure-dependent inflection line. Note that the exact location of the detection points can be lightly shifted from the line position due to the discrete approximation of the directional derivative. Another observation is that the pressure impact can be taken into consideration when detecting the inflection point using the directional derivative.

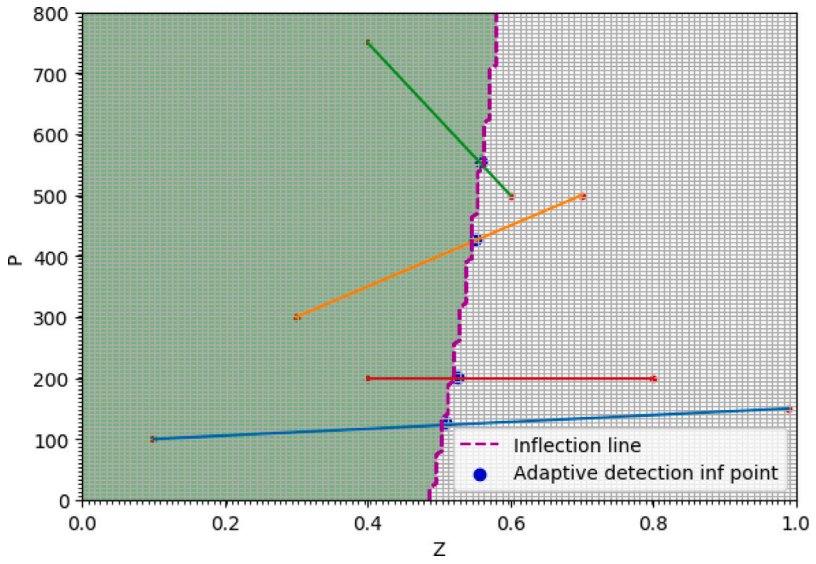
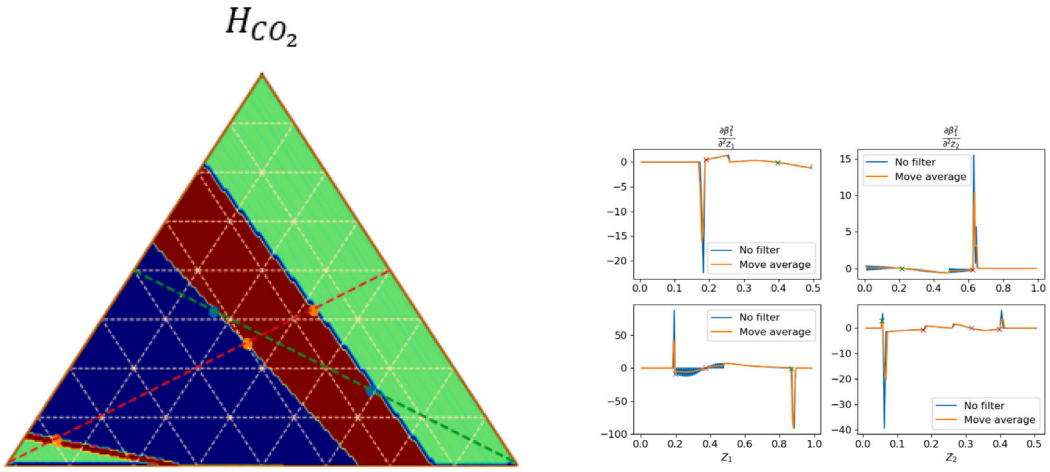


Fig. 4. Adaptive detection of inflection point versus the rigorous inflection line calculated by the full Hessian assembly for several random trajectories in parameter space.



(a) Trial trajectories on H_{CO_2}

(b) Directional derivatives of the random trajectories.

Fig. 5. Full and directional evaluations of second order behavior for ternary compositional kernel.

Similarly, for a ternary compositional kernel, we show in Fig. 5(a) that special points can be detected correctly using directional derivatives along the nonlinear update. However, some numerical artifacts and noise are usually present in the computation of numerical derivatives in directional analysis. To overcome this problem, we use the moving average algorithm (Gilgen, 2006) to smooth second-order directional derivatives as shown in Fig. 5(b).

6. Performance of nonlinear solver

In this section, the performance of the proposed nonlinear solver is demonstrated for several simplified and realistic modeling setups.

6.1. Single cell analysis

We can derive some essential conclusions for simulation problems by investigating the nonlinear behavior of a single-cell compositional transport problem. Fig. 6 shows our single-cell setup. Here our goal is to find the solution z^{n+1} from the initial guess $z^{n+1,0}$ for a given boundary condition on the left and right sides of the cell, defined as $\beta_{c_i L}$ and P_R, respectively. The imposed flux is from left to right. Here we inject pure gas mixture with the composition of (CO_2, C_{10}) = (1, 0) into the reservoir initialized with composition of (0, 1) with the K-values equivalent to $K = \{2.5, 0.3\}$ and the viscosity ratio of 10 between the oil and gas phase

We investigate the convergence map of the pure Newton's solver and trust region solver for binary compositional kernels. In this setup, we fixed the right boundary conditions as $z_r = 1$ and study the convergence of the solution using different nonlinear solvers for all possible starting points $(z^{n+1,0}, z_l) \in (0, 1) \times (0, 1)$. The maximum nonlinear iteration for all these test cases is equal to 50. We perform our analysis for two different dimensionless timesteps expressed as $C=\frac{V_t dt}{dx}$, with t in days and x in meters and V_t velocity in meters per seconds.

As seen in Figs. 7 and 8, once the solution is in the two-phase area, pure Newton's update struggles to converge in a maximum number of iterations. Another point to note is that once the solution is

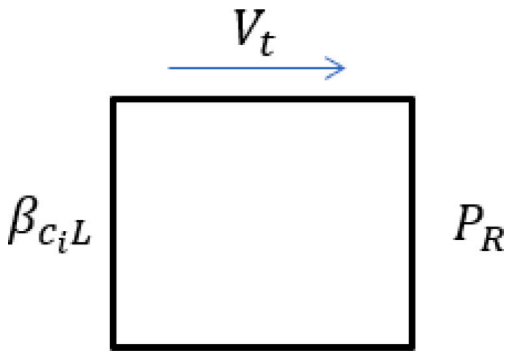


Fig. 6. Single-cell setup with left and right boundary condition. Flow is from left to right with fixed total velocity V_t and fixed total flux $\beta_{c_i L}$.

in the single-phase zone (linear fractional flow), Newton's approach guarantees nonlinear convergence. The proposed nonlinear solver, on the other hand, always has a limited number of nonlinear iterations.

As we increase the timestep in Fig. 8, we can observe that for pure Newton's update, the yellow region (corresponding to the non-convergence zone) increases again in the two-phase region. However, once the solution is in the single-phase region pure Newton is able to find the solution. And again, the proposed Trust Region solver is globally convergent for all initial guesses.

6.2. Front propagation in a single fracture

In fractured reservoirs, the speed of transport front propagation between matrix and fracture is significantly different due to the large contrast in permeabilities. Here, we imitate this process for a one-dimensional reservoir by running the simulation with small timesteps to develop a resolved displacement solution at a particular time. Next, we restart the simulation from this distribution for one control timestep only and account for the number of nonlinear iterations required to converge the solution. We repeat this procedure by gradually increasing the size of the control timestep and detecting the change in the number of nonlinear iterations. Fig. 9 illustrates CO_2 front propagation test case. This test case considers a 1000 m in length 1D homogeneous reservoir with constant permeability and porosity of $K = 10$ mD and $\phi = 0.3$.

Next, we use multiple physical kernels of increasing complexity to test the performance of a nonlinear solver based on directional analysis in the suggested numerical setup. We also compare our trust-region nonlinear method against global and local chop with $\Delta z = 0.1$ as the highest tolerable Δz.

For binary systems, we test the performance of nonlinear solvers for the flow of two-phase immiscible fluids and miscible fluids with phase behavior controlled by constant K-values with $K = \{2.5, 0.3\}$. Initially, the 1D domain is fully saturated by the non-wetting phase and we inject a wetting phase at the left boundary. We ran the simulation with a small timestep for 1000 days. Next, we restart by enlarging the timesteps. Fig. 10 shows the fracture test results comparing different nonlinear solvers for binary kernels. We set the maximum number of nonlinear iterations to 50 for all test cases.

It is clear that the trust region solver performs better (provides fewer nonlinear iterations) for both immiscible and miscible kernels. Here global and local chops only provide a convergence strategy for timesteps corresponding to $\Delta t = 500$ days and fail to converge for larger timesteps.

Furthermore, we run front propagation in a single fracture test case for ternary and quaternary kernels. We inject at 135 bar using bottom hole pressure control and initial reservoir pressure of 95 atm. For the ternary kernel, we inject the gas stream of $\{CO_2, NC_4, C_{10}\}$ with the composition of $z_{inj} = \{0.98, 0.01, 0.01\}$ into the reservoir initialized with $z_{ini} = \{0.1, 0.2, 0.7\}$. Similarly, in 4-component system we inject gas stream of $\{CO_2, NC_4, C_{10}, C_2\}$ with the composition of $z_{inj} = \{0.98, 0.001, 0.001, 0.01\}$ into the reservoir initialized with $z_{ini} = \{0.1, 0.2, 0.2, 0.5\}$. Fig. 11 illustrates the performance of different nonlinear solvers for single fracture test case. In both the ternary and four-component test cases, the TR solver converges with fewer newton iterations.

6.3. Full compositional simulation

In this test case, we run the simulation study for multiple timesteps until reaching the final time. We start the simulation with a small timestep ($\Delta t = 1$ day), if nonlinear iteration converges, we double the timestep for the next timestep and if it does not converge we cut the timestep to half. The maximum Newton iteration is again $N = 50$. In this test case we compare four different nonlinear solvers: *STD*, *TR*, *Global* and *local* chop with damping factor 0.1.

6.3.1. 1D homogeneous model

This is the ternary 1D full simulation test case. The domain is discretized into 1000 grid cells with a homogeneous permeability and porosity of $K = 10$ mD and $\phi = 0.3$. Components are $\{CO_2, NC_4, C_{10}\}$. We inject a gas mixture of composition (0.98, 0.01, 0.01) into the (0.1, 0.25, 0.65) under BHP control (constant pressure 405 bar) at the left boundary and produce it under constant pressure 395 bar at the right boundary.

Fig. 12 illustrates the cumulative number of nonlinear iterations for different final times and compares different nonlinear solver's performances. It is clear that the TR solver has superior performance with respect to other solvers. Fig. 13 compares the composition front for different nonlinear solvers. As we can see, the STD solver can capture the shock accurately. On the other hand, the TR solver smears out the shock since it is capable of converging for a more aggressive timestep and consequently larger time-truncation error is present in the solution.

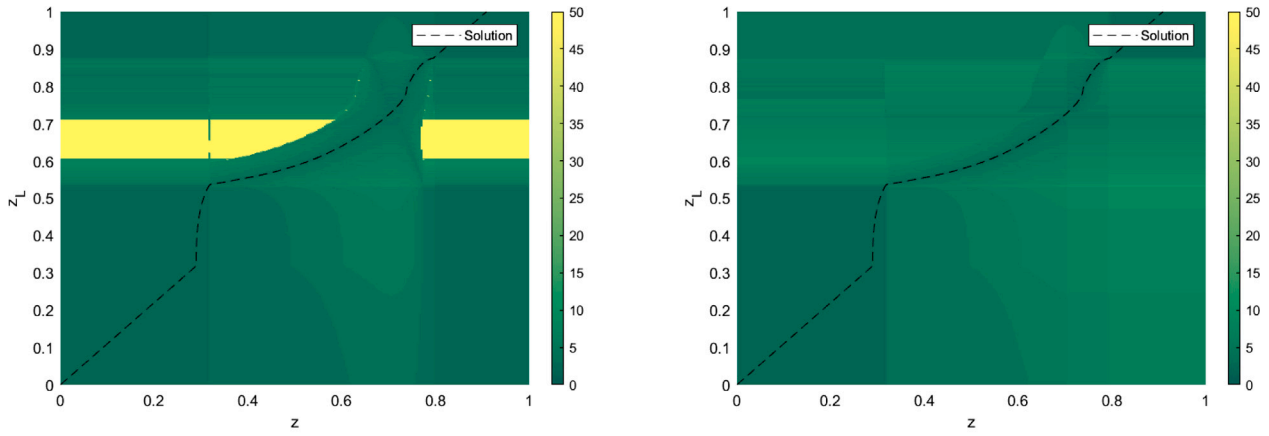
The smearing effect of time truncation is most noticeable at the shock. Because the other solvers cannot converge for larger time steps, they reduce timesteps, resulting in less truncation error and better composition front evolution. While time truncation error analysis is beyond the scope of this paper, having a robust nonlinear solver allows us to choose timestep from practical considerations rather than the necessity to perform a simulation with smaller timesteps.

6.4. 3D heterogenous model

In this test case, we compared the performance of nonlinear solvers for a heterogeneous reservoir. We used a channelized Egg model described in Jansen et al. (2014). The dimension of the model is $60 \times 60 \times 7 = 25200$ grid cells of which 18553 cells are active, with gridsize of 8 m × 8 m × 4 m. The porosity is constant ($\phi = 0.2$). Fig. 14 illustrates the permeability and well locations. The injection well is located in the left upper part and the production well is at the right bottom of the domain. The initial pressure of the reservoir is 400 bar. We inject a gas mixture of composition (0.98, 0.01, 0.01) into the (0.1, 0.25, 0.65) at constant pressure 405 bar and produce at 395 bar. We run the simulation for different nonlinear solvers several times.

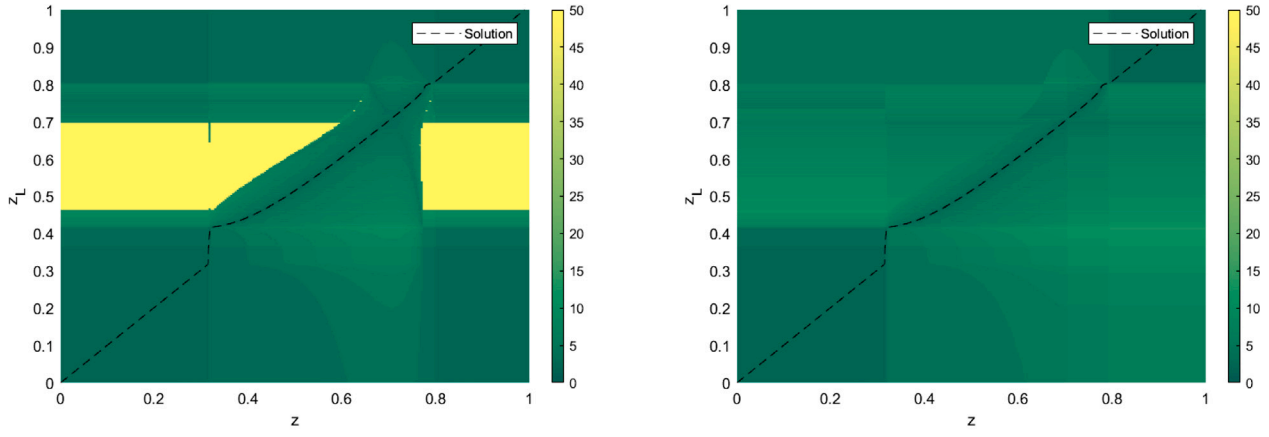
Fig. 15 summarizes the nonlinear iterations for different nonlinear solvers. The TR solver takes a lower number of nonlinear iterations to converge compared to the other solvers. In this test case, the STD solver takes the highest number of Newton iterations since it relies on a pure Newton's update. Local chop performs slightly better than global chop in this setting but still worse than TR.

We take the solution of STD as a reference solution for the time-truncation error analysis of the solution for the single top layer of the model. From Fig. 16 we can see that the error is mainly localized around the trailing and leading shocks. Similar to the 1D homogeneous model test case, the time truncation error affects the accuracy of the solution since the TR solver can converge for more aggressive timesteps. You can see that the difference in the 2D heterogeneous solution is less pronounced than the one detected in homogeneous 1D.



(a) Convergence map for pure newton (b) Convergence map for TR algorithm

Fig. 7. Convergence map for compositional binary problem with C=$\frac{V_t dt}{dx}$ = 10.



(a) Convergence map for pure newton (b) Convergence map for TR algorithm

Fig. 8. Convergence map for compositional binary problem with C=$\frac{V_t dt}{dx}$= 100.

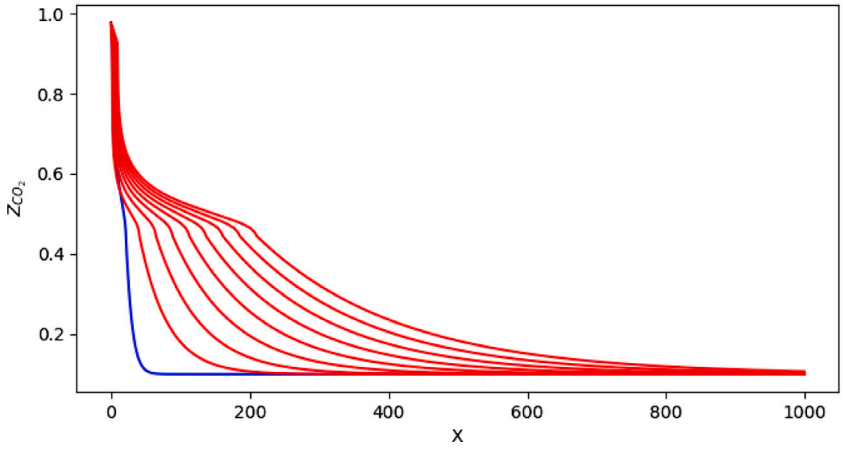


Fig. 9. CO_2 front propagation for different timesteps. The blue curve represents the resolved displacement solution, and the red curves represent the progressive control timestep solutions.

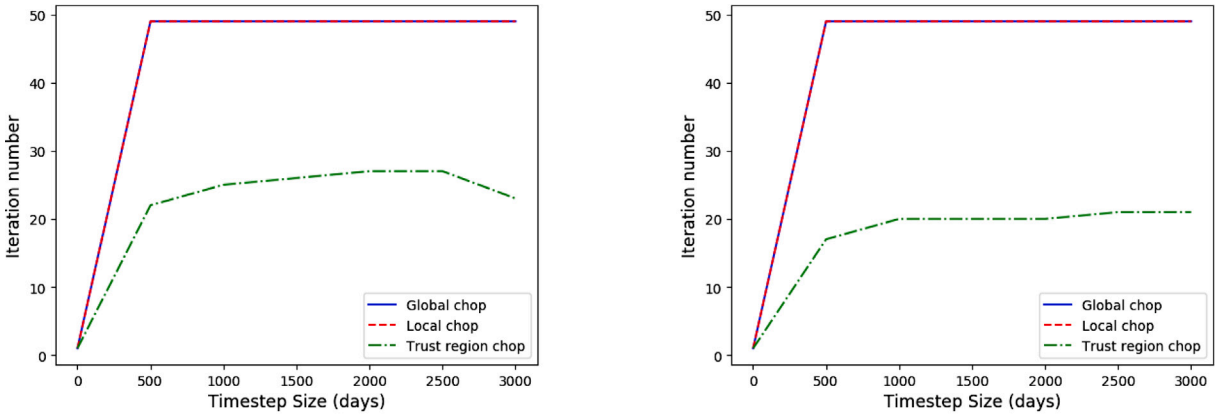


Fig. 10. Comparison of different nonlinear solvers for binary kernel.

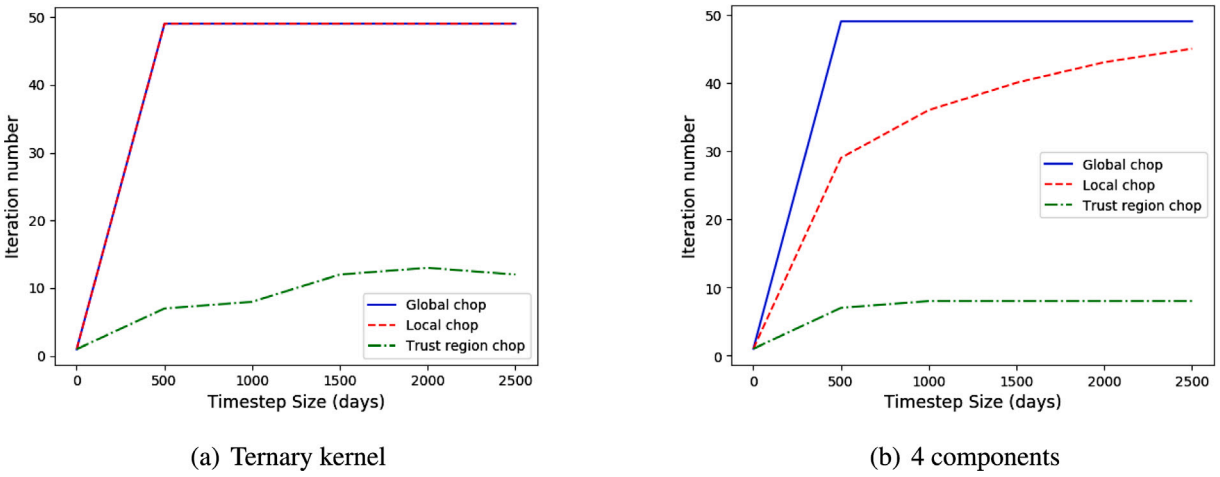


Fig. 11. Comparison of different nonlinear solvers for ternary and quaternary kernel.

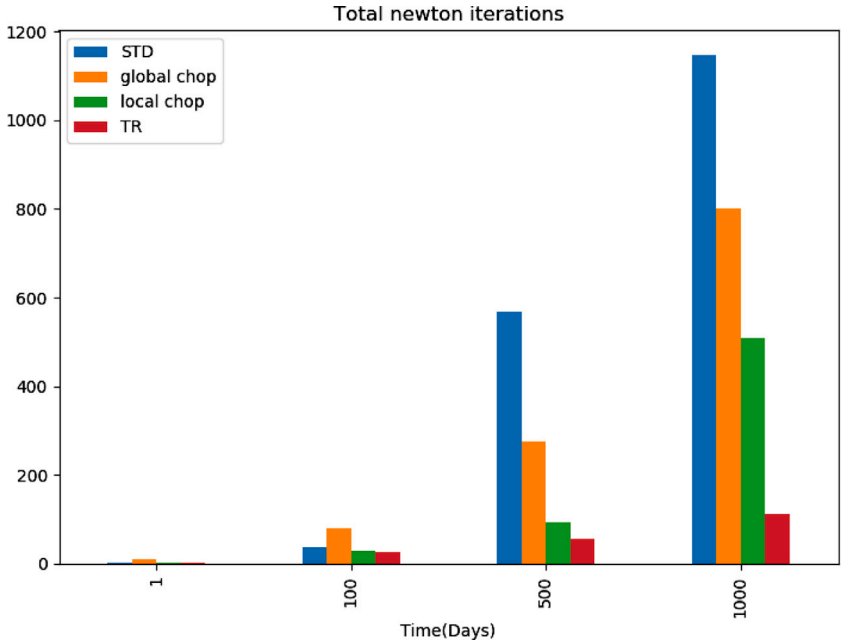


Fig. 12. Comparison of cumulative Newton iterations for different nonlinear solver for ternary kernel.

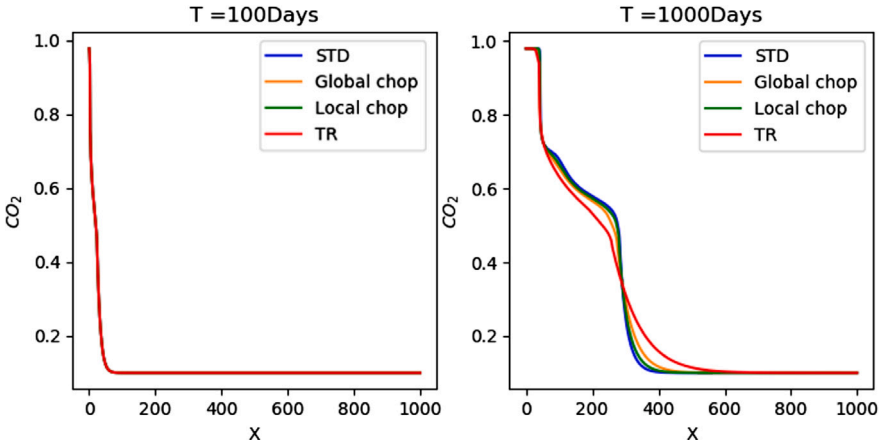


Fig. 13. Comparison of solution for different nonlinear solvers for ternary kernel.

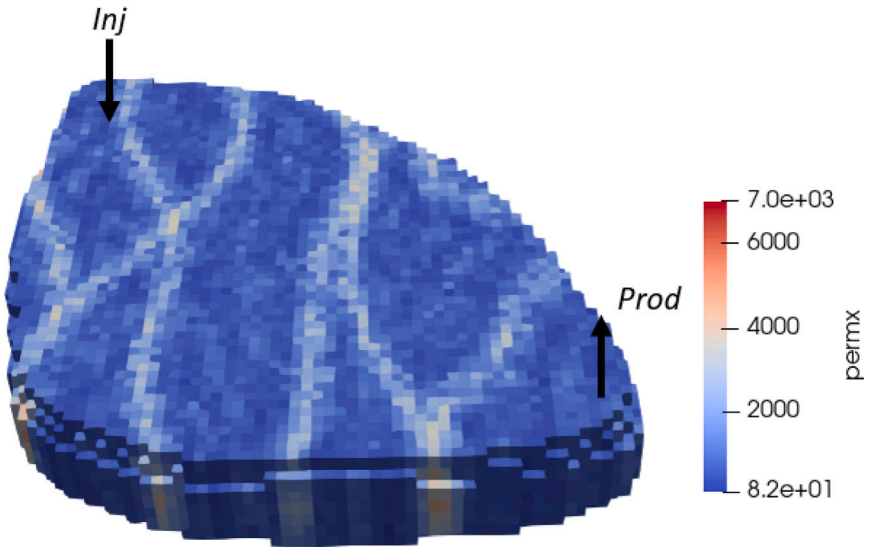


Fig. 14. Permeability K_x map of egg model.

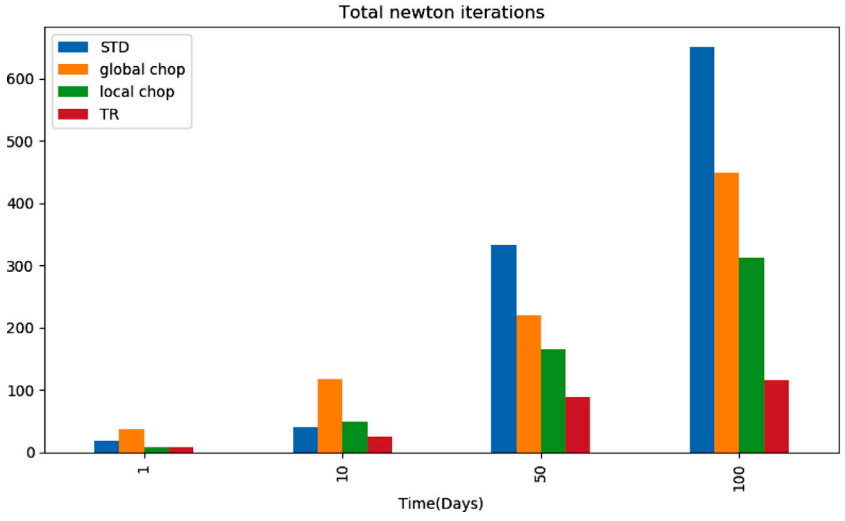


Fig. 15. Comparison of cumulative Newton iteration for different nonlinear solver for egg-model at different time.

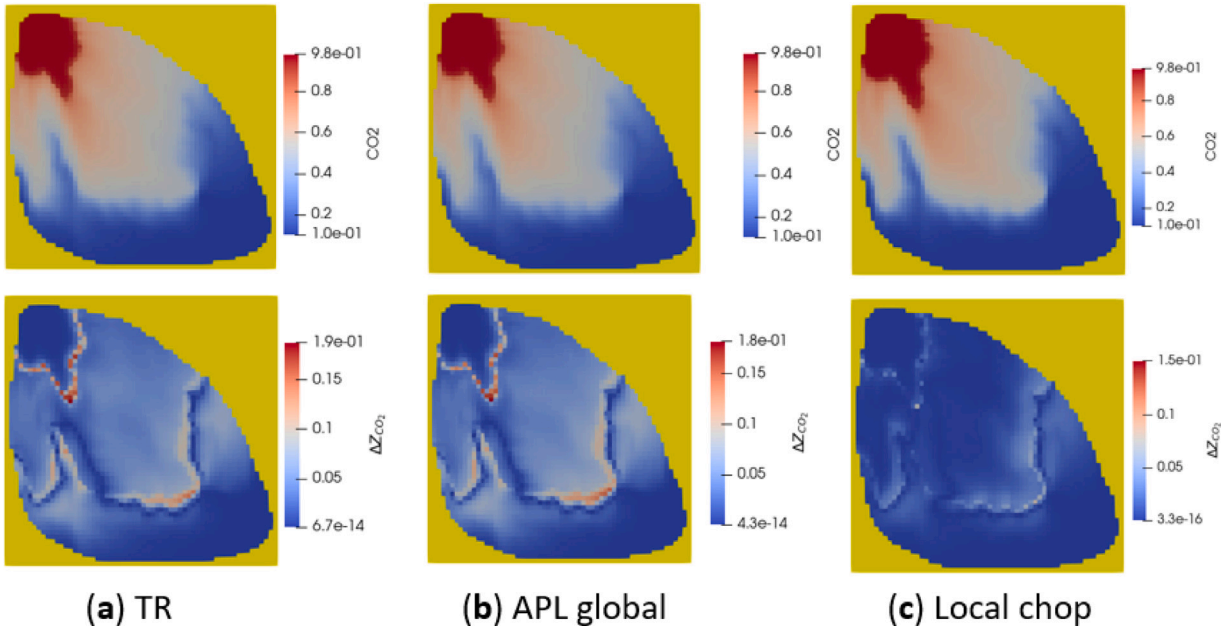


Fig. 16. Comparison of CO_2 saturation maps at the time of $T = 100$ days (1st row). The second row shows the delta of gas saturation between each model and the STD case.

7. Conclusion

For simulation of CO_2 utilization and storage (CCUS) in subsurface reservoirs with complicated heterogeneous structures, a model that includes multiphase compositional flow and transport is needed. The discretized governing equations are highly nonlinear, and Newton's technique is frequently used to solve them. Newton's solution technique does not ensure convergence and is extremely dependent on the timestep choice.

In this work, we investigate the nature of nonlinearities in CCUS simulations and suggest solutions to a general compositional problem. We present an advanced nonlinear solver based on a trust-region technique aimed to solve multiphase multi-component transport problems. The trust region solver is based on the analysis of multi-dimensional tables connected to parameterized highly nonlinear convection operators. These operators are associated with the governing equations and are built for a newly introduced Operator-Based Linearization approach. The inflection line and kinks in the parameter space determine the delineation of the trust-region zones.

According to our nonlinear study of convective operators for binary and ternary systems, each component has its inflection line within the two-phase region. In addition, kink lines appear when phase boundaries are crossed. These boundaries could change in the parameter space of the problems based on the direction of the Newton trajectory. We track the nonlinear trajectory and segment the parameter space of the problem into a set of trust regions where the hyperbolic operators keep their second-order behavior (i.e., they remain either convex or concave). We approximate these trust regions in the solution process by detecting the boundary of convex regions via analysis of the directional derivative. By drawing multiple trial trajectories on binary and ternary diagrams we observe that our algorithm can detect these boundaries correctly. Moreover, it is less computationally expensive since we do not compute the entire hessian in our technique and instead compute the directional derivative while tracking the nonlinear update.

After detecting all the boundaries along the nonlinear trajectory, the proposed nonlinear solver locally constrains the update of the overall compositions across the boundaries of these regions. We tested our nonlinear solver for several reservoir models starting from the single cell to a fully 3D heterogeneous model. Our numerical results show that the trust-region solver avoids overshoots in the nonlinear update which lead to superior convergence in comparison to conventional nonlinear solvers.

A promising future research direction could be to reduce the computational overhead related to locating the trust-region boundaries and carrying out the chopping. While we showed that directional derivative is cheaper than full Hessian assembly, there is still additional overhead since detection of special points (kinks and inflection) are happening on the fly during simulation. The major cause of the overhead is due to the tracking the Newton trajectory in nonlinear operator space.

One possibility is that once the special points have been identified, we save the value to avoid recalculation for the next iteration. Another possible future direction would be to combine the TR solver with another type of solver. Based on our single cell analysis, we can see that once the solution is in a single-phase region, conventional Newton solvers work, so we can switch between different solvers adaptively if we observe that one is struggling.

CRediT authorship contribution statement

Kiarash Mansour Pour: Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Denis Voskov:** Conceptualization, Methodology, Supervision, Writing – review & editing. **David Bruhn:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

We would like to acknowledge the entire DARTS team for the constructive suggestions.

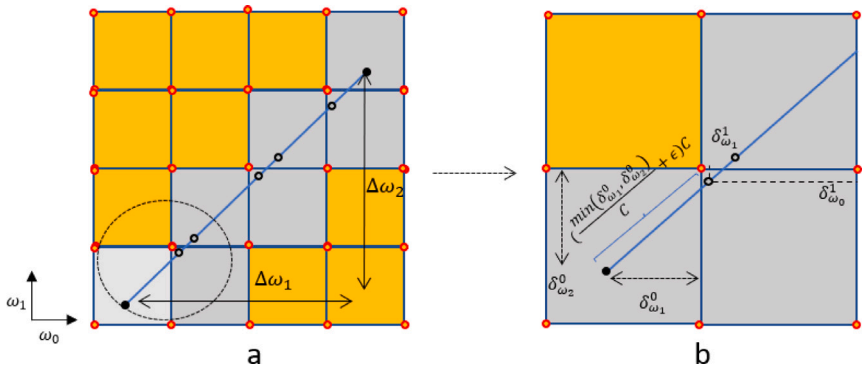


Fig. 17. (a) Newton trajectory passing several OBL cells (b) Zoomed-in view of detecting the point inside the next cell of parameter space.

Appendix A. Algorithm to track Newton trajectory in an arbitrary dimension of OBL space

Newton's trajectory passes several cells or hyper cells in a higher dimension of parameter space. In order to detect those cells or hyper cells, we implement the iterative algorithm 1. The algorithm is incrementing to the next cell iteratively by finding the minimum distance between the initial point with respect to interfaces and moving on in a gradient direction to find the next point until reaching the last hypercubes. Here's the explanation of the following algorithm steps:

1. *COMPUTESIGMA*: This function calculates σ which is the minimum distance between the initial points and all the interfaces of the OBL cube/ hypercube.
2. *NEXTPOINT*: This function increments to the next hypercubes by moving along the Newton trajectory gradient.
3. *NEWBOX*: This function detects the new vertices of the new cubes/hypercubes.

These steps are repeated until reaching the last points of the OBL and all the interfaces detected. Fig. 17 illustrates in 2-d space the Newton trajectory tracking. The algorithm for tracking is independent of the degree of freedom of the system and is able to track Newton trajectories for an arbitrary number of dimensions.

Algorithm 1 Iterative tracking

procedure ITERATIVE TRACKING($initpoint, lastbox$)
  Input:
  $initpoint \leftarrow (p_i, z_{1i}, z_{2i}...)$
  $endpoint \leftarrow (p_e, z_{1e}, z_{2e}...)$
  $C \leftarrow initpoint - endpoint$
  while box != lastbox **do**
    $Sigma \leftarrow ComputeSigma(initpoint, box, C)$
    $initpoint \leftarrow nextPoint(initPoint, C, Sigma)$
    $box \leftarrow newbox(initpoint, z_{vec}, p_{vec})$ ▷ p_{vec}, z_{vec} are uniformly mesh in parameter space
  end while
  return *box*
end procedure

function COMPUTESIGMA($initpoint, box, C$)
  for $i \leftarrow 1, NC$ **do** ▷ NC = Number of components
    $\Lambda_i \leftarrow max(\frac{rightboundary - initpoint(i)}{C(i)}, \frac{leftboundary - initpoint(i)}{C(i)})$
  end for
  $\sigma \leftarrow min(\Lambda)$
  return σ
end function

function NEXTPOINT($initpoint, Sigma, C$)
  for $i \leftarrow 1, NC$ **do** ▷ NC = Number of components
    $nextPoint(i) \leftarrow initpoint(i) + (\sigma + \epsilon)C(i)$ ▷ $+\epsilon$ to make sure passing the interface
  end for
  return *nextPoint*
end function

function NEWBOX($initpoint, z_{vec}, p_{vec}$)
  for $i \leftarrow 1, NC$ **do** ▷ NC = Number of components
    if $i == 1$ **then**
      $newbox(i) \leftarrow [\frac{initpoint(i) - p_{vec(1)}}{p_{vec(2)} - p_{vec(1)}}]$ ▷ [] Rounding to upper integer
    else
      $newbox(i) \leftarrow [\frac{initpoint(i) - z_{vec(1)}}{z_{vec(2)} - z_{vec(1)}}]$
    end if
  end for ▷ We find one of the new vertices of the new box
end function

Table 1
Immiscible fluid properties.

Parameter	Oil	Gas	Desctiption
n	2	2	Corey exponent
μ_{ref}	1.5cp	1cp	Viscosity
ρ_{ref}	1000 kg/m^3	800 kg/m^3	Density

Table 2
Phase properties.

Components	CO_2	C_1	H_2	C_2
K-values	2	0.4	0.1	0.02

Appendix B. Parameters of numerical tests

Immiscible properties
See Table 1.

Miscible fluid properties
See Tables 2 and 3.

Appendix C. Sensitivity analysis to the OBL resolution

In this section, we investigate the effect of the OBL resolution on the nonlinear solver performance. Table 4 summarizes the result of

Table 3
Thermodynamic properties.

Parameter	Oil	Gas	Description
n	2	2	Corey exponent
μ_{ref}	1.5cp	0.05cp	Viscosity
ρ_{ref}	20 kg/m^3	10 kg/m^3	Density

Table 4
Coarsening of OBL resolution.

Resolution	Newton iteration	Inflection points
4 × 4	5	1e−8
8 × 8	6	0.285
16 × 16	8	0.40
32 × 32	10	0.44
64 × 64	12	0.476
120 × 120	14	0.487

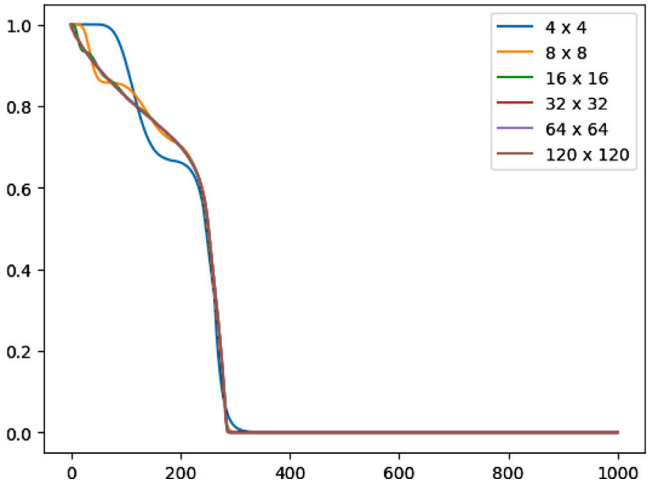


Fig. 18. Solution of front propagation for different OBL resolutions.

coarsening of OBL resolution of the fracture test for the binary kernel on the immiscible test case for the single control timestep $\Delta t = 100$ days. We observe that by coarsening the resolution, the number of Newton iterations decreases. In general, decreasing the resolution of OBL relaxes the nonlinearity of the problem and fewer Newton iterations. Fig. 18 illustrates the solution for different OBL resolutions. Notice that decreasing the resolution of the OBL significantly can degrade the solution. The rigorous error analysis has been carried out on the original paper (Voskov, 2017).

References

Acs, G., Doleschall, S., Farkas, E., 1985. General purpose compositional model. Soc. Petrol. Eng. J. 25 (04), 543–553, https://doi.org/10.2118/10515-PA.

Aziz, K., Settari, A., 1979. Petroleum Reservoir Simulation. Elsevier, London.

Coats, K.H., 1980. An equation of state compositional model. Soc. Petrol. Eng. J. 20 (05), 363–376. http://dx.doi.org/10.2118/8284-PA.

Collins, D., Nghiem, L., Li, Y.-K., Grabenstetter, J., 1992. Efficient approach to adaptive-implicit compositional simulation with an equation of state. SPE Res. Eng. (Soc. Petrol. Eng.) 7 (2), 259–264. http://dx.doi.org/10.2118/15133-PA.

Deuflhard, P., 2004. Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms. Springer, Berlin.

Garipov, T., Tomin, P., Rin, R., Voskov, D., Tchelepi, H., 2018. Unified thermo-compositional-mechanical framework for reservoir simulation. Comput. Geosci. http://dx.doi.org/10.1007/s10596-018-9737-5.

Geoquest, 2011. ECLIPSE Technical Description. Technical report, Schlumberger.

Gilgen, H., 2006. Univariate Time Series in Geosciences: Theory and Examples. Springer Berlin Heidelberg, Berlin.

Jansen, J., Fonseca, R., Kahrobaei, S., Siraj, M., Essen, van, G., Hof, Van den, P., 2014. The egg model - a geological ensemble for reservoir simulation. Geosci. Data J. 1 (2), 192–195. http://dx.doi.org/10.1002/gdj3.21.

Jenny, P., Tchelepi, H.A., Lee, S.H., 2009. Unconditionally convergent nonlinear solver for hyperbolic conservation laws with S-shaped flux functions. J. Comput. Phys. 228 (20), 7497–7512. http://dx.doi.org/10.1016/j.jcp.2009.06.032.

Jiang, J., Pan, H., 2022. Dissipation-based nonlinear solver for fully implicit compositional simulation. SPE J. http://dx.doi.org/10.2118/209233-PA.

Kala, K., Voskov, D., 2020. Element balance formulation in reactive compositional flow and transport with parameterization technique. Comput. Geosci. 24 (2), http://dx.doi.org/10.1007/s10596-019-9828-y.

Khait, M., 2019. Delft Advanced Research Terra Simulator: General Purpose Reservoir Simulator with Operator-Based Linearization (Ph.D. thesis). TU Delft.

Khait, M., Voskov, D.V., 2017. Operator-based linearization for general purpose reservoir simulation. J. Pet. Sci. Eng. 157, 990–998. http://dx.doi.org/10.1016/j.petrol.2017.08.009.

Khait, M., Voskov, D.V., 2018. Adaptive parameterization for solving of thermal/compositional nonlinear flow and transport with Buoyancy. SPEJ http://dx.doi.org/10.2118/182685-PA.

Khebzegga, O., Iranshahr, A., Tchelepi, H., 2021. A nonlinear solver with phase boundary detection for compositional reservoir simulation. Transp. Porous Media 137 (3), 707–737. http://dx.doi.org/10.1007/s11242-021-01584-4.

Li, B., Tchelepi, H.A., 2014. Unconditionally convergent nonlinear solver for multiphase flow in porous media under viscous force, buoyancy, and capillarity. 59, pp. 404–411. http://dx.doi.org/10.1016/j.egypro.2014.10.395,

Li, B., Tchelepi, H.A., 2015. Nonlinear analysis of multiphase transport in porous media in the presence of viscous, buoyancy, and capillary forces. J. Comput. Phys. 297, 104–131. http://dx.doi.org/10.1016/j.jcp.2015.04.057.

Lyu, X., Khait, M., Voskov, D., 2021a. Operator-based linearization approach for modeling of multiphase flow with buoyancy and capillarity. SPE J. 26 (04), 1858–1875. http://dx.doi.org/10.2118/205378-PA.

Lyu, X., Voskov, D., Rossen, W., 2021b. Numerical investigations of foam-assisted CO2 storage in saline aquifers. Int. J. Greenh. Gas Control 108, http://dx.doi.org/10.1016/j.ijggc.2021.103314.

Michelsen, M.L., 1982. The isothermal flash problem. Part II. Phase-split calculation. Fluid Phase Equilib. 9 (1), 21–40. http://dx.doi.org/10.1016/0378-3812(82)85002-4.

Møyner, O., 2017. Nonlinear solver for three-phase transport problems based on approximate trust regions. Computational Geosciences 21 (5-6), 999–1021. http://dx.doi.org/10.1007/s10596-017-9660-1.

Ortega, J., Rheinboldt, W., 1970. Iterative Solution of Nonlinear Equations in Several Variables. In: Computer Science and Applied Mathematics. A Series of Monographs and Textobooks, Academic Press, New York.

Pruess, K., Fensterle, S., Moridis, G., Oldenburg, C., Wu, Y., 1997. General purpose reservoir simulators: the TOUGH2 family. J. Comput. Phys. 62 (2), 265–281.

Voskov, D.V., 2017. Operator-based linearization approach for modeling of multiphase multi-component flow in porous media. J. Comput. Phys. 337, 275–288. http://dx.doi.org/10.2118/205378-PA.

Voskov, D., Tchelepi, H., 2008. Compositional space parametrization for miscible displacement simulation. Transp. Porous Media 75 (1), 111–128. http://dx.doi.org/10.1007/s11242-008-9212-1.

Voskov, Tchelepi, 2011. Compositional nonlinear solver based On Trust Regions of the flux function along key Tie-lines. Soc. Petrol. Eng. http://dx.doi.org/10.2118/141743-MS.

Voskov, D., Tchelepi, H., 2012. Comparison of nonlinear formulations for two-phase multi-component EoS based simulation. J. Pet. Sci. Eng. 82–83, 101–111. http://dx.doi.org/10.1016/j.petrol.2011.10.012.

Wang, X., Tchelepi, H., 2013. Trust-region Newton solver for multiphase flow and transport in heterogeneous porous media. In: Society of Petroleum Engineers - SPE Reservoir Simulation Symposium 2013, Vol. 1. pp. 358–377. http://dx.doi.org/10.2118/163600-MS.

Wang, Y., Voskov, D., Khait, M., Bruhn, D., 2020. An efficient numerical simulator for geothermal simulation: A benchmark study. Appl. Energy 264, http://dx.doi.org/10.1016/j.apenergy.2020.114693.

Younis, R., 2011. Modern advances in software and solution algorithms for reservoir simulation (Ph.D. thesis). Stanford University.