Adaptive nonlinear solver for a discrete fracture model in operator-based linearization framework

Pour, K. Mansour; Voskov, D.

DOI
10.3997/2214-4609.202035094

Publication date
2020

Document Version
Final published version

Published in
ECMOR 2020

Adaptive Nonlinear Solver for a Discrete Fracture Model in Operator-Based Linearization Framework

K. Mansour Pour[1*], D. Voskov[1]

[1] Delft University of Technology

Summary

Simulation of compositional problems in hydrocarbon reservoirs with complex heterogeneous structure requires adopting stable numerical methods that rely on an implicit treatment of the flux term in the conservation equation. The discrete approximation of convection term in governing equations is highly nonlinear due to the complex properties complemented with a multiphase flash solution. Consequently, robust and efficient techniques are needed to solve the resulting nonlinear system of algebraic equations. The solution of the compositional problem often requires the propagation of the displacement front to multiple control volumes at simulation timestep. Coping with this issue is particularly challenging in complex subsurface formations such as fractured reservoirs. In this study, we present a robust nonlinear solver based on a generalization of the trust-region technique to 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 segment the parameter-space of the nonlinear problem into a set of trust regions where the convection operators maintain the second-order behaviour (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 constraints the updating of the overall compositions across the boundaries of convex regions. Besides, we enhance the performance of the nonlinear solver by exploring diverse preconditioning strategies for compositional problems. The proposed nonlinear solution strategies have been validated for both miscible and immiscible gas injection problems of practical interest.

Introduction

Reservoir simulation is based on the solution of the discretized governing equations describing the mass and energy transfer in the reservoir. The main source of nonlinearity is related to an implicit approximation of flux term in conservation equations which is needed for the robustness (unconditional stability) of the reservoir simulation process. Therefore, efficient modeling of multiphase flow in reservoirs with complex heterogeneous structures requires robust nonlinear solvers.

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 (Gabor et al., 1985; Collins et al., 1992) which uses phase-dependent or mass-dependent variables respectively. Usually, linearization is made using Newton's method, which demands the assembly of the Jacobian and the residual for the combined system of equations. Due to the intrinsic nonlinear nature of the equations, Newton's method is not guaranteed to converge, and it is also known to be sensitive to the initial guess (Ortega and Rheinboldt, 1970; Deuflhard, 2004).

Once the solution of the linearized system is found, the nonlinear unknowns are updated and nonlinear iterations are repeated until convergence. In reservoir simulation practice, heuristic techniques are used to select the timesteps (Aziz and Settari, 1979). The use of such heuristics often leads to timestep sizes that are either too conservative (i.e., small) or too large which often leads to wasted computations (Younis, 2011). The limitation of timestep selection can be overcome by applying advanced nonlinear strategy.

There are several advanced nonlinear solvers described in the literature for the natural 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 the continuous integration along the nonlinear trajectory in parameter space. Another approach is the flux-based trust region method, proposed by Jenny et al. (2009) for two-phase immiscible flow with the S-shape fractional flow curves. This nonlinear solver is based on an inflection-point correction. Later, Wang and Tchelepi (2013) extended the flux-based trust region for two-phase immiscible flow and transport where buoyancy, capillary, and viscous forces are present.

To enhance further the performance of the advanced nonlinear solver, Wang (2012) proposed the preconditioning strategy to overcome the slow convergence rate for viscous-dominated flow with initial conditions started at the residual saturation. She demonstrated that the convergence difficulties are due to the low wave propagation speed at the shock position where the injected fluid is about to invade a single-phase cell. To obtain a better performance, Wang (2012) proposed to use the inflection point of the flux function for the nonlinear preconditioning. However, this analysis was performed for simplest physical formulation and was never extended and tested for more complex physics (including compositional) or different formulation.

Even though different advanced nonlinear solvers were developed for the natural formulation, there is a lack of advanced nonlinear strategies for the molar formulation. A version of trust-region correction has been developed for molar formulation (Voskov and Tchelepi, 2011) but was not robust enough in comparison with techniques proposed for the natural formulation. 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 simulation model was approximated using abstract algebraic operators. In the OBL approach, the parameterization is performed dependent on the conventional molar unknowns (pressure and overall composition).

In this work, we present an advanced nonlinear solver based on a generalization of the trust-region technique for compositional multiphase transport in the OBL framework. First, we investigate the nonlinearity and detect boundaries of trust-region for the hyperbolic operator by assembling the directional approximation of hessian matrix. Next, we design the nonlinear solver in which we track the nonlinear

trajectory for binary and ternary kernel and approximate these trust regions in the solution process via directional analysis of derivative. Moreover, we enhance our advanced nonlinear solver performance further by introducing preconditioning for binary compositional kernel which can be directly extended for compositional problems with more components.

Modeling approach

In this section we describe an operator form of governing equations used in our in-house open source Delft Advanced Research Terra Simulator (Khait, 2019).

Governing equations

In this section, we describe the governing equations and nonlinear formulation for a general-purpose compositional simulation. The transport equations for an isothermal multiphase compositional problem with n_p phases and n_c components can be written as:

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

Here, we introduce all variables in the equations as functions of spatial coordinate $\boldsymbol{\xi}$ and/or physical state $\boldsymbol{\omega}$:

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

Spatial discretization

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

$$V\left((\phi\sum_j x_{cj}^l \rho_j s_j)^{n+1} - (\phi\sum_j x_{cj}\rho_j s_j)^n\right) - \Delta t\sum_j\left(\sum_{l\in L} x_{cj}^l \rho_j^l T_j^l \Delta\Psi^l\right) + \Delta t\sum_j \rho_p x_{cj}\overline{q_j} = 0, \tag{2}$$

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 another 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 a phase transmissibility. These assumptions are not required by the proposed approaches, but help to simplify the further description.

Sources of nonlinearity

The main source of nonlinearity is related to the use of Fully Implicit Method (FIM) for time approximation of the governing equations which requires the flux term in Equation 2 to be defined based on the value of the nonlinear unknowns at a new timestep $(n+1)$. The closure assumption of instantaneous

thermodynamic equilibrium further increases the nonlinearity. We used overall molar formulation suggested by (Collins et al., 1992). In this formulation, the following system must be solved at any grid block contained a multiphase (n_p) multi component n_c mixture:

$$\begin{aligned} F_c &= z_c - \Sigma v_j x_{cj} = 0, && (3)\\ F_{c+n_c} &= f_{c1}(p,T,x1) - f_{cj}(p,T,x_j) = 0, && (4)\\ F_{j+n_c*n_p} &= \Sigma(x_{c1} - x_{cj}) = 0, && (5)\\ F_{n_p+n_c*n_p} &= \Sigma v_j - 1 = 0. && (6) \end{aligned}$$

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).

Operator form

We can rewrite Equation 2 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

$$r_c(\boldsymbol{\omega}) = V(\boldsymbol{\xi})\phi_0(\boldsymbol{\xi})(\alpha_c(\boldsymbol{\omega}) - \alpha_c(\boldsymbol{\omega}^n)) - \sum_l \beta_c^l(\boldsymbol{\omega})\Lambda(\boldsymbol{\omega})\Delta t T^{ab}(\boldsymbol{\xi})(p^b - p^a) + \theta_c(\boldsymbol{\xi},\boldsymbol{\omega},\mathbf{u}) = 0, \quad (7)$$

where operators are defined as

$$\begin{aligned} \alpha_c(\boldsymbol{\omega}) &= (1 + c_r(p - p_{ref}))\sum_{j=1}^{n_p} x_{cj}\rho_j s_j, && (8)\\ \beta_c(\boldsymbol{\omega}) &= \frac{\sum_j x_{cj}\frac{k_{rj}}{\mu_j}\rho_j}{\Lambda} = \sum_j x_{cj} f_j \rho_j, && (9)\\ \Lambda(\boldsymbol{\omega}) &= \sum_j \frac{k_{rj}}{\mu_j}, && (10)\\ \theta_c(\boldsymbol{\xi},\boldsymbol{\omega},u) &= \Delta t \sum_j \rho_j x_{cj} q_j(\boldsymbol{\xi},\boldsymbol{\omega},\mathbf{u}). && (11) \end{aligned}$$

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), f_j are fractional flow function. The variables $\boldsymbol{\omega}$ and $\boldsymbol{\omega}^n$ are nonlinear unknowns at the current and the previous timestep respectively, and $\mathbf{u}$ is a vector of well control variables. The operator $\theta_c(\boldsymbol{\xi},\boldsymbol{\omega},\mathbf{u})$ 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 spatial distributed properties. For simplicity, in this study, we will ignore the well source/sink and only apply fixed boundary conditions.

Linearization and solution

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

Standard linearization approach

To solve nonlinear equations 7, 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:

$$\mathbf{J}(\boldsymbol{\omega}^k)(\boldsymbol{\omega}^{k+1} - \boldsymbol{\omega}^k) = -\mathbf{r}(\boldsymbol{\omega}^k), \tag{12}$$

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 numerical approximation of the analytic relations in Equations 8 - 11. 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 a 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 (Garipov et al., 2018) differentiation of each property with respect to nonlinear unknowns.

Operator-Based Linearization

Operator-Based Linearization is a newly proposed strategy for the linearization of the reservoir simulation problem described by Eq. 7. As can be seen from the structure of each operator in Equations 8- 11, 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 a multi-linear interpolation in parameter space for the continuous representation of physics-based operators and discrete representation of their derivatives. The number of points in the interpolation controls the accuracy of approximation of the nonlinear physics, governs the process. This is similar to the accuracy of the approximation in space and time being controlled by the grid and timestep size. The details of OBL approach, test results and convergence analysis can be found in (Khait and Voskov, 2017) and (Khait and Voskov, 2018).

Adaptive Nonlinear Solver for OBL framework

For simplicity, we assume in the following derivations that the system is incompressible which limits the analysis to the convection operators β_c only. We segment the parameter-space of the nonlinear problem into a set of regions where β_c maintain their second order behavior (i.e., they remain either convex or concave). The proposed nonlinear solver locally constraints the updating of hyperbolic unknowns (the overall compositions) across the boundaries of these regions similar to inflection point correction proposed by Jenny et al. (2009). Essentially, it is a cell-wise chopping strategy guided by trust regions of the operators. Our nonlinear solver ensures that the successive iteration update cannot cross any trust-region boundary.

The delineation of the trust regions is dictated by kinks and inflection points in parameter space. 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:

$$\mathbf{H}(\omega) = \mathbf{J}\left(\nabla[\beta_c(\omega)]\right) = \begin{bmatrix} \frac{\partial^2 \beta_1}{\partial \omega_1^2} & \frac{\partial^2 \beta_1}{\partial \omega_1 \partial \omega_2} & \cdots & \frac{\partial^2 \beta_1}{\partial \omega_1 \partial \omega_c} \\ \frac{\partial^2 \beta_1}{\partial \omega_2 \partial \omega_1} & \frac{\partial^2 \beta_1}{\partial \omega_2^2} & \cdots & \frac{\partial^2 \beta_1}{\partial \omega_2 \omega_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \beta_1}{\partial \omega_c \partial \omega_1} & \frac{\partial^2 \beta_1}{\partial \omega_c \partial \omega_2} & \cdots & \frac{\partial^2 \beta_1}{\partial \omega_c^2} \end{bmatrix}, \tag{13}$$

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

For ternary system we uniformly discretize the parameter space and compute the Hessian numerically as follow

$$\left(\frac{\partial^2 \beta}{\partial z_1^2}\right)_{i,j} = \frac{(\frac{\partial \beta}{\partial z_1})_{i+1,j} - (\frac{\partial \beta}{\partial z_1})_{i,j}}{\Delta z}, \tag{14}$$

$$\left(\frac{\partial^2 \beta}{\partial z_2^2}\right)_{i,j} = \frac{(\frac{\partial \beta}{\partial z_2})_{i,j+1} - (\frac{\partial \beta}{\partial z_2})_{i,j}}{\Delta z}, \tag{15}$$

$$\left(\frac{\partial^2 \beta}{\partial z_2 \partial z_1}\right)_{i,j} = \left(\frac{\partial^2 \beta}{\partial z_1 \partial z_2}\right)_{i,j} = \left(\frac{(\frac{\partial \beta}{\partial z_1})_{i,j+1} - (\frac{\partial \beta}{\partial z_1})_{i,j}}{\Delta z}\right). \tag{16}$$

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

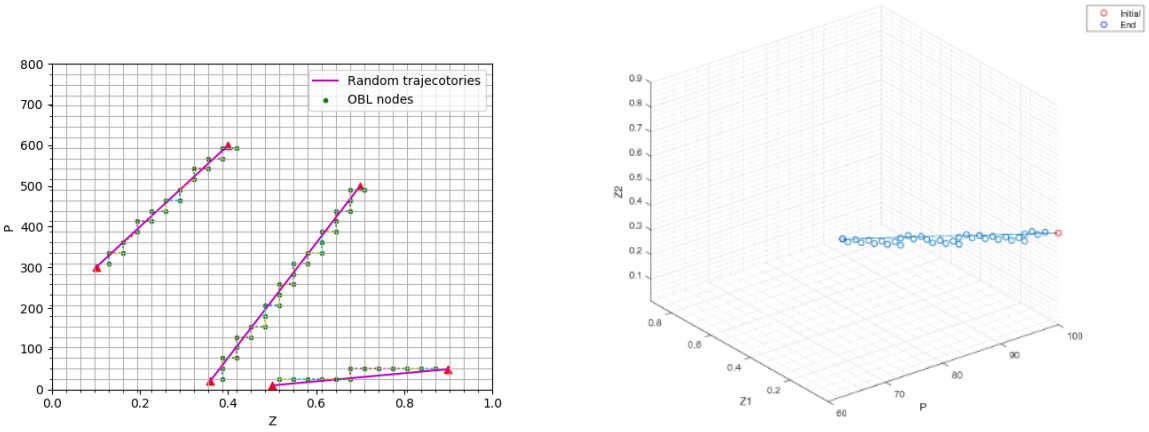
$$Q = d\mathbf{z}\mathbf{H}d\mathbf{z}'. \tag{17}$$

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.

Adaptive detection of trust regions

The full Hessian evaluation used in the trust-region definition 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 follow:

1. detect OBL interfaces along the nonlinear update trajectory in the parameter space (e.g. Fig. 1),
2. compute directional second derivative for each convection operator at each crossing interface,
3. detect inflection point(s) and kinks based on the second derivative information,
4. limit the local nonlinear update by the location inside the trust region.



(a) 2D parameter space

(b) 3D parameter space

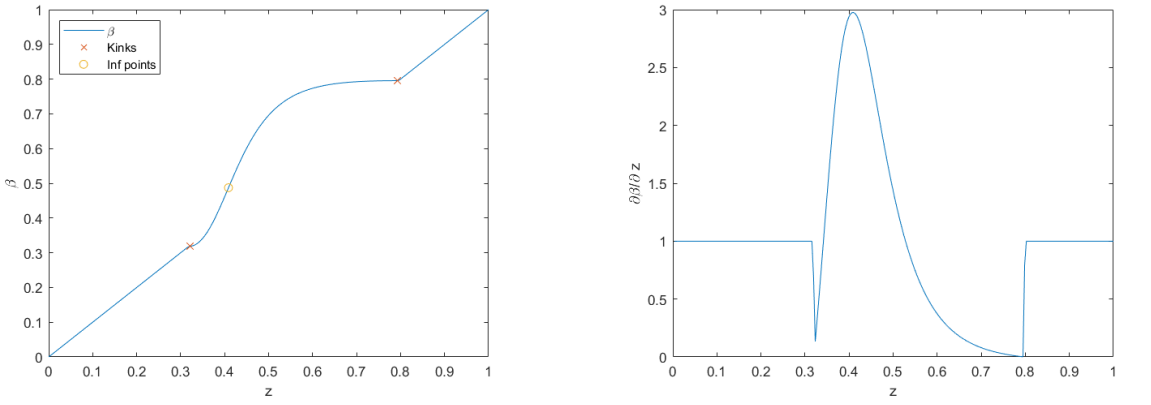
Figure 1: Tracking nonlinear update trajectories in the OBL parameter space

Next, we illustate our approach with exmaples and compare special points detected by directional analysis and reference results based on the construction of the full hessian matrix for binary and ternary systems.

Nonlinear analysis of binary kernel

Hessian is constructed by finite difference method based on first derivatives computed by OBL. Since operators are evaluated using piece-wise linear interpolation, the derivatives cannot be defined in the OBL nodes. Re-sampling points in between the original points where derivatives are essentially constant is problematic. To overcome this obstacle, we need to re-sample at different resolution with respect to the original OBL grid.

As we can see in Fig. 2 for 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 phase transition occurs. Kinks have different properties than inflection points and usually have a negative impact on nonlinear converges (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 in this points.



(a) Fractional flow

(b) Derivative of the fractional flow with respect to z

Figure 2: Binary incompressible compositional kernel

In Fig. 3, we detect an inflection line position in the parameter space based on the convex condition

approximated by quadratic form of the Hessian matrix and compare it to the inflection point detected during tracking the nonlinear update. In this example, we use dead-oil physical kernel with viscosity dependent on pressure which explains the form of inflection line. Notice, 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.

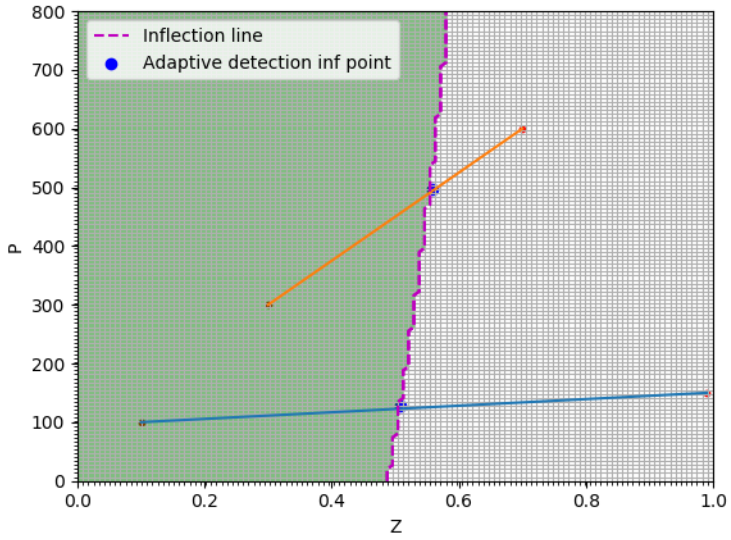


Figure 3: Two trajectories illustrating an adaptive detection of inflection point procedure by directional derivative analysis

Nonlinear analysis of ternary kernel

We extend our TR strategy for the ternary case for which we need to track the nonlinear update in the three dimensional parameter space. In fact, algorithm for tracking is independent to the degree of freedom of the system and is able to track nonlinear trajectories for an arbitrary number of dimensions. In our analysis for the ternary system, we adapt the same method as in the binary test by investigating the compositional system for a fix pressure. In this case we are interested in variation of convective operators with respect to z_1 and z_2. Accordingly, we construct the Hessian matrix for the fix pressure as follow:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \beta}{\partial z_1^2} & \frac{\partial^2 \beta}{\partial z_1 \partial z_2} \\ \frac{\partial^2 \beta}{\partial z_2 \partial z_1} & \frac{\partial^2 \beta}{\partial z_2^2} \end{bmatrix}, \tag{18}$$

Figure 4 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 phase diagram, red color corresponds to two-phase region and blue color corresponds to the single-phase region. In Hessian diagram, each color corresponds to different convex condition of the flux operators. Comparing Hessian diagram to phase diagram, it is clear that there is an an abrupt changes in concavity (kink) on the boundaries between single phase and two-phase regions. Moreover, there is an inflection line in two-phase region that segment two-phase zone into concave and convex part.

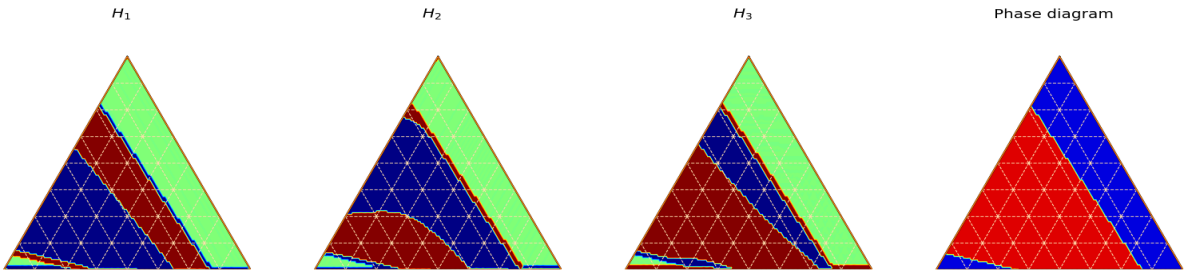
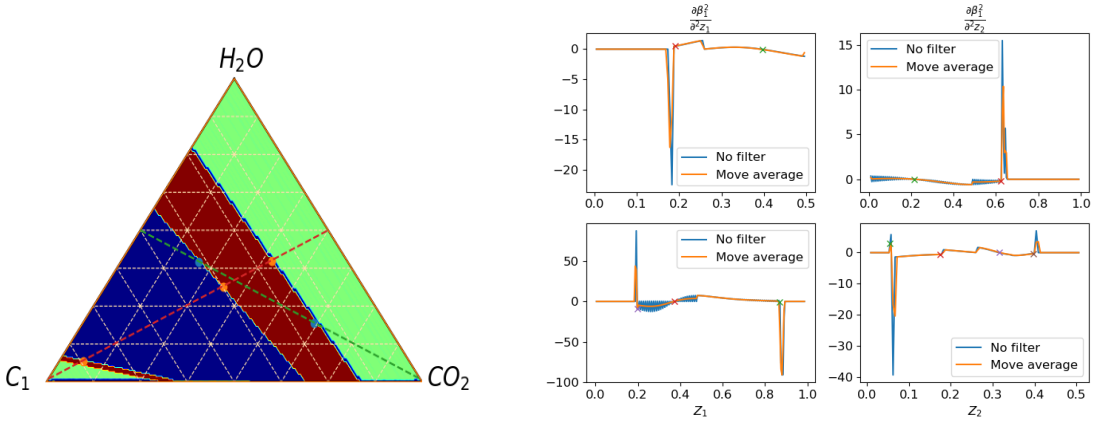


Figure 4: Hessian for three convective operators and phase diagram for incompressible ternary kernel

Computing hessian in each points of the parameter space is computationally expensive and cumbersome. Therefore, we focus on directional analysis in parameter space in which we look at one direction at each time passing an interface in the parameter space. Note that there is discontinuity in derivatives which corresponds to the boundaries between single- and two-phase regions. Drawing random trajectories for the nonlinear update, we show in Fig. 5 that special points can be detected using a 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 moving average algorithm (Gilgen, 2006) to smooth second order directional derivatives.



(a) Random trajectories on H_1

(b) Directional derivatives of the random trajectories.

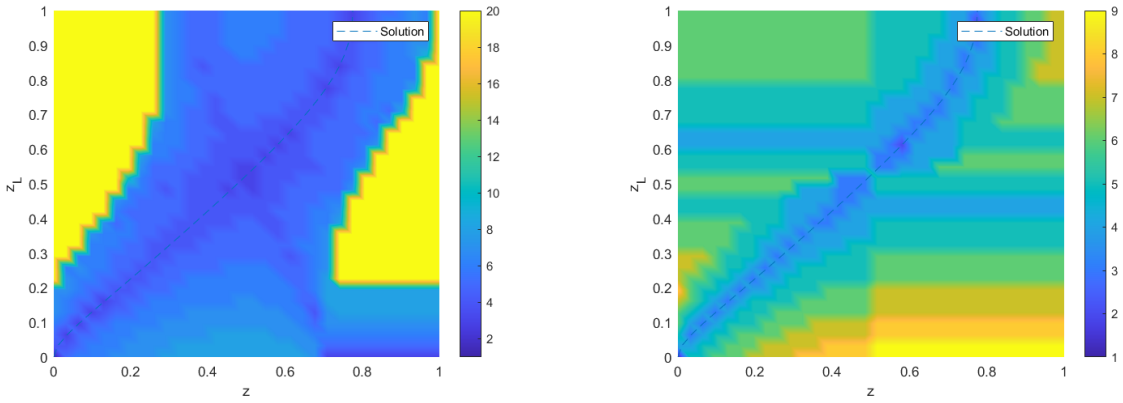
Figure 5: Full and directional evaluations of second order behavior for ternary compositional kernel.

Performance of nonlinear solver

In this section, we illustrate the performance of the proposed nonlinear solver applied for several simulation problems of increasing complexity.

Nonlinear convergence for single cell problem

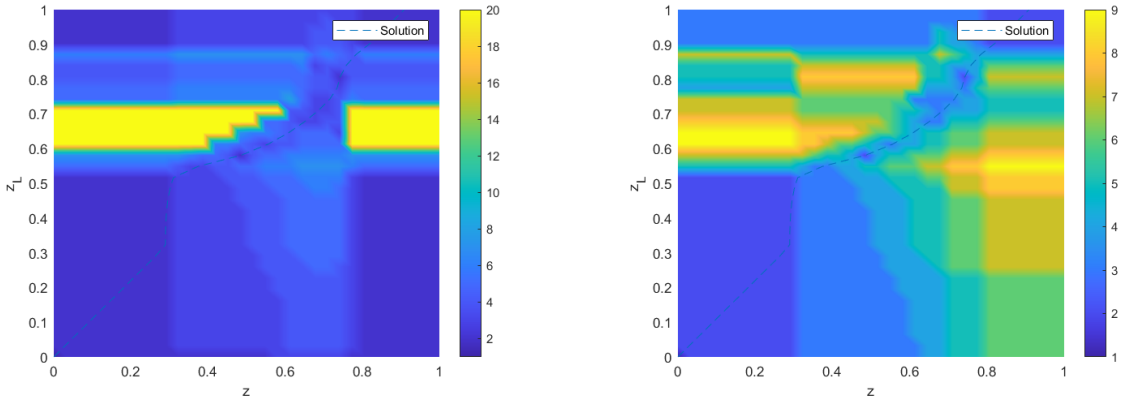
By analyzing the nonlinear behaviour of the single cell binary compositional transport problem, we can reveal some fundamental conclusion that are also valid for multiple cell problems. Our goal is to find the solution z^{n+1} from the initial guess $z^{n+1,0}$ for the given boundary condition on left and right sides (z_l and z_r). We investigate the convergence map of the pure newton and trust region solver for dead-oil and binary compositional kernels. We fix the right boundary conditions as $z_r = 1$ and study if the root of the residual is found by different nonlinear solvers for all possible starting points $(z^{n+1,0}, z_l) \in (0,1) \times (0,1)$. Maximum nonlinear iteration for all these test cases is equal to 20.



(a) Convergence map for pure newton

(b) Convergence map for TR algorithm

Figure 6: Convergence map for binary dead-oil problem

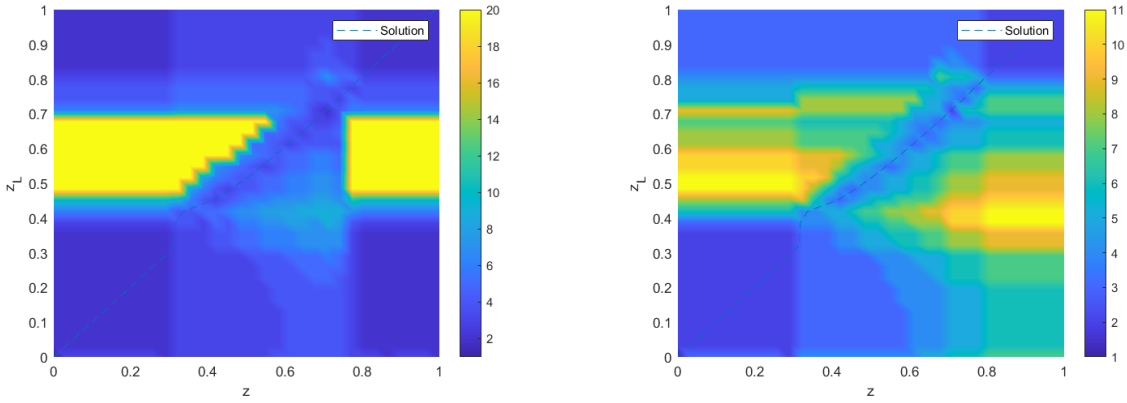


(a) Convergence map for pure newton

(b) Convergence map for TR algorithm

Figure 7: Convergence map for compositional binary problem with C=$\frac{dt}{dx}$ = 10

As it is clear from Fig. 7, pure newton struggles once the solution is in the two-phase region. Another observation is that once the solution is in the single phase region (linear fractional flow), nonlinear convergence based on the Newton's method is guaranteed.



(a) Convergence map for pure newton

(b) Convergence map for TR algorithm

Figure 8: Convergence map for compositional binary problem with C=$\frac{dt}{dx}$= 100

As we increase the timestep (C=$\frac{dt}{dx}$), we can observe that for pure Newton's approach, the yellow region increases when the solution is in the two-phase region. However, once the solution is in the single phase region pure newton is able to find the solution. Trust Region solver is globally convergent for all initial guesses.

Front propagation in fracture

In fracture reservoir, the difference in 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 one dimensional reservoir by running the simulation with small timesteps to develop a resolved displacement solution at particular time. Next, we restart the simulation from this distribution for a one control timestep only and account for the number of nonlinear iterations required to converge the solution. We repeat this procedure gradually increasing the size of the control timestep and detect the change in the number of nonlinear iterations.

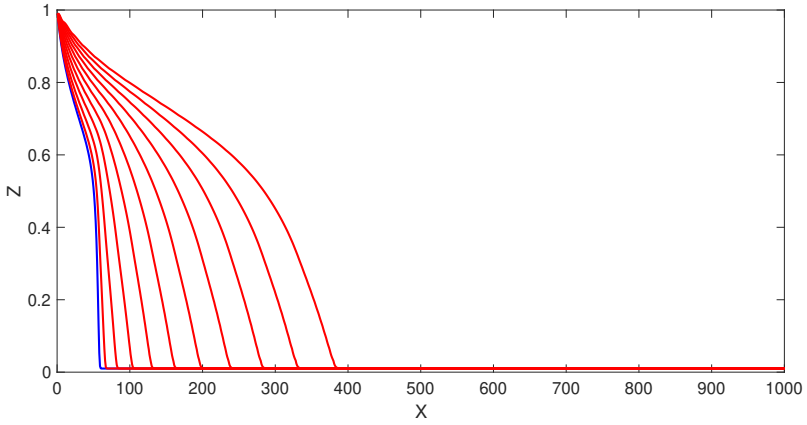


Figure 9: Saturation propagation for different timesteps for a dead-oil kernel from Appendix A

Next we test the performance of nonlinear solver based on directional analysis in the proposed numerical framework using several physical kernels with subsequent grow in complexity. In addition, we make comparisons between our trust-region nonlinear strategy and different state-of-the-art solvers.

Global and local nonlinear solvers can be seen as different methods to damp the newton updates by specifying the diagonal matrix Φ= diag$(\Delta z_1, \Delta z_2, ..., \Delta z_n)$ that can be written in each nonlinear iteration in the general form:

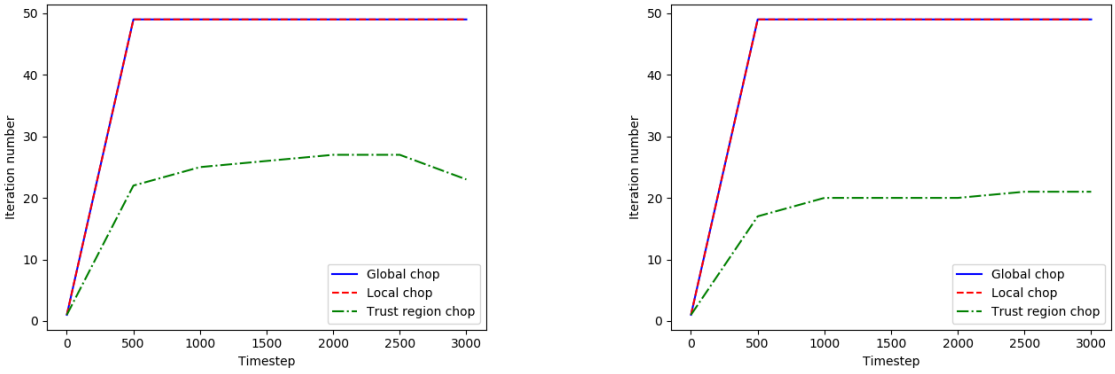
$$\Delta \boldsymbol{\omega} = -\Phi \mathbf{J}^{-1}\mathbf{r}, \tag{19}$$

The standard Newton's method select all of these diagonal weights to be unity. In global-chop nonlinear strategy, all entries of the diagonal are identical, implying that the Newton direction is simply scaled by a constant factor. In the local-chop nonlinear solver, the diagonal scaling entries are not necessarily identical and can change on a cell-by-cell basis to limit the local compositional update Δz to be greater than the specified number ($\Delta z = 0.1$ in our study).

Binary systems

For binary systems, we test nonlinear solvers performance for both dead-oil and compositional kernels with constant k-values equal to $= \{2.5, 0.3\}$. Initially, the 1D domain is fully saturated by non-wetting phase and we inject wetting phase at the left boundary. We ran the simulation with the small timestep for 1000 days. Next, we restart by enlarging the timesteps. Figure 10 shows the fracture test results comparing different nonlinear solvers for binary kernel. We set the maximum number of nonlinear

iterations to 50 for all test cases. It is clear that trust region solver performs better (provide less nonlinear iterations) for both dead-oil and compositional kernels.



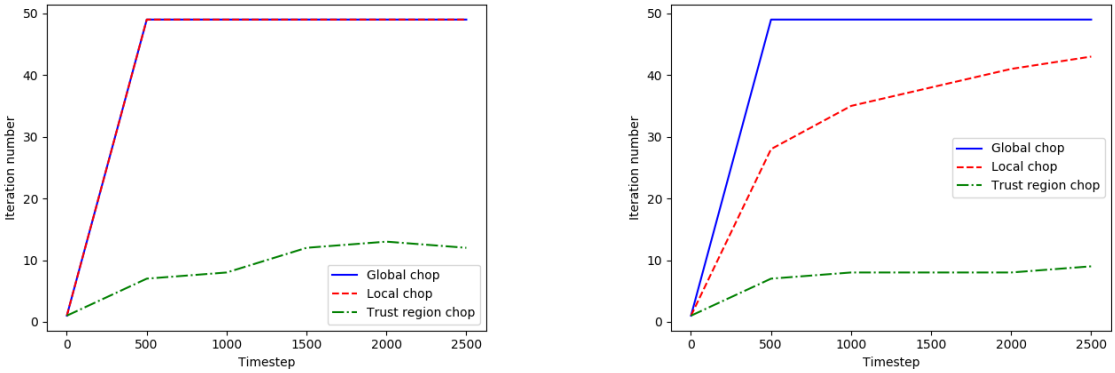
(a) Dead-oil kernel

(b) Compositional kernel

Figure 10: Different nonlinear solver performance for binary systems

Ternary and Quaternary systems

For ternary system, we repeat a similar test with two systems of $\{CO_2, H_2O, C_1\}$ and 4 components $\{CO_2, H_2O, C_1, C_2\}$. We inject at 135 bar using bottom hole pressure control and initial reservoir pressure is 95 atm. For ternary system, reservoir initial condition is $z_{ini} = \{0.1, 0.25\}$ and injection condition $z_{inj} = \{0.98, 0.01\}$ and for 4 component system injection condition $z_{inj} = \{0.98, 0.001, 0.001\}$ and initial condition $z_{ini} = \{0.1, 0.2, 0.2\}$. We ran the full simulation for 100 days. Next, we restart with enlarging the timestep. Figure 11 illustrates different nonlinear solvers performance. TR solver works better for both ternary and four components case.



(a) Ternary kernel

(b) 4 component

Figure 11: Comparison of different nonlinear solvers for incompressible ternary kernel

Nonlinear preconditioning strategy

The proposed trust-region nonlinear solver guarantees to converge for arbitrary timesteps. However, its application may become time prohibitive due to excessive number of nonlinear iterations for large timesteps. To overcome this issue, we proposed the nonlinear preconditioning strategy described in this section. In the OBL framework, we look into the sequential update on nonlinear iterations where changes in transport unknowns in the downstream block depends on the derivatives of convective operator at the

current block. If composition in the block is defined at the residual values (when convection for the phase is absent), the update in the downstream block is zero, and transport wave cannot propagate on this iteration. As the result, the nonlinear solver needs as many iterations as many blocks at residual compositions it needs to propagate.

This problem is especially pronounced in fractured reservoirs, where transport front have to propagate for a large number of grid-blocks once it reaches fracture. Here, we perform the analysis for the propagation of nonlinear wave in OBL approach which is a farther extension of the analysis performed in Wang (2012). To simplify derivations, we assume that the model is limited by a 1D reservoir with Cauchy boundary conditions on left and right side. This reduces the spacial discretization, which yields to the following equation in vector form (the length of vector corresponds to the number of components n_c) for the block i:

$$\mathbf{r}_i(\boldsymbol{\omega}_{i-1},\boldsymbol{\omega}_i,\boldsymbol{\omega}_{i+1},\boldsymbol{\omega}_i^n) = (\boldsymbol{\alpha}(\boldsymbol{\omega}_i)-\boldsymbol{\alpha}(\boldsymbol{\omega}_i^n))a_i - \boldsymbol{\beta}(\boldsymbol{\omega}_i)b_{i^+}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_{i+1}) + \boldsymbol{\beta}(\boldsymbol{\omega}_{i-1})b_{i^-}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_{i-1}), \quad (20)$$

where

$$a_i = \phi_0 V_i, \quad (21)$$

$$b_{i^+}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_{i+1}) = \Delta t T_{i,i+1}(p_{i+1}-p_i)\Lambda(\boldsymbol{\omega}_i), \quad (22)$$

$$b_{i^-}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_{i-1}) = \Delta t T_{i-1,i}(p_i-p_{i-1})\Lambda(\boldsymbol{\omega}_{i-1}). \quad (23)$$

For simplicity, we assume a homogeneous reservoir with V, ϕ_0 and T constants. Equation 20 can be written for an internal reservoir block as

$$\mathbf{r}_i = (\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_i^n) + \gamma(\boldsymbol{\beta}_i b_{i+} + \boldsymbol{\beta}_{i-1} b_{i^-}). \quad (24)$$

Here

$$\boldsymbol{\alpha}_i = \boldsymbol{\alpha}(\boldsymbol{\omega}_i), \quad \boldsymbol{\beta}_i = \boldsymbol{\beta}(\boldsymbol{\omega}_i), \quad \gamma = \Delta t \frac{T^{ab}}{\phi_0 V}, \quad (25)$$

and

$$b_{i^+} = (p_i - p_{i+1})\Lambda(\boldsymbol{\omega}_i), \quad b_{i^-} = (p_i - p_{i-1})\Lambda(\boldsymbol{\omega}_i - 1). \quad (26)$$

Now the internal Jacobian row of the equation can be written as:

$$\begin{bmatrix} \gamma \mathbf{B}_{i-1} b_{i^-} + \gamma \boldsymbol{\beta}_{i-1} \times \mathbf{b}'_{i^-,i-1} \\ \mathbf{A}_i + \gamma(\mathbf{B}_i b_{i+} + \boldsymbol{\beta}_i \times \mathbf{b}'_{i+,i} + \boldsymbol{\beta}_{i-1} \times \mathbf{b}'_{i^-,i}) \\ \gamma \boldsymbol{\beta}_i \times \mathbf{b}'_{i+,i+1} \end{bmatrix}^T, \quad (27)$$

where

$$\mathbf{A}_i = \left[\frac{\partial \boldsymbol{\alpha}_i}{\partial \boldsymbol{\omega}_i}\right] = \left[\frac{\partial \alpha_c}{\partial p_i} \frac{\partial \alpha_c}{\partial z_{i,1}} \ldots \frac{\partial \alpha_c}{\partial z_{i,n_c-1}}\right], c = 1,\ldots,n_c, \quad (28)$$

$$\mathbf{B}_i = \left[\frac{\partial \boldsymbol{\beta}_i}{\partial \boldsymbol{\omega}_i}\right] = \left[\frac{\partial \beta_c}{\partial p_i} \frac{\partial \beta_c}{\partial z_{i,1}} \ldots \frac{\partial \beta_c}{\partial z_{i,n_c-1}}\right], c = 1,\ldots,n_c, \quad (29)$$

$$\mathbf{b}'_{i^-,i-1} = \left[\frac{\partial b_{i-}}{\partial \boldsymbol{\omega}_{i-1}}\right]^T = (p_i - p_{i-1})\frac{\partial \Lambda_{i-1}}{\partial \boldsymbol{\omega}_{i-1}} - \begin{bmatrix} \Lambda_{i-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (30)$$

$$\mathbf{b}'_{i-1,i} = \left[\frac{\partial b_{i-}}{\partial \boldsymbol{\omega}_i}\right]^T = \begin{bmatrix} \Lambda_{i-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (31)$$

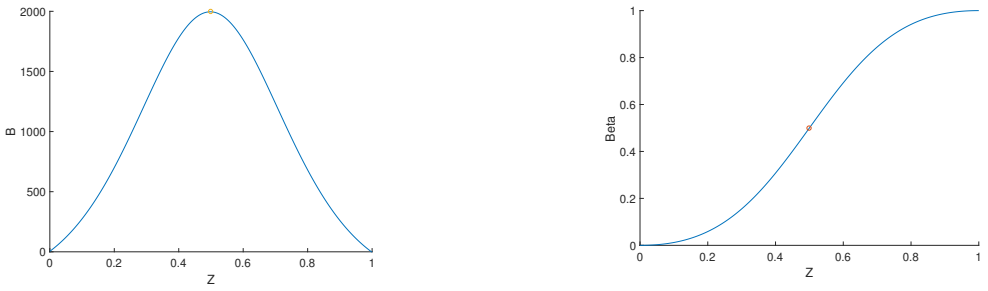
$$\mathbf{b}'_{i+,i} = \left[\frac{\partial(-b_{(i+1)^-})}{\partial \boldsymbol{\omega}_{(i+1)-1}}\right] = -\mathbf{b}'_{(i+1)-,(i+1)^-1}, \tag{32}$$

$$\mathbf{b}'_{i+,i+1} = \left[\frac{\partial(-b_{(i+1)^-})}{\partial \boldsymbol{\omega}_{(i+1)}}\right] = -\mathbf{b}'_{(i+1)-,(i+1)}. \tag{33}$$

For binary system, under the assumption of incompressible fluid, α_c and β_c are equivalent to overall composition z_c and the compositional fractional flow curve (F_c) respectively. When the residual saturation is zero at the initial conditions, the β-operator becomes zero since the compositional fractional flow is zero at residual saturation equals to zero. Therefore, the internal Jacobian for the nonlinear iteration can be written as:

$$\begin{bmatrix}\gamma \mathbf{B}_{i-1} b_{i^-} & 1+\gamma(\mathbf{B}_i b_{i+}) & 0\end{bmatrix}, \tag{34}$$

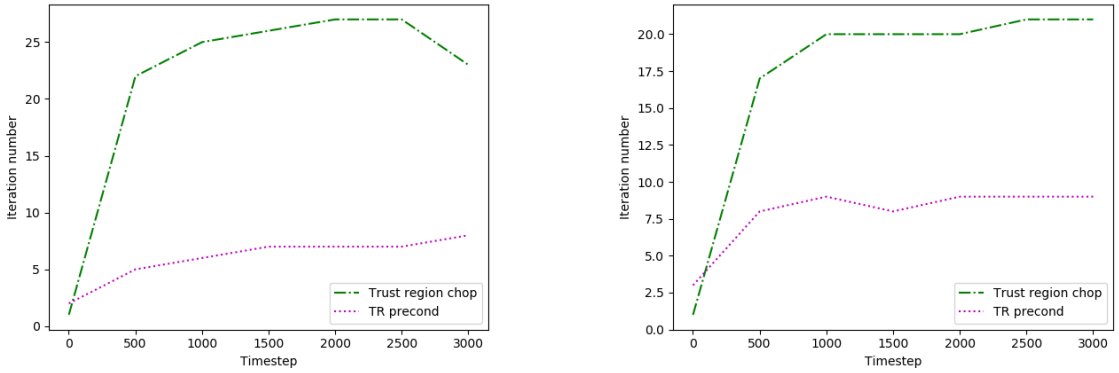
In Eq. 34, if **B** becomes zero at end point of fractional flow for the component, the internal jacobian matrix becomes identity matrix which does not allow the composition to propagate downstream. Therefore, to maximize the propagation for the composition front downstream in single iteration, we need to maximize **B** term.



(a) (b)

Figure 12: (a) The derivative of the β-operator with respect to z for dead-oil system (b) Fractional flow for dead-oil system

From Fig. 12 for the incompressible version of dead-oil kernel without dependency of viscosity on pressure, it is clear that the **B** is maximum at the inflection point. Therefore, to maximize the composition propagation in more than one gridblocks in one iteration, we introduced the inflection point of the β-operator for each gridblock as an initial guess. Fig. 13 compares our nonlinear solver performance with and without preconditioning for both dead-oil with pressure variation and binary compositional kernel. We can see that by applying preconditioning, the number of iterations for increasing timestep growing slower for the preconditioned system. We are currently working on the similar preconditioning strategy for general compositional systems.



(a) Dead-oil kernel

(b) Compositional kernel

Figure 13: TR nonlinear solver with/withoout preconditioning

Conclusions

In this work, we investigate the nature of nonlinearities in general-purpose simulation and propose solution methods for a general compositional problem. The pure Newton's strategy will not guarantee to converge, and it is highly dependent on the timestep selection. We present an advanced nonlinear solver based on a trust-region technique aimed to solve multiphase multi-component transport problems. 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 associated with the governing equations. We track the nonlinear trajectory in the parameter-space and segment the parameter-space of the problem into the set of trust-region 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. The proposed nonlinear solver locally constraints the updating of the overall compositions across the boundaries of these regions.

Besides, we address the issue related to the slow convergence of the advanced nonlinear solver for large timestep and see how the number of iteration evolves nonlinearly by increasing the timestep in the fractured reservoir for the binary kernel. The solution of the transport problem often requires the propagation of displacement front to multiple control volumes per single timestep. To overcome this issue, we introduce the composition corresponding to the inflection point as preconditioning for the nonlinear solver. The nonlinear convergence benefits from such preconditioning, and the resulting number of iteration reduces significantly. We are working on extending this strategy for general compositional problems.

References

Aziz, K. and Settari, A. [1979] *Petroleum Reservoir Simulation*. Elsevier Applied Science.

Coats, K.H. [1980] An Equation of State Compositional Model. *Society of Petroleum Engineers*.

Collins, D., Nghiem, L., Li, Y.K. and Grabonstotter, J. [1992] An Efficient Approach to Adaptive-Implicit Compositional Simulation With an Equation of State. **83**, 111.

Deuflhard, P. [2004] *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg.

Gabor, A., Doleschall, S. and Farkas, E. [1985] General Purpose Compositional Models. *Society of Petroleum Engineers*.

Garipov, T., Tomin, P., Rin, R., Voskov, D. and Tchelepi, H. [2018] Unified Thermo-Compositional-Mechanical Framework for Reservoir Simulation. *Computational Geosciences*.

Geoquest [2011] ECLIPSE Technical Description. Tech. rep., Schlumberger.

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

Jenny, P., Tchelepi, H.A. and Lee, S.H. [2009] Unconditionally convergent nonlinear solver for hyperbolic conservation laws with S-shaped flux functions. *Journal of Computational Physics*, **228**(20), 7497 – 7512.

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

Khait, M. and Voskov, D.V. [2017] Operator-based linearization for general purpose reservoir simulation. *Journal of Petroleum Science and Engineering*, **157**, 990 – 998.

Khait, M. and Voskov, D.V. [2018] Adaptive Parameterization for solving of Thermal/Compositional Nonlinear flow and transport with Buoyancy. *SPEJ*.

Li, B. and Tchelepi, H.A. [2015] Nonlinear analysis of multiphase transport in porous media in the presence of viscous, buoyancy, and capillary forces. *Journal of Computational Physics*, **297**, 104 – 131.

Michelsen, M.L. [1982] The isothermal flash problem. Part II. Phase-split calculation. *Fluid Phase Equilibria*, **9**(1), 21 – 40.

Ortega, J. and Rheinboldt, W. [1970] *Iterative Solution of Nonlinear Equations in Several Variables*. Classics in Applied Mathematics. Elsevier.

Pruess, K., Fensterle, S., Moridis, G., Oldenburg, C. and Wu, Y. [1997] General Purpose Reservoir Simulators: the TOUGH2 Family. *Journal of Computational Physics*, **62**(2), 265 – 281.

Voskov and Tchelepi [2011] Compositional Nonlinear Solver Based On Trust Regions Of The Flux Function Along Key Tie-Lines. *Society of Petroleum Engineers*.

Voskov, D. and Tchelepi, H. [2012] Comparison of nonlinear formulations for two-phase multi-component EoS based simulation. *Journal of Petroleum Science and Engineering*, **82-83**, 101–111.

Voskov, D.V. [2017] Operator-based linearization approach for modeling of multiphase multi-component flow in porous media. *Journal of Computational Physics*, **337**, 275 – 288.

Wang, X. [2012] *TRUST-REGION NEWTON SOLVER FOR MULTIPHASE FLOW AND TRANSPORT IN POROUS MEDIA*. Ph.D. thesis, Stanford University.

Wang, X. and Tchelepi, H.A. [2013] Trust-Region Newton Solver for Multiphase Flow and Transport in Heterogeneous Porous Media.

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

Appendix A: Parameters for numerical tests

The parameter defined in the table 1 are common for all the cases.

Dead oil fluid description

Table 1: Rock-Fluid parameters

Parameter	Value	Decription
C_r	1.00E-09	Rock compressibility
S_{pr}	0.01	Phase residual saturation
S_{or}	0.01	Oil residual saturation
n_o	2	Oil exponents

Table 2: Dead oil properties

Parameter	Oil	Descrpition
ρ	1000	Surface density
μ	1-5 cp	Viscosity range
B_o	1	Formation Volume Factor

Parameter	Water	Desctiption
ρ	1000 kg/cm$\hat{3}$	surface density
c_p	10^{-9}	Compressibility
μ	1cp	Viscosity

Compositional properties

Table 3: Thermodynamic properties

Components	C_1	CO_2	H_2	C_2
K-values	6	0.4	0.1	0.02

Table 4: Phase properties

Parameter	Water	Gas	Desctiption
n	2	2	Corey exponent
μ	1.5cp	0.1cp	Viscosity

Appendix B: Sensitivity of preconditioning to transport parameters

Next, we report the performance of the advanced nonlinear solver after running the simulation with the small timestep ($\Delta t = 0.001$) days until ($T = 0.3$) days and save the solution. Next, we restart from the solution for one large timestep equals to 0.2 days with different preconditioning strategy. Considering the initial guess the inflection point and initial guess equals to the initial condition and initial guess the solution of the saturation after running the simulation until $T = 0.3d$. We test the transport problems for four types of the relative permeability curves.

$$K_{rw} = S^{n_w}; K_{ro} = (1-S)^{n_o}, \tag{35}$$

Table 5: Different preconditioning strategy

K_{rp}	M	Initial guess from		
		without preconditioning	new timestep	inflection point
$n_o = n_p$ =2	0.5	60	13	9
	1	56	14	9
	10	11	4	10
$n_o = n_p$ = 3	0.5	142	22	11
	1	149	25	10
	10	111	5	10
$n_o = n_p$ = 10	0.5	198	27	10
	1	204	27	12
	10	229	30	11
n_o = 2, n_p = 10	0.5	142	20	11
	1	149	21	10
	10	111	22	10

It is noticed that:

(i) With preconditioning strategy, we accelerate the convergence for all the cases. As expected, choosing the inflection point as an initial guess always works better since the derivative **B** is maximized which maximizes the propagation of the compositional front downstream.

(ii) By increasing the mobility ratio, the number of iteration generally increase for the same time step and the same exponents. It can be explained by the fractional flow theory when by increasing the M (unfavorable displacement), the shock speed increases. The preconditioning strategy helps in all cases.