Sequential-implicit Newton method for multiphysics simulation

Wong, Zhi Yang; Kwok, Felix; Horne, Roland N.; Tchelepi, Hamdi A.

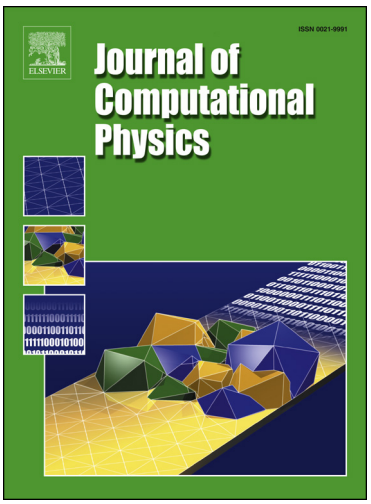
Published in:
Journal of Computational Physics

DOI:
10.1016/j.jcp.2019.04.023

Published: 15/08/2019

Document Version:
Peer reviewed version

Accepted Manuscript



Sequential-Implicit Newton Method for Multiphysics Simulation

Zhi Yang Wong, Felix Kwok, Roland N. Horne, Hamdi A. Tchelepi

PII: S0021-9991(19)30256-6
DOI: https://doi.org/10.1016/j.jcp.2019.04.023
Reference: YJCPH 8624

To appear in: *Journal of Computational Physics*

Received date: 1 October 2018
Revised date: 10 April 2019
Accepted date: 11 April 2019

Please cite this article in press as: Z.Y. Wong et al., Sequential-Implicit Newton Method for Multiphysics Simulation, *J. Comput. Phys.* (2019), https://doi.org/10.1016/j.jcp.2019.04.023

Highlights

- Sequential-implicit Newton's method developed for multiphysics problems.
- Improves sequential-implicit methods from a linear to a quadratic convergence rate.
- Effective variable splitting and coupling condition is required for Sequential-implicit Newton's method.
- Improvement in convergence rate is shown for geomechanics and geothermal porous-media problems.

Sequential-Implicit Newton Method for Multiphysics Simulation

Zhi Yang Wong[a], Felix Kwok[b], Roland N. Horne[a], Hamdi A. Tchelepi[a,*]

[a]*Department of Energy Resources Engineering, Stanford University, 367 Panama St, Stanford, CA, USA*
[b]*Department of Mathematics, Hong Kong Baptist University, FSC 1209, Fong Shu Chuen Building, Kowloon Tong, Hong Kong*

Abstract

Efficient simulation of multiphysics problems is a challenging task. This is often due to the multiscale nature of the physics and nonlinear coupling between the different processes. One approach to this problem is to solve the entire multiphysics problem simultaneously in a fully coupled manner. However, due to the strong coupling and multiphysics interactions, it is difficult to design and analyze fully coupled solvers, which often entail the construction and solution of global fully coupled Jacobian systems. Another approach is the sequential-implicit method, whereby the full multiphysics problem is split into different subproblems. Each subproblem is constructed and solved separately; then the solutions of the subproblems are stitched together in a specific sequence. Isolation of each subproblem allows for the design of specialized solvers that tackle the complexity of the particular subproblem efficiently. The sequential-implicit approach offers wide flexibility and extensibility. However, these advantages are often offset by slow convergence rates when there is strong coupling between the subproblems. This slow convergence rate of the overall coupled system is directly linked to the use of a fixed-point outer-loop iteration in sequential-implicit methods. We present a Sequential Implicit Newton (SIN) approach, whereby the sequence of implicit subproblems is wrapped with an outer full Newton scheme. We demonstrate that the SIN formulation improves the overall convergence rate from linear to quadratic. The SIN method uses the same sequential-implicit scheme as the fixed-point method, but after each sequential iteration a Newton update is computed. Wrapping a Newton loop around the traditional sequential-implicit scheme leads to significant improvements in the convergence rate. The SIN method allows for the ability to split a multiphysics problem into individual subproblems while taking advantage of the quadratic convergence rate of the Newton method. We demonstrate the effectiveness of SIN using two different multiphysics porous-media problems: flow-thermal in geothermal reservoir simulation and flow-mechanics in geomechanics reservoir simulation. Just as with the fixed-point iteration version of the sequential-implicit method, SIN benefits from careful design of the constraints used to stitch the sequence of the subproblems. Our numerical

*Please address correspondence to Hamdi A. Tchelepi
Email addresses: zhiyangw@stanford.edu (Zhi Yang Wong), felix_kwok@hkbu.edu.hk (Felix Kwok), horne@stanford.edu (Roland N. Horne), tchelepi@stanford.edu (Hamdi A. Tchelepi)

experiments show that the SIN approach improves the overall convergence rate for all the nonlinear multiphysics problems considered. For specific cases where there is strong nonlinear coupling between the subproblems, we see up to two orders of magnitude decrease in the number of sequential iterations when using SIN compared with the fixed-iteration scheme.

Keywords: sequential-implicit, Newton's method, MSPIN, geothermal, geomechanics, reservoir simulation, multiphysics

1. Introduction

In the simulation of multiphase flow in porous media, one generally needs to solve very large systems of nonlinear equations arising from spatial and time discretization of the conservation equations of mass, energy, and momentum. These multiphysics problems yield systems of equations that are nonlinear and coupled. The length and time scales associated with the different transport phenomena may be quite different, and the interactions between the different physical effects usually lead to complex nonlinear dynamics. The nonlinear nature of the problem and the strong coupling between the different equations and the constitutive relations pose serious challenges to the nonlinear solver. Newton-based solvers often lose the quadratic convergence and even diverge due to a poor choice of the initial guess [1]. In practice, the system of equations is often dealt with using specialized solvers that resolve the complexities of the different physical processes separately. If the physical processes lie in different regions in space, then fixed-point methods of the domain decomposition type [2, 3, 4] are often sufficient to isolate the nonlinearities, yielding convergent and efficient solvers. These domain decomposition methods can also be used as a preconditioner to Newton's method; prominent examples of this approach are the additive Schwarz preconditioned inexact Newton (ASPIN) method [5] and its two-level variants [6, 7]. A survey on composing scalable nonlinear algebraic solvers can be found in [8].

If the nonlinear dynamics are separated instead in the phase-field, i.e., when different variables follow different dynamics, then one can use sequential-implicit methods - a class of fixed-point methods whereby one alternately solves for some of the variables while fixing the remaining ones. Such a sequential-implicit strategy was proposed for isothermal multiphase flow and transport problems in which a multiscale finite-volume formulation was used [9]. In this approach, each iteration entails solving a sequence of implicit subproblems. In their sequential-implicit fixed-point (SIFP) method, Jenny et al. [9] solved (1) the near-elliptic flow problem and (2) the highly nonlinear hyperbolic transport problems, and they iterated until convergence is achieved. A great deal of work has been invested to improve the overall convergence rate of the SIFP scheme. The efforts have focused on modifying the coupling strategies between the subproblems [10, 11, 12, 13, 14].

In this investigation, we were interested in improving the efficiency of sequential-implicit fixed-point (SIFP) methods by wrapping them inside a full Newton scheme. The focus here is on geomechanics and geothermal porous-media problems, but the approach is of general applicability.

SIFP methods have proven to be attractive for solving flow-mechanics problems, where a fully coupled approach was shown to suffer from significant linear solver scalability issues [15, 16, 17, 18]. A sequential strategy reduces the complexities and requirements on the linear solver; thus making them easily scalable in comparison to a fully coupled approach. Sequential coupling of flow and mechanics can also be beneficial when the two problems have different computational domain sizes (geomechanics domain is often much larger than flow), different spatial discretization schemes (finite volume for flow and finite element for mechanics) or different simulators for flow and mechanics (e.g. TOUGH2 [19] for flow/thermal and ABAQUS for mechanics [20]). To improve on these sequential methods, the coupling between the flow and mechanics problem has been extensively investigated. It was shown that the fixed-stress and undrained split are unconditionally stable and that the fixed stress scheme converges faster than the undrained split [21, 22, 23, 24]. However, the performance of the fixed stress method is strongly dependent on the coupling strength between the flow and mechanics problems [21]. If the coupling strength is too high, this could result in a slow convergence rate consequently requiring too many sequential iterations to converge to the solution of the coupled problem. This slow convergence limits the capability of these SIFP methods.

The sequential-implicit method was also investigated for the flow and thermal problem in geothermal reservoir simulation. Inspired by the different splitting strategies for flow and mechanics, the authors of [25] investigated enforcing different constraints when solving the flow and thermal equations. They found that a naive split based on fixing the enthalpy when solving the flow equation and fixing the pressure when solving the thermal equation converged for single-phase cells, but diverged for two-phase cells. As a result, they developed a hybrid method where a fixed pressure was used for single-phase cells and fixed density for two-phase cells. Although this proved to be the best out of the sequential schemes examined, it still suffered from a large number of outer loop iterations for strongly coupled flow and thermal problems. Similar to Moncorgé et al. [10], to improve the outer loop convergence, Wong et al. [26] enriched the flow equations based on the phase state and the Courant-Friedrichs-Lewy (CFL) number of the cells. This improved the convergence of the outer loop for the sequential formulation; however, these additional equations increased the cost of each sequential iteration.

We observe that the main bottleneck in SIFP methods lies in the outer fixed-point loop, which typically converges linearly (i.e., the error decreases to zero like a geometric sequence). To obtain superlinear convergence, one way is to apply Anderson acceleration [27], which is essentially a nonlinear analogue of GMRES [28]; the Anderson iteration is also related to multisecant quasi-Newton methods

[29] and is convergent when the underlying fixed-point map is a contraction [30]. Such an acceleration has been applied successfully for fixed-stress splitting schemes for nonlinear poromechanics of unsaturated materials [31]. Another approach is to use the fixed-point map as a preconditioner to Newton's method, which is the idea behind the multiplicative Schwarz preconditioned inexact Newton (MSPIN) algorithm proposed in [32]. MSPIN uses a partitioning of the primary variables by field type, and it solves for the groups of variables successively in a nonlinear multiplicative Schwarz scheme. This mapping is then used to precondition Newton's method with an approximate Jacobian, similar to ASPIN. MSPIN has been shown to be effective for high Reynolds number Navier-Stokes problems; however, as already observed in [32]: "the determination of the partition of the physical variables can be the most interesting part of implementation, because the best choice is generally problem-specific." Our own observations indicate that, in addition to the choice of variable sets, the coupling conditions between these sets are equally important to the efficiency of method, just like for SIFP methods.

Thus, our contribution in this work was to improve on the MSPIN approach by identifying an effective variable splitting and coupling conditions for the flow-mechanics and flow-thermal problems. Our new method, called the Sequential Implicit Newton (SIN) method, is constructed by using SIFP as a preconditioner to Newton's method. One key difference of this method from MSPIN is the coupling conditions, which we implement by augmenting the sequential subproblems with physically motivated constraints. A second key difference is that we use an exact Jacobian to compute the Newton update rather than an inexact Jacobian as in MSPIN. A similar approach was used for the restricted additive Schwarz preconditioned exact Newton method (RASPEN) [33], but here our Newton loop is wrapped around a multiplicative Schwarz scheme. We show that matrix-vector multiplication involving the exact Jacobian can be performed by reusing matrix factorizations performed at earlier steps of the algorithm. Thus, our method enjoys both the local quadratic convergence of an exact Newton, but at a relatively low computational cost. This leads to a significant improvement over the unaccelerated SIFP methods, which we confirm using challenging numerical experiments.

The paper is organized as follows. In Section 2, we describe the sequential-implicit Newton method in detail for an abstract nonlinear system. In particular, we explain how the Jacobian matrix-vector product, which is required for calculating the Newton update using GMRES, can be calculated using matrix factorizations that have already been computed when solving the sequence of subproblems. In Section 3, we present the coupled flow-mechanics problem, and we show how to implement the SIN method for the partitioning of variables and coupling schemes widely used in the porous-media community. We then show numerical results illustrating the improvements obtained using SIN over the SIFP method. We do the same in Section 4, but for the flow-thermal problem. Conclusions are given in Section 5.

2. Sequential-implicit Newton's Method

In this section, we show the derivation of the sequential-implicit Newton's method for a general multiphysics problem. Assume that the multiphysics problem is given by a set of residual equations $R(x) = 0$ that can be split into $R(x) = (R_1(x), R_2(x))$, where R_1 and R_2 are nonlinear functions that represent different physical processes. Moreover, the unknowns themselves can be split into $x = (x_1, x_2)$, with the two groups of unknowns potentially following different dynamics. The fully coupled (FC) method is then simply Newton's method applied to the entire multiphysics problem with each Newton update is obtained by solving the linear system:

$$\begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \delta x_1^k \\ \delta x_2^k \end{bmatrix} = - \begin{bmatrix} R_1(x^k) \\ R_2(x^k) \end{bmatrix}, \tag{1}$$

for $(\delta x_1^k, \delta x_2^k)$, where $J_{ij} = \frac{\partial R_i}{\partial x_j}$ are the partial derivatives of the i^{th} subproblem residual equation with respect to the j^{th} subproblem primary variable set. The next iterate x^{k+1} is then defined by:

$$x_i^{k+1} = x_i^k + \delta x_i^k, \qquad i = 1, 2.$$

Although the focus of this study was to compare the sequential-implicit Newton method with the fixed-point iteration, it is useful to compare with the fully coupled method to understand the nonlinear behavior of the underlying coupled nonlinear problem.

2.1. Sequential-implicit Fixed Point Method

The sequential-implicit fixed point (SIFP) method involves partitioning the overall problem into multiple subproblems that are solved sequentially. In the case of a multiphysics problem, each of these subproblems will often correspond to a specific physical problem. Here we consider a splitting strategy that involves two different subproblems and an auxiliary constraint applied to each of the solution steps. The sequential-implicit scheme begins with an initial guess $x^0 = (x_1^0, x_2^0)$ for the entire problem. Here x_1 and x_2 correspond to the variable set for the two different subproblems. To solve for the next time step, the following sequential process is used:

1. Solve
$$\begin{cases} R_1(x_1^*, x_2^*) = 0, \\ b(x_1^*, x_2^*) = b(x_1^k, x_2^k), \end{cases} \tag{2}$$
for (x_1^*, x_2^*), using e.g. Newton's method.
2. Solve
$$\begin{cases} R_2(x_1^{**}, x_2^{**}) = 0, \\ c(x_1^{**}, x_2^{**}) = c(x_1^*, x_2^*), \end{cases} \tag{3}$$
for (x_1^{**}, x_2^{**}), using e.g. Newton's method.

3. Update solution:

$$x_1^{k+1} = x_1^{**}, \qquad x_2^{k+1} = x_2^{**}.$$

4. Repeat steps 1-3 until convergence is reached:

$$\left|\left|R_1(x_1^{k+1}, x_2^{k+1})\right|\right|_\infty \leq \epsilon_1 \text{ and } \left|\left|R_2(x_1^{k+1}, x_2^{k+1})\right|\right|_\infty \leq \epsilon_2 \tag{4}$$

A flow chart of the method is shown on the left panel of Figure 1. This algorithm is the current approach used in sequential-implicit porous-media simulations [21, 23, 22, 34, 14, 25, 35, 36]. Use of carefully chosen constraints $b(x_1, x_2)$ and $c(x_1, x_2)$ can enhance the convergence of the method. In domain decomposition methods, where the equations and unknowns are split across subdomain boundaries, the choice $b(x_1, x_2) = x_2$, $c(x_1, x_2) = x_1$ corresponds to a block Gauss-Seidel method, also known as a classical alternating Schwarz method with Dirichlet transmission conditions and minimal overlap [37]. Alternatively, if the constraints are chosen to match discrete Robin traces (i.e., a linear combination of function values and fluxes), then one obtains an optimized Schwarz method, which may converge a lot more quickly than block Gauss-Seidel [38, 39]. Note that although (2) and (3) formally have the same dimensions as the fully coupled problem, the constraints $b(x_1, x_2)$ and $c(x_1, x_2)$ are generally chosen to be very simple (cf. $b(x_1, x_2) = x_2$, $c(x_1, x_2) = x_1$ for the block Gauss-Seidel case), so the subproblems in practice have much smaller effective sizes and are much cheaper to solve than the fully coupled problem. For multiphysics problems, such constraints are often required to ensure convergence and stability [21, 22, 23, 25]. Through those studies, it was shown that a naive splitting of the subproblems and simply fixing the primary variables was insufficient for convergence and stability for these sequential schemes.

2.2. Sequential-implicit Newton method

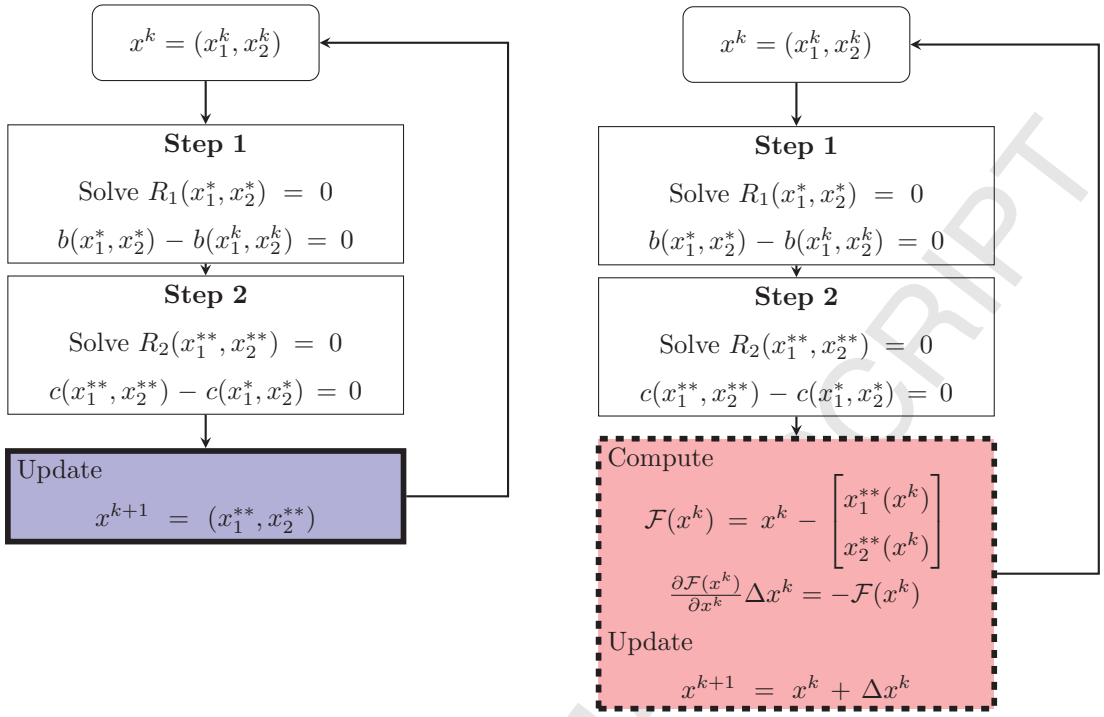
The key idea for accelerating the SIFP method is to note that at convergence, the fixed point of the SIFP method satisfies the equation:

$$x^{k+1} = (x_1^{**}(x^k), x_2^{**}(x^k)) = (G_1(x^k), G_2(x^k)) = G(x^k), \tag{5}$$

where $G_1(x^k)$ and $G_2(x^k)$ are the mappings defined by steps 1 and 2 in the SIFP method to obtain x_1^{**} and x_2^{**} based on the input (x_1^k, x_2^k). Thus, if we define the new function:

$$\mathcal{F}(x) = x - G(x), \tag{6}$$

then the fixed point of the SIFP method must be a solution of $\mathcal{F}(x) = 0$. The Sequential-implicit Newton (SIN) method consists of solving this equation using Newton's method.



(a) Sequential-implicit fixed-point (SIFP) (b) Sequential-implicit Newton (SIN)

Figure 1: Flowchart of sequential scheme using a fixed-point iteration update (Left) and a Newton update (Right)

A flowchart of this algorithm is presented in the right panel of Figure 1. The key difference with SIFP is how the solution is updated after all the subproblems are solved. SIFP takes the subproblem solutions $(x_1^{**}, x_2^{**}) = (G_1(x^k), G_2(x^k))$ as the new iterate, but SIN performs a Newton update by linearizing the fixed-point function $\mathcal{F}$ at x^k. The added complexity for SIN is that the Jacobian for the nonlinear function $\mathcal{F}$ is required. However, rather than computing an explicit representation of $\frac{\partial \mathcal{F}(x)}{\partial x}$, we exploit the fact that the Jacobian $\frac{\partial \mathcal{F}(x)}{\partial x}$ is only required when we solve the system of linear equations

$$\frac{\partial \mathcal{F}(x^k)}{\partial x} \Delta x^k = -\mathcal{F}(x^k). \tag{7}$$

Hence, if we have a routine for calculating the matrix-vector product involving the Jacobian $\left(\frac{\partial \mathcal{F}(x^k)}{\partial x}\right)$, then we can use a Krylov solver such as GMRES [28] to solve the linear equations, thus avoiding explicit calculation and storage of the Jacobian.

We will now explain how the Jacobian matrix-vector product for the SIN update can be computed. For notational convenience, we rewrite x_1^* and x_2^* calculated in Step 1 of SIFP as functions of the input arguments $x = (x_1, x_2)$, i.e., we define:

$$x_1^* = H_1(x), \qquad x_2^* = H_2(x).$$

Then, H_1 and H_2 satisfy:

$$R_1(H_1(x), H_2(x)) = 0 \tag{8}$$

and also the constraint:

$$b(H_1(x), H_2(x)) = b(\Pi_1 x, \Pi_2 x), \tag{9}$$

where Π_1 and Π_2 are projections that restrict the solution vector to its first and second block components, respectively (i.e., $\Pi_1 x = x_1$ and $\Pi_2 x = x_2$). Now, Step 2 of the algorithm can be written as:

$$R_2(G_1(x), G_2(x)) = 0, \tag{10}$$

$$c(G_1(x), G_2(x)) = c(H_1(x), H_2(x)). \tag{11}$$

We want to solve the equation $\mathcal{F}(x) = x - G(x) = 0$ using Newton's method, which involves repeatedly solving linear systems described in Equation (7). Solving this linear system using GMRES requires us to derive an expression for $\left(\frac{\partial \mathcal{F}}{\partial x}\right) v$, where v is an arbitrary vector. This can be done using implicit differentiation. Differentiating equations (8), (9) with respect to x leads to:

$$\frac{\partial R_1}{\partial x_1}(H_1(x), H_2(x))\frac{\partial H_1}{\partial x} + \frac{\partial R_1}{\partial x_2}(H_1(x), H_2(x))\frac{\partial H_2}{\partial x} = 0, \tag{12}$$

$$\frac{\partial b}{\partial x_1}(H_1(x), H_2(x))\frac{\partial H_1}{\partial x} + \frac{\partial b}{\partial x_2}(H_1(x), H_2(x))\frac{\partial H_2}{\partial x} = \frac{\partial b}{\partial x_1}(\Pi_1 x, \Pi_2 x)\Pi_1 + \frac{\partial b}{\partial x_2}(\Pi_1 x, \Pi_2 x)\Pi_2. \tag{13}$$

Multiplying Equations (12) and (13) from the right by an arbitrary vector v and noting that $\Pi_i x = x_i$, $\Pi_i v = v_i$ for $i = 1, 2$, we can rewrite the result as a linear system of the form:

$$\begin{bmatrix} \frac{\partial R_1}{\partial x_1}(H_1(x), H_2(x)) & \frac{\partial R_1}{\partial x_2}(H_1(x), H_2(x)) \\ \frac{\partial b}{\partial x_1}(H_1(x), H_2(x)) & \frac{\partial b}{\partial x_2}(H_1(x), H_2(x)) \end{bmatrix} \begin{bmatrix} \frac{\partial H_1}{\partial x} v \\ \frac{\partial H_2}{\partial x} v \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial b}{\partial x_1}(x_1, x_2)v_1 + \frac{\partial b}{\partial x_2}(x_1, x_2)v_2 \end{bmatrix}. \tag{14}$$

Thus, $(\frac{\partial H_1}{\partial x} v, \frac{\partial H_2}{\partial x} v)$ can be computed by solving the linear system (14). Note that the matrix on the left-hand side is simply the Jacobian of the nonlinear subproblem in Step 1 of SIFP, evaluated at the solution $(H_1(x), H_2(x)) = (x_1^*, x_2^*)$. If Newton's method is used to solve the subproblem, the LU factors of this Jacobian would have already been computed. Thus, they can be reused to solve (14). Also note that the derivatives of the constraints $\frac{\partial b}{\partial x_j}$ appear on both sides of the equation, but they are evaluated using different arguments (at the solution (x_1^*, x_2^*) on the left, and at the inputs (x_1, x_2) on the right). Similarly, we can apply implicit differentiation to Equations (10), (11) to obtain:

$$\begin{bmatrix} \frac{\partial R_2}{\partial x_1}(G_1(x), G_2(x)) & \frac{\partial R_2}{\partial x_2}(G_1(x), G_2(x)) \\ \frac{\partial c}{\partial x_1}(G_1(x), G_2(x)) & \frac{\partial c}{\partial x_2}(G_1(x), G_2(x)) \end{bmatrix} \begin{bmatrix} \frac{\partial G_1}{\partial x} v \\ \frac{\partial G_2}{\partial x} v \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial c}{\partial x_1}(x_1^*, x_2^*)\frac{\partial H_1}{\partial x} v + \frac{\partial c}{\partial x_2}(x_1^*, x_2^*)\frac{\partial H_2}{\partial x} v \end{bmatrix}, \tag{15}$$

where the right hand side contains the previously calculated terms $\frac{\partial H_1}{\partial x} v$ and $\frac{\partial H_2}{\partial x} v$. Again, the left-hand side matrix is simply the Jacobian of the nonlinear subproblem in Step 2, evaluated at the

solution (x_1^{**}, x_2^{**}), so reusing the LU factors is possible. Finally, we can compute the matrix-vector product for an arbitrary vector v via the formula:

$$\frac{\partial \mathcal{F}}{\partial x} v = v - \begin{bmatrix} \frac{\partial G_1}{\partial x} v \\ \frac{\partial G_2}{\partial x} v \end{bmatrix}, \tag{16}$$

where $\frac{\partial G_1}{\partial x} v$ and $\frac{\partial G_2}{\partial x} v$ are calculated by solving (15).

We summarize the full SIN algorithm below. For $k = 0, 1, 2, \ldots$, proceed as follows:

1. Solve
$$R_1(x_1^*, x_2^*) = 0, \qquad b(x_1^*, x_2^*) = b(x_1^k, x_2^k)$$
for (x_1^*, x_2^*). At convergence, store the most recent Jacobian matrix $\begin{bmatrix} \frac{\partial R_1}{\partial x_1} & \frac{\partial R_1}{\partial x_2} \\ \frac{\partial b}{\partial x_1} & \frac{\partial b}{\partial x_2} \end{bmatrix}$, evaluated at (x_1^*, x_2^*), and its LU factors for use in Step 4.
2. Solve
$$R_2(x_1^{**}, x_2^{**}) = 0, \qquad c(x_1^{**}, x_2^{**}) = c(x_1^*, x_2^*)$$
for (x_1^{**}, x_2^{**}). At convergence, store the most recent Jacobian matrix $\begin{bmatrix} \frac{\partial R_2}{\partial x_1} & \frac{\partial R_2}{\partial x_2} \\ \frac{\partial c}{\partial x_1} & \frac{\partial c}{\partial x_2} \end{bmatrix}$, evaluated at (x_1^{**}, x_2^{**}), and its LU factors for use in Step 4.
3. Compute the residual
$$r^k := \mathcal{F}(x^k) = \begin{bmatrix} x_1^k - x_1^{**} \\ x_2^k - x_2^{**} \end{bmatrix}. \tag{17}$$
4. Solve
$$\frac{\partial \mathcal{F}}{\partial x} \Delta x^k = -r^k \tag{18}$$
for Δx^k using a Krylov solver such as GMRES. To multiply $\frac{\partial \mathcal{F}}{\partial x}$ by an arbitrary vector v, first solve (14) for $(\frac{\partial H_1}{\partial x} v, \frac{\partial H_2}{\partial x} v)$, then solve (15) for $(\frac{\partial G_1}{\partial x} v, \frac{\partial G_2}{\partial x} v)$, and finally compute $\frac{\partial \mathcal{F}}{\partial x} v$ via (16).
5. Update the solution by setting
$$x^{k+1} = x^k + \Delta x^k.$$
6. Repeat Steps 1–5 until convergence.

2.3. Computational cost of SIFP vs SIN

We now show a theoretical comparison of the cost of the SIFP and SIN methods, based on their local convergence properties and the computational cost of each step. We first define a number of parameters needed for estimating the run time. In Steps 1 and 2, which are common for both methods, suppose we solve the nonlinear sequential subproblems using Newton's method. This requires a number of inner

Newton iterations per nonlinear solve, which we denote by K_{Newton}. Typically, this number should be somewhere between 3 and 10, but it can be higher for difficult problems. Because the sequential subproblems are similar for the two methods, we will use the same K_{Newton} for both. Within each Newton iteration, we need to calculate the LU factors of the Jacobian matrix. Then, we need to perform forward and backward substitutions to solve the associated systems. Let T_{fact} and T_{sub} be the corresponding run times (where typically T_{fact} is between one and two orders of magnitude larger than T_{sub}, see for instance [40, 41]). In addition, Step 4 of SIN requires the solution of a linear system by GMRES, where each GMRES iteration requires solving (but not factoring) a linear system containing the Jacobian of the sequential subproblems. Assuming that K_{GMRES} such iterations are needed, we deduce that the total run times for SIFP and SIN are given by:

$$T_{\text{SIFP}} = K_{\text{SIFP}} K_{\text{Newton}} (T_{\text{fact}} + T_{\text{sub}}), \tag{19}$$

$$T_{\text{SIN}} = K_{\text{SIN}} (K_{\text{Newton}} (T_{\text{fact}} + T_{\text{sub}}) + K_{\text{GMRES}} T_{\text{sub}}), \tag{20}$$

where K_{SIFP} and K_{SIN} are the number of outer SIFP and SIN iterations required for convergence.

We now compare the various quantities above. First, we compare K_{SIFP} and K_{SIN}. Observe that the SIFP method is a fixed-point method of the type $x^{k+1} = G(x^k)$. If the method is locally convergent near the fixed point $\bar{x} = G(\bar{x})$, then it is well known [42] that the asymptotic convergence rate, which is defined as:

$$\rho := \limsup_{k \to \infty} \|x^k - \bar{x}\|,$$

is given by the spectral radius of the Jacobian matrix of G evaluated at the fixed point. In other words, we have:

$$\rho = \max_j |\lambda_j|,$$

where the λ_j are the eigenvalues of $G^* := \frac{\partial G}{\partial x}(\bar{x})$. Thus, unless all the eigenvalues of the Jacobian are zero (i.e., the matrix is nilpotent), we can only expect the SIFP method to converge linearly in a neighborhood of the fixed point. In contrast, SIN is a Newton method applied to the nonlinear equation $\mathcal{F}(x) = 0$. Since the exact Jacobian is used to calculate the Newton update at every iteration, we expect the method to converge quadratically close to the solution. Thus, we expect $K_{\text{SIFP}} \gg K_{\text{SIN}}$. In practice, our numerical experiments show that K_{SIN} is usually less than 5, which is comparable to (or even slightly lower than) K_{Newton}.

Next, we given an estimate of K_{GMRES}. To do so, we need to consider the properties of the Jacobian matrix $\frac{\partial \mathcal{F}}{\partial x}(\bar{x})$ close to the fixed point. Recall that for a general nonsingular linear system $Au = b$, GMRES finds in k iterations the solution u^k that minimizes the 2-norm of the residual $r^k = b - Au^k$. Equivalently, for a given initial residual $r^0 = b - Au^0$, GMRES chooses the best polynomial $p(z)$ of degree k or lower, such that $p_k(0) = 1$ and $r^k = p_k(A)r^0$ is minimized [43]. Thus, for any other degree

k polynomial q_k with $q_k(0) = 1$, we necessarily have:

$$\|r^k\|_2 = \|p_k(A)r^0\|_2 \leq \|q_k(A)r^0\|_2 \leq \|q_k(A)\|_2\|r^0\|_2.$$

This property allows us to estimate the convergence rate of GMRES by guessing a polynomial $q_k(z)$, knowing that the true residual must be smaller. In the case of SIN, when x^k is close to $\bar{x}$, we have:

$$\frac{\partial \mathcal{F}}{\partial x}(x^k) \approx \frac{\partial \mathcal{F}}{\partial x}(\bar{x}) = I - G^*.$$

Thus, by choosing $q_k(z) = (1-z)^k$, we see that $q_k(0) = 1$ and $q_k(I - G^*) = (G^*)^k$. Thus, we have:

$$\|r^k\|_2 \leq \|q_k(I - G^*)r^0\|_2 = \|(G^*)^k r^0\|_2 \leq \|(G^*)^k\|_2\|r^0|_2 \leq C\rho^k\|r^0\|_2.$$

Thus, GMRES converges at an asymptotic rate that is at least as good as ρ, so we have $K_{\text{GMRES}} \leq K_{\text{SIFP}}$. The difference is usually quite significant: when $I - G^*$ is symmetric positive definite and for a relative tolerance of ϵ, we have

$$K_{\text{GMRES}} \approx (\log|\epsilon|)\sqrt{\kappa(I - G^*)}, \qquad K_{\text{SIFP}} \approx (\log|\epsilon|)\kappa(I - G^*),$$

where $\kappa(\cdot)$ denotes the two-norm condition number of the matrix, see [43]. So for even moderate condition numbers, e.g., $\kappa(I - G^*) \approx 20$, K_{GMRES} will only be a fraction of K_{SIFP}. The situation for non-symmetric matrices is more complicated, but our numerical experiments show that K_{GMRES} is indeed much smaller than K_{SIFP}, see Sections 3 and 4.

Referring back to Equations (19) and (20), we see that the run time of SIN is lower than that of SIFP if:

$$K_{\text{SIN}}K_{\text{GMRES}}T_{\text{sub}} \leq (K_{\text{SIFP}} - K_{\text{SIN}})K_{\text{Newton}}(T_{\text{fact}} + T_{\text{sub}}). \tag{21}$$

Using the fact that $T_{\text{fact}} \gg T_{\text{sub}}$, $K_{\text{SIN}} \approx K_{\text{Newton}}$ and that $K_{\text{GMRES}} \lesssim K_{\text{SIFP}} - K_{\text{SIN}}$ (since $K_{\text{GMRES}} \ll K_{\text{SIFP}}$ and K_{SIN} is usually less than 5–10), we see that (21) holds in the vast majority of cases, meaning that we expect SIN to require less computational cost than SIFP in most cases. Our numerical experiments shown in the next sections confirmed that this is indeed the case for the two problem classes that we consider.

3. Flow-Mechanics Problem

3.1. Governing Equations

In this section, we consider the interaction between the flow of a single component, namely water, and the mechanics of the rock that surrounds it. Assuming constant temperature, the water component can only exist in a single-phase (liquid water). The flow equation takes the form:

$$\frac{\partial}{\partial t}(\phi\rho_w) - \nabla.(\rho_w \boldsymbol{v}_w) - Q_M = 0 \tag{22}$$

where:

- ϕ is the porosity of the rock;
- ρ_w is the mass density of the liquid water;
- $\boldsymbol{v}_w$ is the velocity of the water;
- Q_M is the mass source/sink term.

The density ρ_w of the liquid water is a function of pressure. To model the flow rate of each phase, Darcy's law is used to describe the flow through the porous medium:

$$\boldsymbol{v}_w = -\frac{kk_{rw}}{\mu_w}\nabla(p_w + \rho_l g z) \tag{23}$$

where:

- k is the rock permeability;
- p_l is the pressure of phase l (here we neglect any capillary pressure effects, so the pressure of the phases are equal);
- g is the gravitational constant;
- μ_l is the viscosity of the phase l;
- z is the coordinate direction of gravity.

The porosity ϕ depends on both the pressure and the deformation of the rock, which is described by the mechanics equation below. The quasistatic momentum conservation equation for the aggregate volume (rock skeleton and fluid) is written as:

$$\nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} = 0 \tag{24}$$

where $\rho = \rho_s(1-\phi) + \rho_f \phi$ is the overall mass density, ρ_s is the density of the rock skeleton and ρ_f is the cell-averaged fluid density. The total stress tensor $\boldsymbol{\sigma}$ consists of the both the fluid and rock-skeleton stresses [44]:

$$\boldsymbol{\sigma} = \mathbb{C}\epsilon^e - \boldsymbol{b}p \tag{25}$$

where ϵ^e is the second-order elasticity strain tensor, $\mathbb{C}$ is the fourth-order tensor elasticity moduli tensor, $\boldsymbol{b} = \mathbf{1}b$ is the second-order tensor of Biot coefficients b and p is the fluid pressure.

From 'small deformation' theory and assuming that the total-strain tensor is only composed of elastic contributions (i.e., ignoring plastic and thermal effects):

$$\epsilon = \epsilon^e \tag{26}$$

We can rearrange from Equation (25), (26):

$$\boldsymbol{\sigma} = \mathbb{C}\epsilon - \boldsymbol{b}p \tag{27}$$

The total strain is defined as:

$$\epsilon = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u}\right) \tag{28}$$

The mechanical effect on the flow equation is captured through the porosity's relationship with stress and strain. We follow Coussy [44] to capture the porosity change through the ratio of the volume of connected porous space to the total volume:

$$\phi = \phi_0 + \frac{(b-\phi_0)(1-b)}{K_d}(P - P_0) + b(\epsilon_v - \epsilon_{v,0}) \tag{29}$$

where K_d is the local drained bulk modulus, $\epsilon_v = \mathrm{tr}(\epsilon)$ is the volumetric total strain, ϕ_0, P_0, $\epsilon_{v,0}$ are the reference porosity, pressure, and volumetric strain. Note that the flow equation (22) is coupled to (24) through the porosity ϕ, and the momentum balance equation (24) depends on (22) through the fluid pressure p.

3.2. Fully Coupled Formulation

For the fully coupled solution, we discretize the flow equations (22) fully implicitly in time (backward Euler) and using the finite-volume method in space [45], and the momentum balance equation (24) using P^1 finite elements in space [46]. Choosing the pressure and displacement as primary variables and eliminating all other quantities using Equations (23), (25)–(29), we arrive at the discrete algebraic problem at time step $n+1$, which takes the form

$$\begin{cases} \tilde{R}_F(p_{n+1}, \boldsymbol{u}_{n+1}) := R_F(p_{n+1}, \sigma(p_{n+1}, \boldsymbol{u}_{n+1})) = 0, \\ R_u(p_{n+1}, \boldsymbol{u}_{n+1}) = 0, \end{cases} \tag{30}$$

where

- $p_{n+1} \in \mathbb{R}^{N_c}$ is the vector of pressures at each of the N_c cell centers at time t_{n+1};
- $\boldsymbol{u}_{n+1} \in \mathbb{R}^{N_D N_v}$ is the displacement vector at each of the N_v vertices in the N_D dimensions;
- $R_F : \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \to \mathbb{R}^{N_c}$ is the residual form of the mass conservation equation (22) as a function of the mean stress σ;
- $\sigma : \mathbb{R}^{N_c} \times \mathbb{R}^{N_D N_V} \to \mathbb{R}^{N_c}$ is a function that computes the cell center mean stress based on the pressure and the displacement field;
- $R_u : \mathbb{R}^{N_c} \times \mathbb{R}^{N_D N_v} \to \mathbb{R}^{N_D N_v}$ is the residual form of the momentum balance for the mechanics equations (24).

The fully coupled algorithm consists of applying Newton's method to the system (30) using the solution of the previous time step $(p_n, \boldsymbol{u}_n)$ as the initial guess. More precisely, the iterative process is as follows:

1. Solve for $p_{n+1}^{k+1}, \boldsymbol{u}_{n+1}^{k+1}$ using:

$$\begin{bmatrix} \frac{\partial \tilde{R}_F}{\partial p} & \frac{\partial \tilde{R}_F}{\partial \boldsymbol{u}} \\ \frac{\partial R_u}{\partial p} & \frac{\partial R_u}{\partial \boldsymbol{u}} \end{bmatrix}_{n+1}^{k} \begin{bmatrix} \Delta p^k \\ \Delta \boldsymbol{u}^k \end{bmatrix} = - \begin{bmatrix} \tilde{R}_F(p_{n+1}^k, \sigma(p_{n+1}^k, \boldsymbol{u}_{n+1}^k)) \\ R_u(p_{n+1}^k, \boldsymbol{u}_{n+1}^k) \end{bmatrix} \tag{31}$$

   where $\Delta p^k = p_{n+1}^{k+1} - p_{n+1}^k$, $\Delta \boldsymbol{u}^k = \boldsymbol{u}_{n+1}^{k+1} - \boldsymbol{u}_{n+1}^k$, and the Jacobian matrices $\frac{\partial \tilde{R}_F}{\partial p}$, $\frac{\partial \tilde{R}_F}{\partial \boldsymbol{u}}$, $\frac{\partial R_u}{\partial p}$, $\frac{\partial R_u}{\partial \boldsymbol{u}}$ are all evaluated at $(p_{n+1}^k, \boldsymbol{u}_{n+1}^k)$.

2. Step 1 is repeated until convergence:

$$\left|\left|\tilde{R}_F(p_{n+1}^{k+1}, \sigma(p_{n+1}^{k+1}, \boldsymbol{u}_{n+1}^{k+1}))\right|\right|_\infty \leq \epsilon_p \quad \text{and} \quad \left|\left|R_u(p_{n+1}^{k+1}, \boldsymbol{u}_{n+1}^{k+1})\right|\right|_\infty \leq \epsilon_u. \tag{32}$$

3.3. Fixed Stress Sequential-implicit Formulation

In this section, we demonstrate the effectiveness of the SIN method for the fixed-stress formulation for flow and mechanics. Here, we combine the finite-volume method [45] to discretize the flow equation (Equation 22) and the finite-element method for the momentum balance (Equation 24) [46]. We follow the same framework for the sequential-implicit method as described in [36] for isothermal flow and mechanics problems. We make use of the geomechanics implementation in the AD-GPRS simulation framework [47, 48].

We use the solution of the previous timestep as the initial guess $x_{n+1}^0 = (p_{n+1}^0, \boldsymbol{u}^0) = (p_n, \boldsymbol{u}_n) = x_n$. Where $p^0 \in \mathbb{R}^{N_c}$ is the pressure at the cell centers and $\boldsymbol{u}^0 \in \mathbb{R}^{N_D N_v}$ is the displacement vector at each of the N_v vertices in the N_D dimensions. For simplicity, we will now drop the subscript $n+1$ and all following terms represent the solution at the $n+1$ timestep.

The fixed-stress sequential iterative process is as follows:

1. Compute $\sigma^k = \sigma(p^k, \boldsymbol{u}^k)$
2. Solve for p^* where $R_F(p^*, \sigma^k) = 0$, where convergence is defined as: $\left|\left|R_F(p^*, \sigma^k)\right|\right|_\infty \leq \epsilon_p$
3. Solve for $\boldsymbol{u}^*$ where $R_u(p^*, \boldsymbol{u}^*) = 0$ where convergence is defined as: $||R_u(p^*, \boldsymbol{u}^*)||_\infty \leq \epsilon_u$
4. Update x^{k+1} by fixed-point iteration or Newton's method
5. Repeat steps 1-4 until converged:

$$\left|\left|R_F(p^{k+1}, \sigma(p^{k+1}, \boldsymbol{u}^{k+1}))\right|\right|_\infty \leq \epsilon_p \text{ and } \left|\left|R_u(p^{k+1}, \boldsymbol{u}^{k+1})\right|\right|_\infty \leq \epsilon_u \tag{33}$$

3.4. Sequential-implicit fixed-point Fixed Stress Algorithm

The update for the SIFP fixed-stress algorithm simply uses the solutions from the flow and mechanics residual equations as the update:

$$x^{k+1} = (p^*, \boldsymbol{u}^*) \tag{34}$$

3.5. Sequential-implicit Newton Fixed-Stress Algorithm

To obtain the SIN update, the below nonlinear system is solved using Newton's method:

$$\mathcal{F}(p^k, \boldsymbol{u}^k) = \begin{bmatrix} p^k - p^*(p^k, \boldsymbol{u}^k) \\ \boldsymbol{u}^k - \boldsymbol{u}^*(p^k, \boldsymbol{u}^k) \end{bmatrix} \tag{35}$$

To solve this nonlinear system using Newton's method we need to compute the matrix-vector product for the Jacobian $\frac{\partial \mathcal{F}}{\partial x}$ and an arbitrary vector $v = (p, \boldsymbol{u})$, $p \in \mathbb{R}^{N_c}$, $\boldsymbol{u} \in \mathbb{R}^{N_D N_v}$, $v \in \mathbb{R}^{N_c} \times \mathbb{R}^{N_D N_v}$:

$$\frac{\partial \mathcal{F}}{\partial x} v = \begin{bmatrix} p - w_1 \\ \boldsymbol{u} - w_2 \end{bmatrix} \tag{36}$$

The steps to compute to obtain the update Δx^k given a $x^k = (p^k, \boldsymbol{u}^k)$ are as follows:

1. Compute $\sigma_k = \sigma(p^k, \boldsymbol{u}^k) \in \mathbb{R}^{N_c}$
2. Solve for p^*, from $R_F(p^*, \sigma^k) = 0$, at convergence ($\left|\left|R_F(p^*, \sigma(p^*, \boldsymbol{u}^k))\right|\right|_\infty \leq \epsilon_F$) store:
   - $J_{11} = \frac{\partial R_F}{\partial p}\Big|_{\partial \sigma = 0}(p^*, \sigma^k) \in \mathbb{R}^{N_c \times N_c}$ or store the LU factors for multiplying J_{11}^{-1}, that was used to solve $R_F(p^*, \sigma^k) = 0$.
   - $J_{12} = \frac{\partial R_F}{\partial \sigma}\Big|_{\partial p = 0}(p^*, \sigma^k) \in \mathbb{R}^{N_c \times N_c}$
   - $J_\sigma = \frac{\partial \sigma}{\partial x}(p^*, \sigma^k) \in \mathbb{R}^{N_c \times (N_c + N_D N_v)}$
3. Solve for $\boldsymbol{u}^*$, from $R_u(p^*, \boldsymbol{u}^*) = 0$, at convergence ($||R_u(p^*, \boldsymbol{u}^*)||_\infty \leq \epsilon_u$) store:
   - $J_{21} = \frac{\partial R_u}{\partial p}\Big|_{\partial u = 0}(p^*, \boldsymbol{u}^*) \in \mathbb{R}^{N_D N_v \times N_c}$
   - $J_{22} = \frac{\partial R_u}{\partial u}\Big|_{\partial p = 0}(p^*, \boldsymbol{u}^*) \in \mathbb{R}^{N_D N_v \times N_D N_v}$ or store the LU factors for multiplying J_{22}^{-1}, that was used to solve $R_u(p^*, \boldsymbol{u}^*)$
4. Solve the system of equations:
$$\frac{\partial \mathcal{F}}{\partial x} \Delta x^k = -\mathcal{F} \tag{37}$$
using GMRES and the matrix-vector product defined as:
$$\frac{\partial \mathcal{F}}{\partial x} v = \begin{bmatrix} p - w_1 \\ \boldsymbol{u} - w_2 \end{bmatrix} \tag{38}$$
where:
   - $w_1 = -(J_{11})^{-1}(J_{12} J_\sigma v) \in \mathbb{R}^{N_c}$
   - $w_2 = -(J_{22})^{-1} J_{21} w_1 \in \mathbb{R}^{N_D N_v}$ Here if steps 2 and 3 are solved with a direct solver, we can utilize the same LU factors for the multiplication of J_{11}^{-1} and J_{22}^{-1}
5. Update
$$\begin{bmatrix} p^{k+1} \\ \boldsymbol{u}^{k+1} \end{bmatrix} = \Delta x^k + \begin{bmatrix} p^k \\ \boldsymbol{u}^k \end{bmatrix} \tag{39}$$
6. Repeat steps 1-5 until:
$$\left|\left|R_F(p^{k+1}, \sigma(p^{k+1}, \boldsymbol{u}^{k+1}))\right|\right| \leq \epsilon_F \text{ and } \left|\left|R_u(p^{k+1}, \boldsymbol{u}^{k+1})\right|\right| \leq \epsilon_u \tag{40}$$

3.6. Mandel's Problem

We considered Mandel's consolidation problem [49, 17]. Mandel's problem is a two-dimensional problem with a homogeneous and isotropic poroelastic rock fixed with two impermeable, rigid and frictionless plates on the top and bottom boundaries. The grid has 20 cells in the x and z directions.

Table 1: Rock and fluid properties used for Mandel's problem.

Property	**Value**	**Unit**
Reference porosity	37.5	%
Young's modulus	$\{10^9, 2\times10^8, 10^8\}$	Pa
Biot's constant	1.0	-
Poisson ratio	0.25	-
Undrained Poisson ratio	0.47	-
Permeability,	1	md
Fluid viscosity	9.81×10^{-5}	Pa · s
Fluid compressibility	4.4×10^{-10}	Pa^{-1}
Reference fluid density	1000	$\mathrm{kg/m}^3$

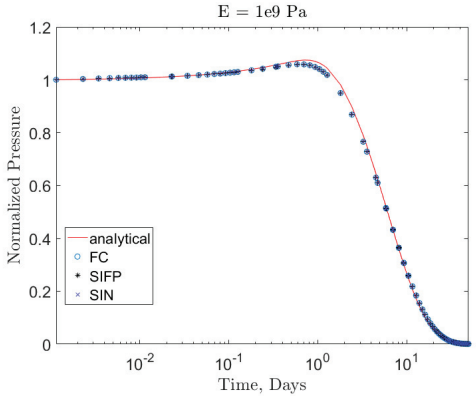
For the SIFP method it has been shown that the coupling strength of the problem [21, 22, 23] is related to the parameter:

$$\tau \equiv \frac{b^2 M}{K_{dr}} \tag{41}$$

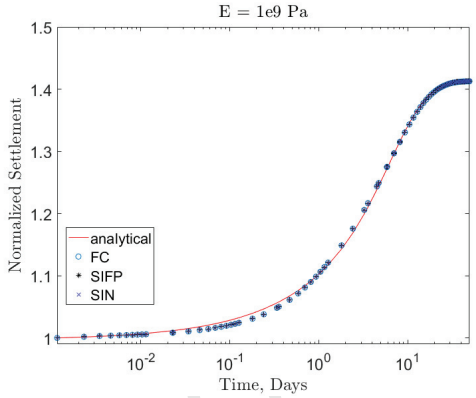
where b is Biot's coefficient, M is the Biot modulus, and K_{dr} is related to the bulk modulus. We can increase the coupling strength by decreasing the Young modulus of the problem [36]. We examined the results for three different Young's moduli, $10^9, 2\times10^8, 10^8$ Pa. A lower Young's modulus results in greater coupling strength, which is expected to require more SIFP outer loop iterations to converge.

The SIN method was implemented and integrated into the Automatic-Differentiation General Purpose Research Simulator (AD-GPRS), which is a general sequential-implicit coupling framework for solving multiphysics problems for reservoir simulation [35]. This framework allows for the consistent testing and development of SIN and SIFP methods. The framework employs a modular code design by splitting each individual physics into a set of different subproblems. The main components of this framework use a subproblem tree structure and abstract computational domains to separate and organize the variable sets for each subproblem. This allows for minimal code duplication and ensures consistent comparisons between formulations and algorithms. In all cases, the convergence tolerance was set to $\epsilon_F = \epsilon_u = 10^{-6}$ and the maximum number of sequential iterations was set to 30. The tolerance for the SIN GMRES solution step was set to 10^{-8}. The linear solver for multiplying by J_{11}^{-1}

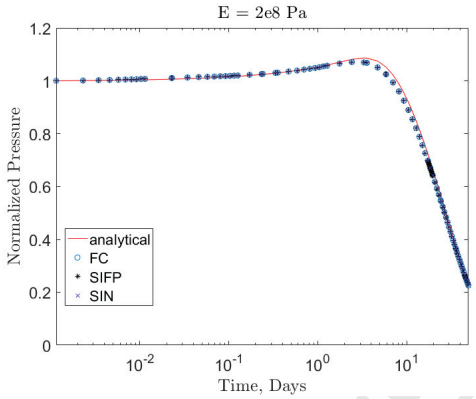
and J_{22}^{-1} at all steps was SuperLU [41]. This choice was made to decouple the effects of nonconvergence of the linear solver on the sequential updates.



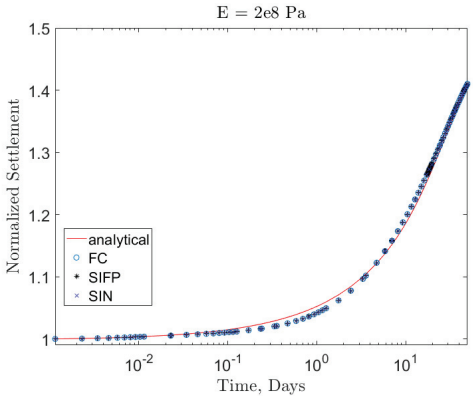
(a) Pressure profile at x = 0 m,y = 100 m ($E = 1 \times 10^9$ Pa)



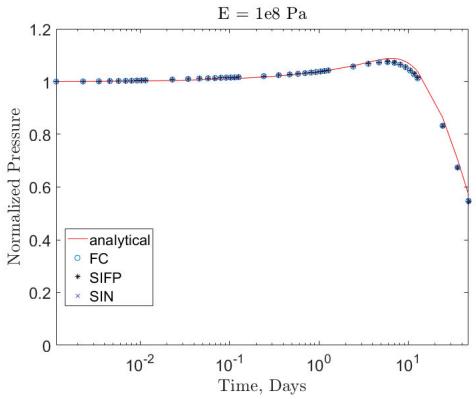
(b) Displacement profile at x = 0 m,y = 100 m ($E = 1 \times 10^9$ Pa)



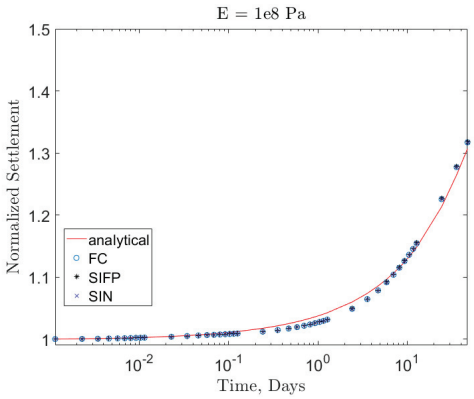
(c) Pressure profile at x = 0 m,y = 100 m ($E = 2 \times 10^8$ Pa)



(d) Displacement profile at x=0 m,y = 100 m ($E = 2 \times 10^8$ Pa)



(e) Pressure profile at x = 0 m,y = 100 m ($E = 1 \times 10^8$ Pa)



(f) Displacement profile at x = 0 m,y = 100 m ($E = 1 \times 10^8$ Pa)

Figure 2: Plot of pressure and displacement profiles for 20×20 grid for Mandel's problem

As shown in Figure 2, the fully coupled, SIFP, and SIN solutions all match the analytical solution. Mandel's problem is very close to a linear problem, with most of the nonlinearity coming from the very small compressibility of the fluid. The full nonlinear results for Mandel's problem are shown in Table 2. The full Newton iterations row represents the number of Newton iterations that the fully coupled problem required for the entire simulation. The inner Newton iterations row is the number of inner loop Newton iterations performed (these Newton iterations are cheaper than the full Newton iterations). The sequential iterations are the number of outer loop iterations to converge to the solution of the full problem. The GMRES iterations is the total number of GMRES iterations to solve Equation (36). The wasted timesteps is the number of timesteps that did not converge either due to an unphysical update, an inner Newton loop not converging, or the maximum number of sequential iterations was reached. The Wasted Full Newtons is the number of full Newton iterations that were computed for those timesteps that were wasted; this is only applicable for the fully coupled method, since the SIFP and SIN only run the smaller inner Newtons. The wasted inner Newtons is the number of Newton iterations computed for the timesteps that were wasted, this only applies for SIFP and SIN. Newton/Seq per timestep is the number of full Newton iterations per time step for FC or the number of sequential outer iterations for the SIFP and SIN methods. Our goal here, and for the remainder of the paper, is not to compare directly the cost of FC versus SIFP/SIN, since the cost per iteration is different for the two classes of methods. Instead, we will use the iteration numbers to understand the convergence behaviour and effectiveness of a particular method for the class of problems in question.

Table 2: Nonlinear results for Mandel's problem (Fully Coupled (FC), Sequential-implicit fixed-point (SIFP), Sequential-implicit Newton (SIN))

	$E = 10^9$ Pa			$E = 2 \times 10^8$ Pa			$E = 10^8$ Pa		
	FC	**SIFP**	**SIN**	**FC**	**SIFP**	**SIN**	**FC**	**SIFP**	**SIN**
Number of Timesteps	90	90	90	92	154	92	93	860	93
Full Newton Iterations	102	-	-	102	-	-	108	-	-
Inner Newton Iterations	-	766	418	-	2519	516	-	10746	549
Sequential Outer Iterations	-	384	226	-	3339	291	-	29569	313
GMRES Iterations	-	-	1958	-	-	3585	-	-	4484
Wasted Timesteps	0	0	0	0	65	0	0	791	0
Wasted Full Newtons	0	-	-	0	-	-	0	-	-
Wasted Inner Newtons	-	0	0	-	2795	0	-	34046	0
Newton/Seq per timestep	1.1	4.3	2.5	1.1	21.7	3.2	1.1	34.3	3.4

Here, we see that even for an almost linear problem, the SIFP problem can struggle due to the

strong coupling between the two subproblems. However, with the SIN method we see that it is able to overcome this strong coupling between the two subproblems. Although the FC outperforms both SIN and SIFP methods; the comparison demonstrates that the SIN method is able to overcome the difficulties faced by SIFP. As expected, as we decrease the Young's modulus, the number of sequential iterations for SIFP increases, and it is convergent only for a very small time step. For the SIN approach, we see a milder trend. We observe only a slight increase in the sequential iterations with coupling strength. As a result of the quadratic convergence of Newton's method, it requires about 2-3 outer sequential iterations per timestep. We see that with SIN, we can get nearly two orders of magnitude less sequential iterations than the SIFP method when the problem is strongly coupled ($E = 10^8$). We see that this slow convergence problem is further intensified by the number of wasted timesteps for the nonconvergent timesteps. The nonconvergent timesteps for the SIFP method were a result of the maximum number of outer sequential iterations (30) being reached.

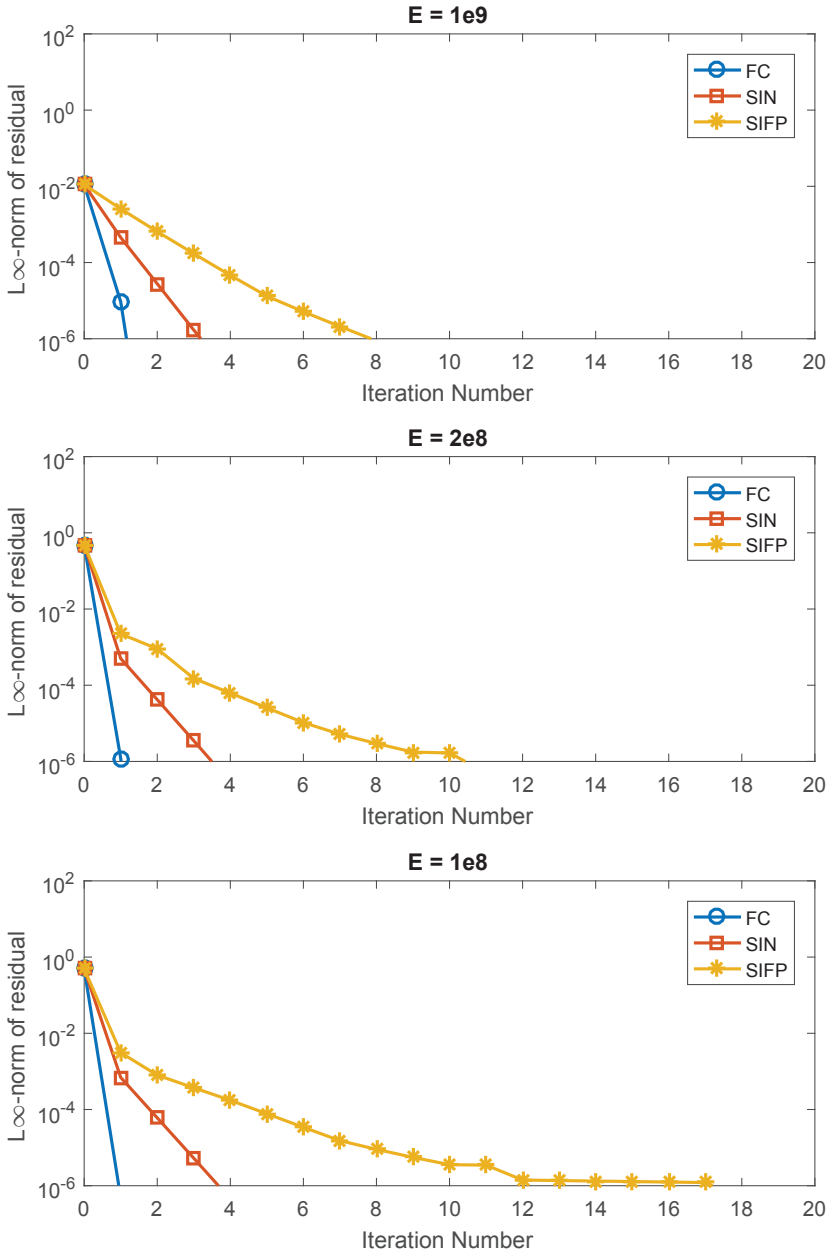


Figure 3: Plot of the L_∞-norm for the flow and mechanics residual after each sequential (SIFP,SIN) or Newton (FC) iteration for different Young's modulus

Figure 3 shows the residual error for sequential iteration for the first time step ($\Delta t = 0.001$ days). We see that as we increase the coupling strength of the problem (increase Young's modulus), as expected the outer loop iterations for the fixed-point iteration increases. However, we see that for both SIN and FC, the convergence profile remains relatively similar.

4. Flow-Thermal Problem

In this section, we demonstrate the effectiveness of the SIN method for the flow and transport of pure water in two-phases with thermal effects. We again consider the flow equation (22) but now the fluid can exist in two different phases liquid water and steam,

$$\frac{\partial}{\partial t}\left(\phi \sum_{l=1}^{2} \rho_l S_l\right) - \nabla.\left(\sum_{l=1}^{2} (\rho_l \boldsymbol{v}_l)\right) - Q_M = 0, \tag{42}$$

where:

- ϕ is the porosity of the rock;
- ρ_l is the mass density of phase l;
- S_l is the saturation of phase l;
- $\boldsymbol{v}_l$ is the velocity of the phase l;
- Q_M is the mass source/sink term.

The subscript l represents the phase of the fluid. The density ρ_l of each phase depends on the phase state of the fluid and is a function of pressure. To model the flow rate of each phase, Darcy's law is used to describe the flow through the porous medium:

$$\boldsymbol{v}_l = -\frac{k k_{rl}}{\mu_l} \nabla(p_l + \rho_l g z) \tag{43}$$

where:

- k is the rock permeability;
- p_l is the pressure of phase l (here we neglect any capillary pressure effects, so the pressure of the phases are equal);
- g is the gravitational constant;
- μ_l is the viscosity of the phase l;
- z is the coordinate direction of gravity.

In addition to these the conservation equations, the saturation constraint must be satisfied; that is, the sum of all the phase saturation is unity:

$$\sum_{l=1}^{2} S_l = 1 \tag{44}$$

The thermal residual equation invovles solving the energy conservation equation:

$$\frac{\partial}{\partial t}\left[(1-\phi)\,\rho_R U_R + \phi \sum_{l=1}^{2} \rho_l U_l S_l\right] - \nabla.\left(\sum_{l=1}^{2}(\rho_l h_l \boldsymbol{v}_l)\right) - \nabla.\,(K\nabla T) - Q_E = 0 \tag{45}$$

where Q_E is the energy source/sink term, h_l is the phase enthalpy of phase l, U_l is the internal energy of phase l, and K is the total conductivity of the fluids and rock. Unlike the isothermal case, the mass density ρ_l and enthalpy h_l of each phase depends on the phase state of the fluid. For single-phase conditions, ρ_l and h_l are functions of pressure and temperature. The thermodynamic relationships of water are taken from Faust and Mercer [50]. However, for two-phase conditions, ρ_l and h_l will depend only on the pressure because now pressure and temperature are dependent: we have $p = p_{sat}(T)$, where p_{sat} is the saturated pressure as a function of temperature. To model the flow rate of each phase, Darcy's law is used to describe the flow through the porous medium (43).

We adapted the geothermal implementation in AD-GPRS [51, 25] for the SIN method. The SIFP for this problem suffers from a large number of outer iterations [25], thus making it a suitable problem to test the effectiveness of SIN.

4.1. Fully Coupled Formulation

For the fully coupled solution, we discretized the mass and energy conservation equations in space using the two-point flux approximation (TPFA) finite-volume method with single-point upstream weighted for the flux discretization. We used a pressure-enthalpy formulation[52]. After discretization, we obtain a discrete algebraic problem of the form:

$$\begin{cases} R_F(p_{n+1}, h_{n+1}) = 0, \\ R_T(p_{n+1}, h_{n+1}) = 0, \end{cases} \tag{46}$$

where

- $p_{n+1}, h_{n+1} \in \mathbb{R}^{N_c}$ is the vector of pressures and enthalpies at each of the N_c cell centers at time t_{n+1};
- $R_F : \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \to \mathbb{R}^{N_c}$ is the residual form of the mass conservation equation, as a function of pressure and enthalpy;[1]
- $R_T : \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \to \mathbb{R}^{N_c}$ is the residual form of the energy balance equations (45).

At the start of each timestep, we use the solution of the previous time step (p_n, h_n) as the initial guess. Thus, fully coupled iterative process for the flow-thermal problem is:

[1]This is different from the R_F for the flow-mechanics problem. However, as we will not talk about both problems simultaneously, no confusion should arise, so we prefer to use the same notation for both flow equations.

1. Solve for $p_{n+1}^{k+1}, h_{n+1}^{k+1}$ for:

$$\begin{bmatrix} \frac{\partial R_F}{\partial p} & \frac{\partial R_F}{\partial h} \\ \frac{\partial R_T}{\partial p} & \frac{\partial R_T}{\partial h} \end{bmatrix}_{n+1}^{k} \begin{bmatrix} \Delta p^k \\ \Delta h^k \end{bmatrix} = \begin{bmatrix} R_F(p_{n+1}^k, h_{n+1}^k) \\ R_T(p_{n+1}^k, h_{n+1}^k) \end{bmatrix} \tag{47}$$

where

- $\Delta p^k = p_{n+1}^{k+1} - p_{n+1}^k$
- $\Delta h^k = h_{n+1}^{k+1} - h_{n+1}^k$
- The Jacobian matrices $\frac{\partial R_F}{\partial p}, \frac{\partial R_F}{\partial h}, \frac{\partial R_T}{\partial p}, \frac{\partial R_T}{\partial h}$ are all evaluated at (p_{n+1}^k, h_{n+1}^k)

2. Step 1 is repeated until convergence:

$$\left|\left|R_F(p_{n+1}^{k+1}, h_{n+1}^{k+1})\right|\right|_\infty \leq \epsilon_p \quad \text{and} \quad \left|\left|R_T(p_{n+1}^{k+1}, h_{n+1}^{k+1})\right|\right|_\infty \leq \epsilon_T. \tag{48}$$

4.2. Sequential Formulation

We apply the general formulation described in Section 2.1 to the flow and thermal problem, and we describe the solution process for solving each of the steps in terms of the specific primary variables and constraints applied for the flow and thermal problem.

At the start of each timestep, we use the solution of the previous timestep as the initial guess $x_{n+1}^0 = (p_{n+1}^0, h_{n+1}^0) = (p_n, h_n) = x_n$. Where $p_{n+1}^0 = p^0 \in \mathbb{R}^{N_c}$ and $h_{n+1}^0 = h^0 \in \mathbb{R}^{N_c}$ are the pressure and enthalpy at the cell centers (N_c is the number of cells). For simplicity, we will now drop the subscript $n+1$ as all following terms represent the solution for the $n+1$ timestep. The sequential iterative process for the flow-thermal problem is:

1. Solve for p^* where $R_F(p^*, h^k)$=0, the constraint for his case is $h^* = h^k$, so we do not need to compute the Jacobian for the b constraint. Convergence is defined as $\left|\left|R_F(p^*, h^k)\right|\right|_\infty \leq \epsilon_F$;
2. Solve for p^{**}, h^{**} where $R_T(p^{**}, h^{**}) = 0$, while satisfying $c(p^*, h^k) = c(p^{**}, h^{**})$, here c is the constraint employed. Convergence is defined as:

$$||R_T(p^{**}, h^{**})||_\infty \leq \epsilon_T \quad \text{and} \quad \left|\left|c(p^*, h^0) - c(p^{**}, h^{**})\right|\right|_\infty \leq \epsilon_T;$$

3. Update x^{k+1} by fixed-point iteration or Newton's method;
4. Repeat steps 1-3 until convergence:

$$\left|\left|R_F(p^{k+1}, h^{k+1})\right|\right|_\infty \leq \epsilon_F \quad \text{and} \quad \left|\left|R_T(p^{k+1}, h^{k+1})\right|\right|_\infty \leq \epsilon_T. \tag{49}$$

4.3. Constraints

In this study, we investigated three different constraints: fixed pressure, fixed density, and a hybrid approach [25]. All three constraints are applied in the second step when the thermal residual is solved.

4.3.1. Fixed Pressure

The fixed pressure scheme assumes that the pressure at each cell is fixed when solving for the thermal residual $c := p$. So $p^{**} = p^*$:

$$R_T(p^{**}, h^{**}) = 0 \tag{50}$$

$$p^{**} - p^* = 0 \tag{51}$$

4.3.2. Fixed Density

The fixed density scheme assumes that the density at each cell is fixed when solving for the thermal residual $c := \rho$. So $\rho^* = \rho(p^*, h^k)$, where ρ is a function that computes the cell center densities based on the pressure and enthalpy of each cell. $(\rho : \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \to \mathbb{R}^{N_c})$

$$R_T(p^{**}, h^{**}) = 0 \tag{52}$$

$$\rho(p^{**}, h^{**}) - \rho(p^*, h^k) = 0 \tag{53}$$

4.3.3. Fixed Pressure and Fixed Density (Hybrid)

The hybrid scheme combines the fixed pressure and fixed density constraints. This constraint varies cell-wise and is based on the phase state of the cell. A fixed pressure is enforced for single-phase cells and fixed density for two-phase cells. We determine the phase state of the cell by comparing the enthalpy with the saturated enthalpy of the cell, $h_w(p)$ and $h_s(p)$ are the saturated water and steam enthalpy respectively. The constraint for each cell i with pressure p_i and and enthalpy h_i:

$$c(p_i, h_i) := \begin{cases} \rho(p_i, h_i) & h_w(p_i) \leq h_i \leq h_s(p_i) \text{ (Two-phase)} \\ p_i & \text{otherwise (Single-phase)} \end{cases} \tag{54}$$

4.4. Sequential-implicit Fixed Point Algorithm

The update for the SIFP fixed stress algorithm is simply using the solutions from the flow and thermal residual equations as the update:

$$x^{k+1} = (p^{**}, h^{**}) \tag{55}$$

4.5. Sequential-implicit Newton Algorithm

The primary operation required to obtain an update for the SIN method is to compute the matrix-vector product for the Jacobian $\frac{\partial \mathcal{F}}{\partial x}$, where $\mathcal{F}$ is the preconditioned system defined in Equation (56). For the flow and thermal problem this preconditioned system is:

$$\mathcal{F}(p^k, h^k) = \begin{bmatrix} p^k - p^{**}(p^k, h^k) \\ h^k - h^{**}(p^k, h^k) \end{bmatrix} \tag{56}$$

The steps to compute the update Δx^k for a given iteration p^k and h^k are as follows:

1. Calculate p^*, by solving $R_F(p^*, h^k) = 0$ using Newton's method, at convergence ($\left|\left|R_F(p^*, h^k)\right|\right|_\infty \leq \epsilon_F$) store:
   - $J_{11} = \frac{\partial R_F}{\partial p}\Big|_{\partial h=0}(p^*, h^k) \in \mathbb{R}^{N_c \times N_c}$
   - $J_{12} = \frac{\partial R_F}{\partial h}\big|_{\partial p=0}(p^*, h^k) \in \mathbb{R}^{N_c \times N_c}$
   - $c_p^0 = \frac{\partial c}{\partial p}\Big|_{\partial h=0}(p^*, h^k) \in \mathbb{R}^{N_c \times N_c}$
   - $c_h^0 = \frac{\partial c}{\partial h}\big|_{\partial p=0}(p^*, h^k) \in \mathbb{R}^{N_c \times N_c}$
2. Calculate p^{**}, h^{**}, by solving $R_T(p^{**}, h^{**}) = 0, c(p^*, h^0) - c(p^{**}, h^{**}) = 0$, at convergence ($||R_T(p^{**}, h^{**})||_\infty \leq \epsilon_T$ and $||c(p^{**}, h^{**}) - c(p^*, h^*)||_\infty \leq \epsilon_T$) store:
   - $J_{21} = \frac{\partial R_T}{\partial p}\Big|_{\partial h=0}(p^{**}, h^{**}) \in \mathbb{R}^{N_c \times N_c}$
   - $J_{22} = \frac{\partial R_T}{\partial h}\big|_{\partial p=0}(p^{**}, h^{**}) \in \mathbb{R}^{N_c \times N_c}$ or store the LU factors used to multiply by $\left(J_{22} - J_{21}(c_p^1)^{-1}c_h^1\right)^{-1}$ that is computed when solving $R_T(p^{**}, h^{**}) = 0, c(p^*, h^0) - c(p^{**}, h^{**}) = 0$
   - $c_p^1 = \frac{\partial c}{\partial p}\Big|_{\partial h=0}(p^{**}, h^{**}) \in \mathbb{R}^{N_c \times N_c}$
   - $c_h^1 = \frac{\partial c}{\partial h}\big|_{\partial p=0}(p^{**}, h^{**}) \in \mathbb{R}^{N_c \times N_c}$
3. Calculate:
$$r^k := \mathcal{F}(x^k) = \begin{bmatrix} p^k - p^{**} \\ h^k - h^{**} \end{bmatrix} \in \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \tag{57}$$
4. Solve
$$\frac{\partial \mathcal{F}}{\partial x}\Delta x^k = -r^k \tag{58}$$
for Δx^k using GMRES. This will require multiplying $\frac{\partial \mathcal{F}}{\partial x}$ by an arbitrary vector $v = (v_1, v_2)$, where:
$$\frac{\partial \mathcal{F}}{\partial x}v = v - \begin{bmatrix} \frac{\partial G_1}{\partial x} \\ \frac{\partial G_2}{\partial x} \end{bmatrix} v = \begin{bmatrix} v_1 - w_1 \\ v_2 - w_2 \end{bmatrix} \tag{59}$$
   (a) Compute $z_1 = -J_{11}\left(J_{12}v_2\right) \in \mathbb{R}^{N_c}$
   (b) Compute
$$w_2 = \left(J_{22} - J_{21}\left(c_p^1\right)^{-1}c_h^1\right)^{-1}\left(-J_{21}(c_p^1)^{-1}\left[c_p^0 z_1 + c_h^0 v_2\right]\right) \in \mathbb{R}^{N_c} \tag{60}$$
   Here we compute the Schur complement of the large block matrix system in Equation 15. The two inverses $\left(c_p^1\right)^{-1}$ and $\left(c_p^1\right)^{-1}$ are cheap to calculate since they are both diagonal matrices. The matrix multiplication by $\left(J_{22} - J_{21}(c_p^1)^{-1}c_h^1\right)^{-1}$ could also utilize the LU factors at step 2 if a direct solver was used.
   (c) Compute
$$w_1 = \left(c_p^1\right)^{-1}\left(c_p^0 z_1 + c_h^0 h - c_h^1 w_2\right) \tag{61}$$

5. Update

$$\begin{bmatrix} p^{k+1} \\ h^{k+1} \end{bmatrix} = \Delta x^k + \begin{bmatrix} p^k \\ h^k \end{bmatrix} \tag{62}$$

6. Repeat steps 1-5 until

$$\left|\left|[R_F(p^{k+1}, h^{k+1})\right|\right|_\infty \leq \epsilon_F \text{ and } \left|\left|[R_T(p^{k+1}, h^{k+1})\right|\right|_\infty \leq \epsilon_T \tag{63}$$

4.5.1. One-dimensional radial model example

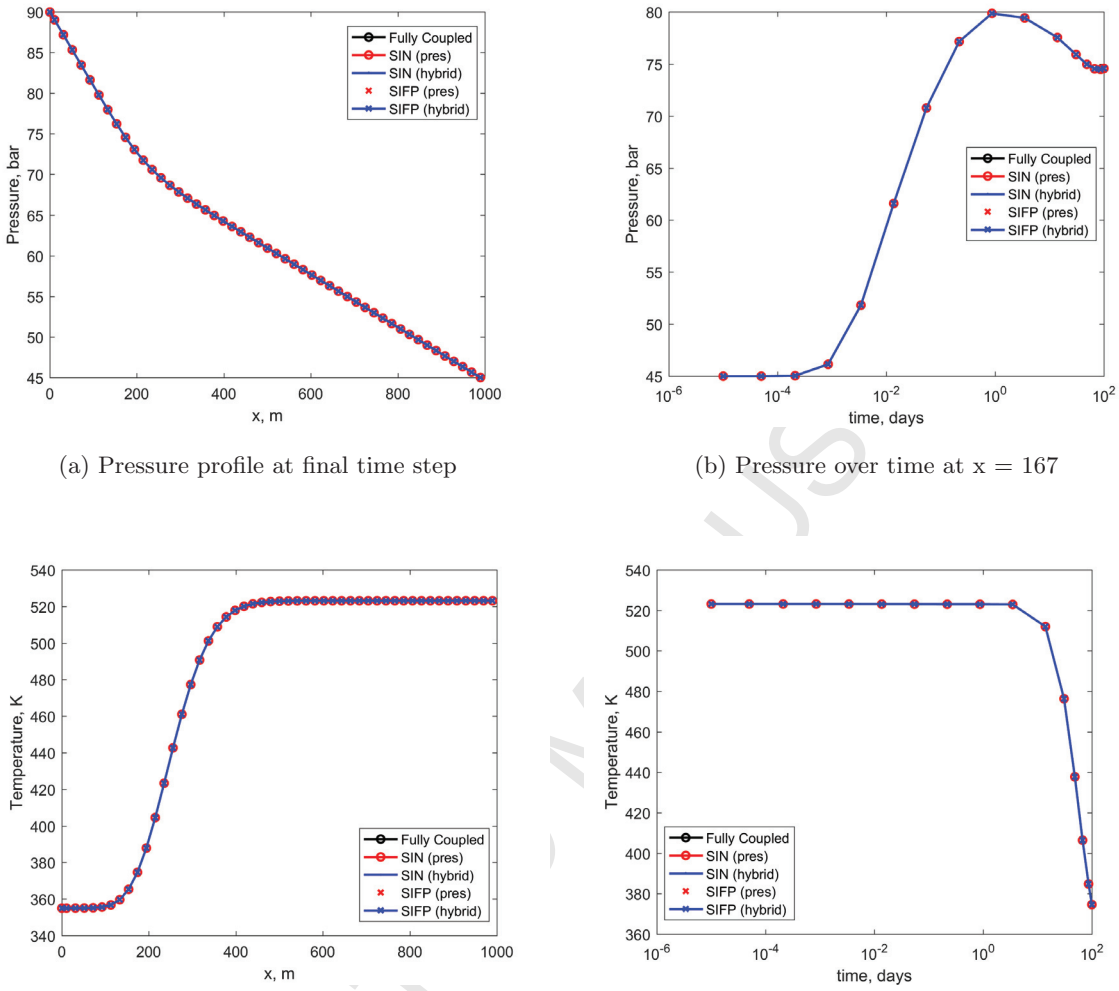
This numerical example is a one-dimensional radial model. Here we consider single- and two-phase flow regimes. The single-phase case involves injection of cold water into a hot water reservoir. For the two-phase case, cold water is injected into a two-phase reservoir at saturated conditions. This model is based on the one-dimensional model investigated in the Stanford Code Comparison study [53].

The rock parameters for this model are:

Table 3: Rock parameters

Property	**Value**	**Unit**
Reservoir Length	1000	m
Reservoir Thickness	100	m
Permeability	100	md
Rock density	2500	kg/m3
Rock specific heat capacity	1.0	J/(gK)
Porosity	0.2	%

In this study we looked at a 50 cell one-dimensional radial model. Here we use $\epsilon_F = \epsilon_T = 10^{-4}$. The tolerance for the SIN GMRES solution step was set to 10^{-8}. The linear solver used to multiply $\left(J_{22} - J_{21}(c_p^1)^{-1}c_h^1\right)^{-1}$ was SuperLU [41], this was to decouple any effects the linear solver would have on the sequential updates.



(a) Pressure profile at final time step

(b) Pressure over time at x = 167

(c) Temperature profile at final time step

(d) Temperature over time at x = 167

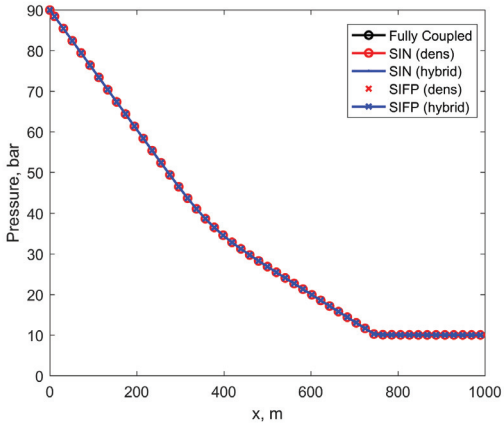
Figure 4: The pressure and temperature profiles for 50 cell single-phase injection

Figure 4 shows that the results of both SIN and SIFP are in close agreement with the fully coupled method. This is consistent with the results shown by [32] that they should both converge to the same solution.

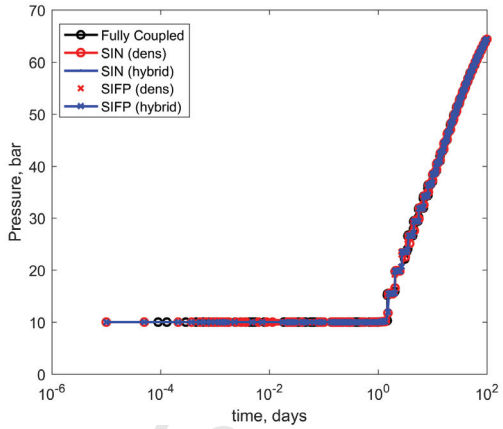
For the single-phase model, we omitted the results of the fixed-density approach for SIFP and SIN as this was divergent. This is consistent with the results shown by Wong et al. [25] where the fixed-density method has poor convergence for single-phase cells. The fixed pressure and hybrid approach have identical performance for both the SIFP and SIN methods. The SIN method results in about three times fewer sequential iterations than SIFP. We also notice that the SIN method has about the same number of sequential iterations to the Newton iterations of the fully coupled method.

Table 4: Nonlinear results for 50 cell one-dimensional radial model for single-phase injection

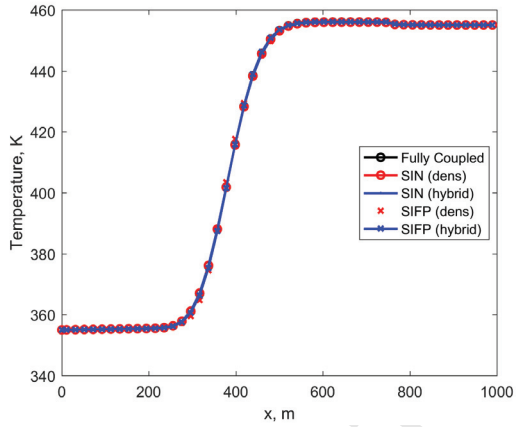
	SIFP		**SIN**		
	Pressure	**Hybrid**	**Pressure**	**Hybrid**	**Fully Coupled**
Number of timesteps	16	16	16	16	16
Full Newton Iterations	-	-	-	-	36
Inner Newton Iterations	282	282	87	87	-
GMRES Iterations	-	-	319	319	-
Sequential Outer Iterations	109	109	35	35	-
Wasted Timesteps	0	0	0	0	0
Newton/Seq per timestep	6.8	6.8	2.2	2.2	2.3



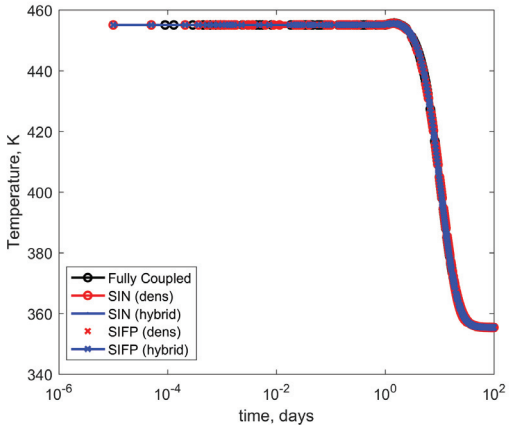
(a) Pressure profile at final timestep



(b) Pressure over time at $x = 167$



(c) Temperature profile at final time step



(d) Temperature over time at x = 167

Figure 5: The pressure and temperature profiles for 50 cell two-phase injection

The focus of this test case was to investigate the convergence behavior of the different methods with two-phase effects. Due to the phase transitions, we see that this results in a nonmonotonic pressure solution over time at the position $x = 167$m (Figure 5b). The nonlinear behavior for this type of problem is described fully by Wong et al. [54]. We see that the solutions for the different methods are consistent up to the time stepping schemes for which they are convergent for. The results for the fixed pressure scheme was omitted as it is divergent for two-phase problems. The small differences between the approaches are attributed to the different convergent time-stepping schemes.

Table 5: Nonlinear results for two-phase 50 cell one-dimensional radial two-phase model

	SIFP		SIN		
	Density	**Hybrid**	**Density**	**Hybrid**	**Fully Coupled**
Number of timesteps	1803	442	234	243	304
Full Newton Iterations	28544	8349	2035	1750	1159
Inner Newton Iterations	28544	8349	2035	1750	1159
GMRES Iterations	-	-	20382	8730	-
Sequential Iterations	49374	8571	811	832	-
Wasted Timesteps	1794	432	223	232	293
Wasted Full Newtons	-	-	-	-	686
Wasted Inner Newtons	76597	14039	2281	2670	-
Newton/Seq per timestep	27.4	19.4	3.5	3.4	3.8

For the 50-cell two-phase model, we see that the SIN method outperforms SIFP for both the fixed-density and hybrid approaches. Again, we see that the fixed-density SIFP has very poor nonlinear convergence, requiring about 4.1 times more timesteps than the hybrid SIFP method and 7.7 times more timesteps than fixed-density SIN. Comparing SIFP and SIN, the total number of sequential iterations has decreased by a factor of 10 for the hybrid and a factor of 61 for the fixed density approach. It is interesting to note that although the focus of this study was to show how the SIN outperforms the SIFP method, we see that the fully coupled approach struggles with this problem with about 1.3 times more time steps than SIN method for both coupling strategies. The issues that the FC faces for this problem are well described in [54]. The wasted time steps for the fully coupled method and SIN were due to a unphysical updates to the solution. The wasted timesteps for the SIFP method were a combination of both unphysical updates and reaching the maximum number of sequential iterations.

To further investigate the wasted full and inner Newton iterations, a plot of the number of Newton iterations computed for each time step is shown (Figure 6). The full Newtons for the FC and the inner Newtons for SIFP and SIN are not comparable, as the full Newtons are solving for two times more residual equations with the two physics. This makes each Newton step for FC considerably more expensive than a single inner Newton step. However, it is useful to see qualitatively how the FC and SIN methods compare. One clear trend is that the SIN requires fewer Newton iterations than the SIFP method and at the later time, we see that the SIFP is consistently wasting about 50 inner Newton iterations. This is due to the maximum number of sequential iterations being reached by the SIFP method. This is issue is not faced by SIN due to the faster outer loop sequential convergence.

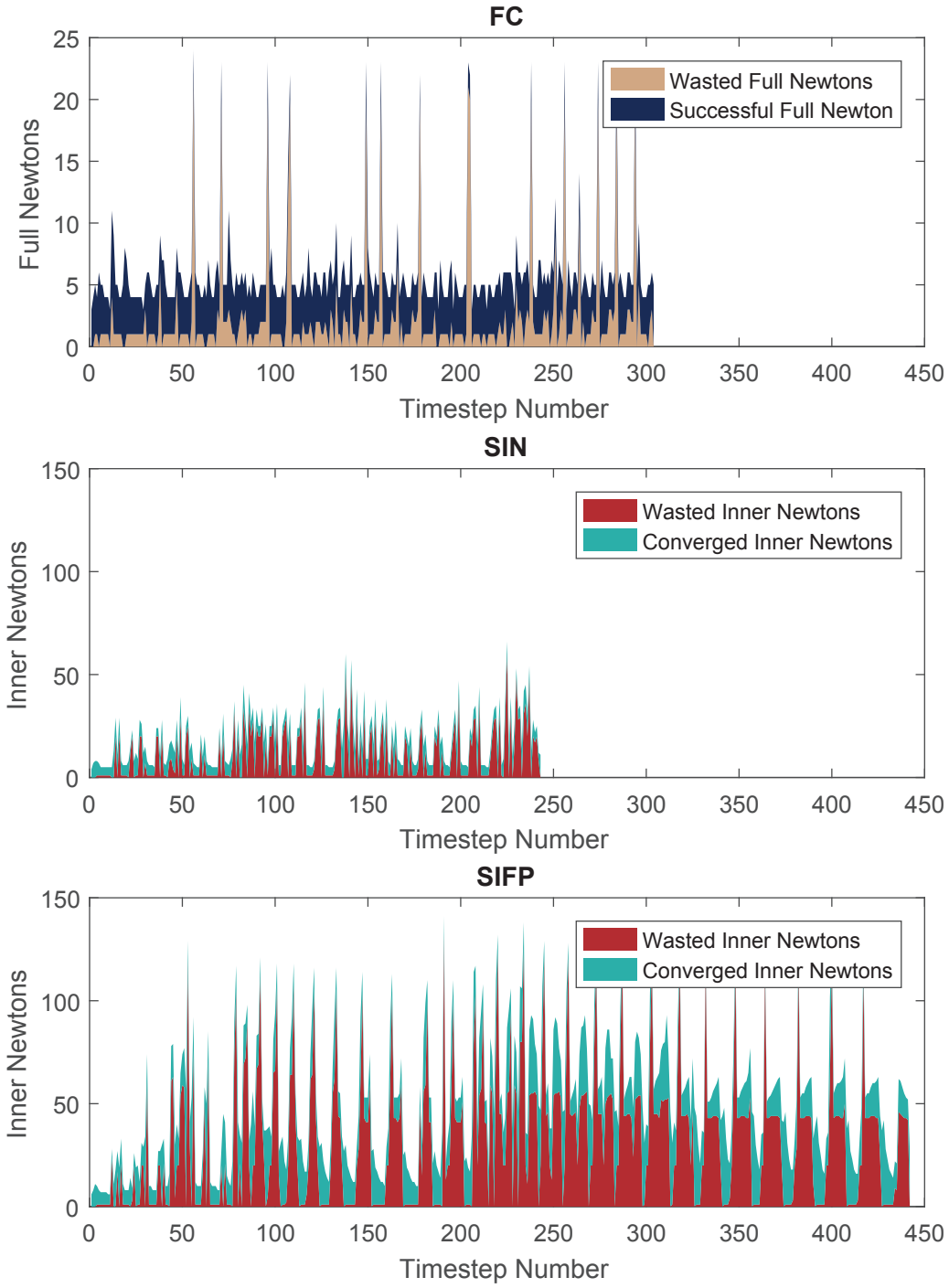


Figure 6: Plot of Newton iterations for each time step for the different schemes for the two-dimensional two-phase problem, here we compare the SIN, SEQ hybrid cases

From these two different one-dimensional problems, we see qualitatively a few convincing trends comparing the different approaches. Across the two flow regimes, the SIN method consistently reduced the number of sequential iterations in comparison to SIFP. The number of sequential iterations for SIN is now comparable and can be smaller than the fully coupled method. This shows qualitatively the quadratic convergence rate of the SIN outer loop updates. We also notice that when comparing different constraints for SIN, we see the same qualitative trend as SIFP, that the hybrid approach is superior to the fixed density and fixed pressure as the fixed density fails for the single-phase case and the fixed pressure fails for the two-phase case. This underscores the statement mentioned in Liu and Keyes[32] that the partition of the physical variables (and constraints) is the most interesting part of the implementation.

4.5.2. Two-dimensional Heterogeneous Example

The permeability and porosity distribution for this example was taken from a section of the top layer of the SPE10 model, the permeability and porosity can be seen in Figure 7. There are $10 \times 10 \times 1$ grid cells, each with size $20 \times 10 \times 70$m. The purpose of this test was to investigate how the different sequential strategies perform with a heterogeneous permeability field for single- and two-phase flow regimes. A constant pressure and temperature condition were specified for the cell on the top left and bottom right cells. The top left cell was set at 90 bar and 350 K. The bottom right cell was set at 60 bar and 350 K for single-phase conditions and 10 bar, 0.2 water saturation, 350 K at two-phase conditions. Both flow problems we simulated for 100 days. Figure 8a and 8b show the solutions for the single-phase and two-phase solutions at two simulation times. The relative difference between the SIN and SIFP solutions can be seen in Figure 9. For this study we set the maximum number of sequential iterations to be 30 and $\epsilon_F = \epsilon_T = 10^{-4}$. Similar to all the other numerical cases, SuperLU [41] was used as the linear solver for all the linear equations solved. The tolerance for the GMRES algorithm when solving 58 was set to 10^{-8}.

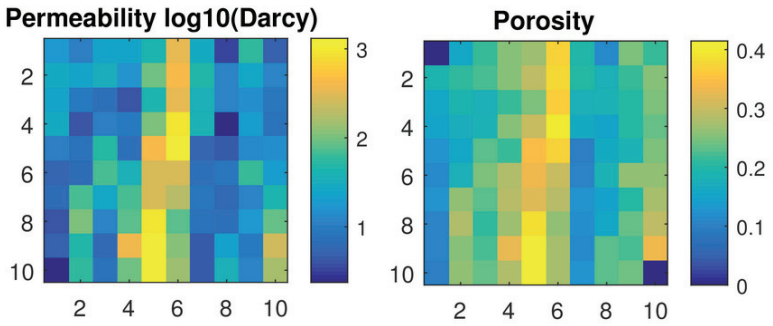
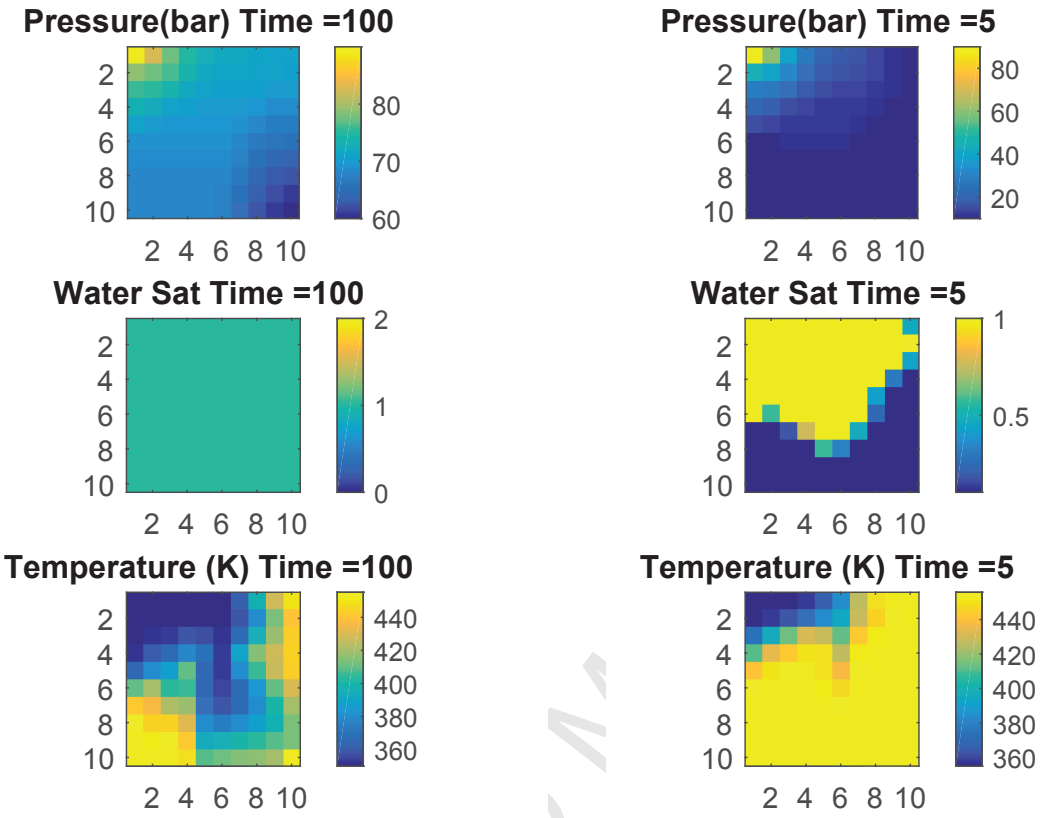


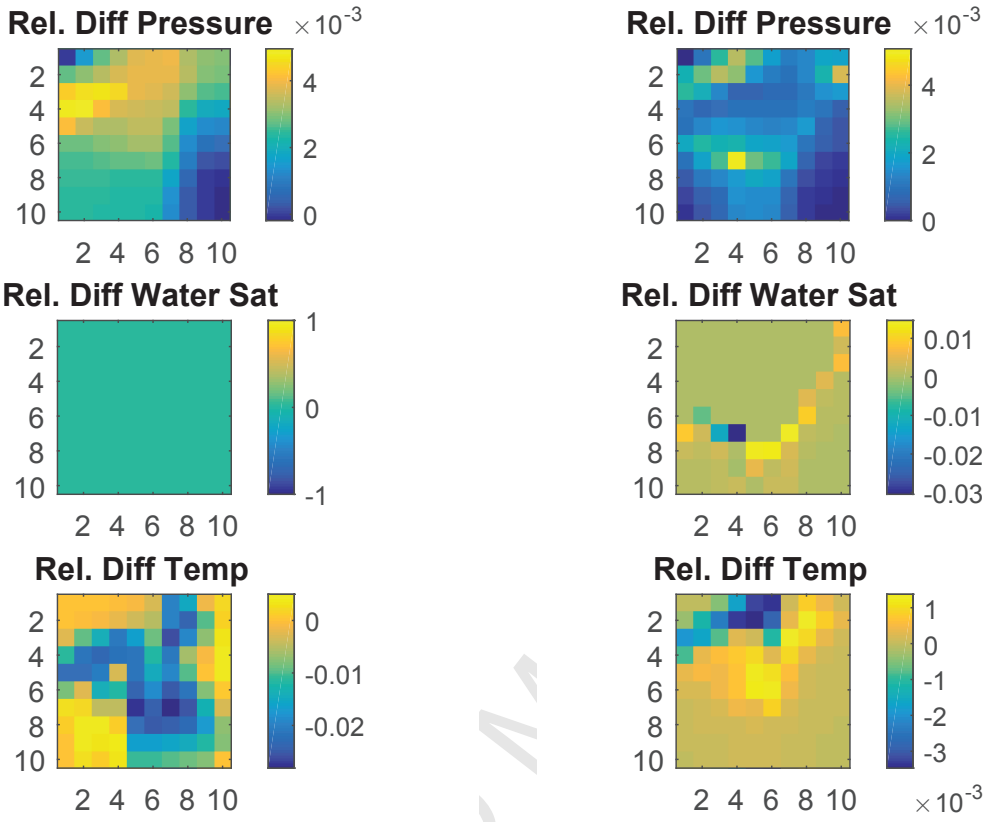
Figure 7: Input Permeability (Left) and Porosity (Right), Dark blue cells in porosity indicate the boundary cells



(a) Two-dimensional single-phase

(b) Two-dimensional two-phase

Figure 8: Solution for the heterogeneous single- and two-phase models



(a) Two-dimensional single-phase

(b) Two-dimensional two-phase

Figure 9: Relative difference of SIFP and SIN solutions $\left(\frac{x_{SIN}-x_{SIFP}}{x_{SIFP}}\right)$ for the heterogeneous single- and two-phase models

From the one-dimensional problem, we demonstrated how the hybrid approach was superior to the other constraints for all flow regimes. Thus, we only investigated the hybrid approach for SIFP and SIN. Table 6 shows the nonlinear performance of the fully coupled method, hybrid SIFP, and the hybrid SIN method. For the single-phase problem, we see that the number of sequential iterations for SIFP is 7.2 times more than SIN. We also notice that there were 6 wasted iterations. However, for SIN, we see that the number of sequential outer loop iterations (36) is identical to the fully coupled Newton iterations.

For the two-phase problem, where cold water is invading a two-phase reservoir, we again see the effectiveness of SIN. We see that SIFP requires 27 times more outer loop sequential iterations than the SIN method. Similar to the one-dimensional problem, because the fully coupled approach struggles with this type of physics problem, we see that SIN has better nonlinear convergence than the fully

coupled method and requires fewer timesteps. Again, we see a similar number of sequential iterations per timestep for SIN and the fully coupled method. This is larger than the other problems examined as for this particular flow regime, there is a stronger nonlinearity from the phase change and two-phase flow.

Table 6: Nonlinear results for heterogeneous two-dimensional model

	Single-phase			**Two-phase**		
	FC	**SIFP**	**SIN**	**FC**	**SIFP**	**SIN**
Number of Timesteps	20	25	20	129	373	96
Full Newton Iterations	36	-	-	607	-	-
Inner Newton Iterations	-	295	84	-	9531	1089
GMRES Iterations	-	-	325	-	-	6565
Sequential Iterations	-	259	36	-	10421	381
Wasted Timesteps	0	6	0	109	353	76
Wasted Full Newtons	0	-	-	555	-	-
Wasted Inner Newtons	-	282	0	-	14369	1550
Newton/Seq per timestep	1.8	10.4	1.8	4.7	27.9	4.0

To compare the Newton iterations for the three different methods of the two-phase case, we again plot the number of Newton iterations for the FC, SIN, and SIFP method in Figure 10. Here, we see even more clearly that SIFP has a large number of timesteps with 50 wasted Newton iterations. This is again due to the maximum number of sequential iterations being reached. The time steps wasted for the SIN method is because of unphysical updates rather than the maximum number of sequential iterations being reached. Thus, we see that because SIN is able to overcome the outer loop convergence issues faced by SIFP, it is able to converge to a timestepping scheme that requires less timesteps than FC. From this Figure 10 we see clearly how the poor sequential convergence of the SIFP method can affect the time steps of the problem.

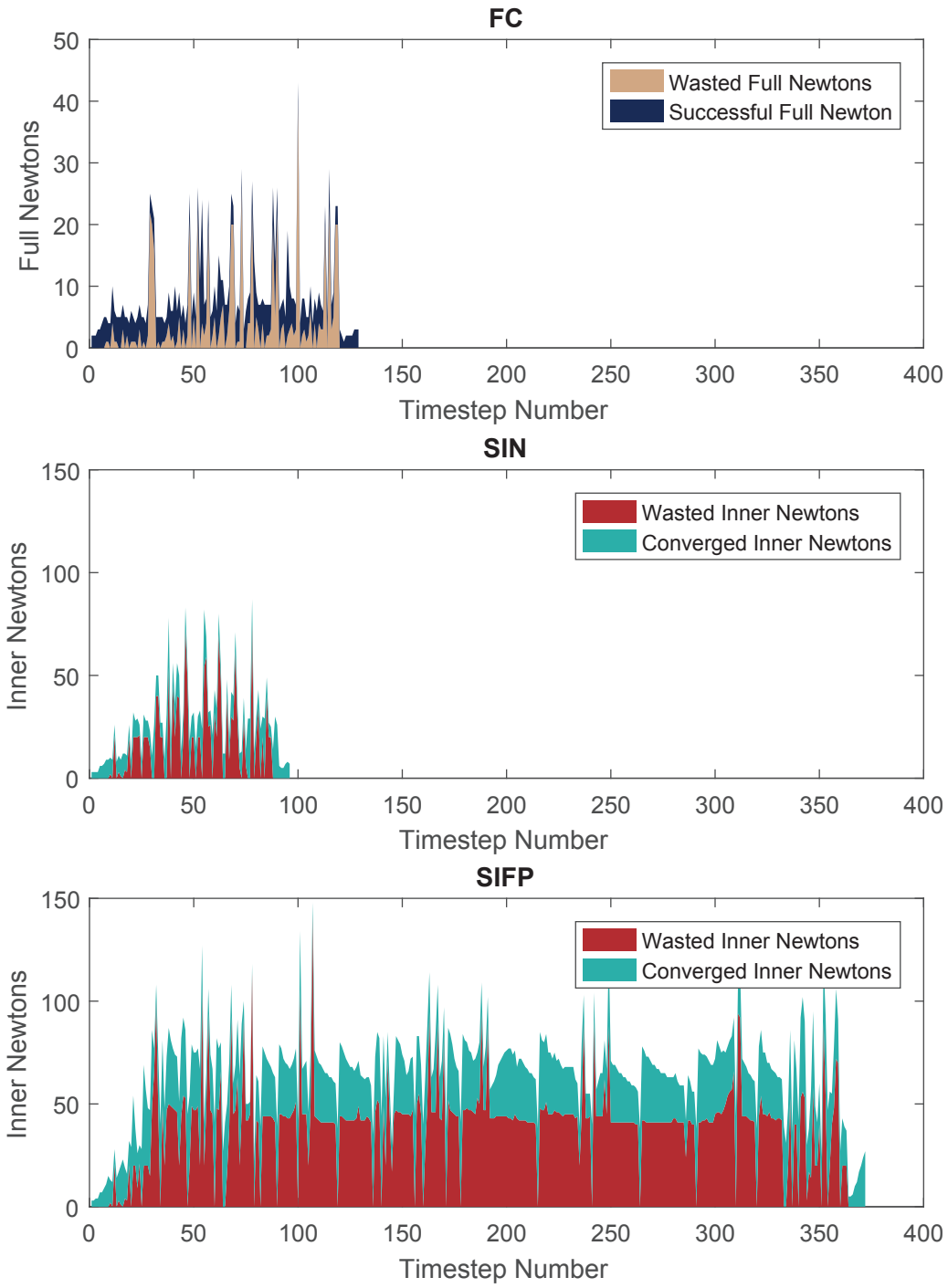


Figure 10: Plot of Newton iterations for each time step for the different schemes for the two-dimensional two-phase problem

Increasing the complexity of the problem with heterogeneity and two-dimensional flow further improves the speed up of SIN compared to SIFP. As shown similarly by Wong et al. [25], the SIFP with a hybrid approach has been shown to struggle as more complexities such as heterogeneity are added. However, for SIN, it is able to have a comparable number of sequential iterations with the Newton iterations for a fully coupled method. This is due to the quadratic convergence rate of which Newton's method is able to take advantage. Here the six wasted time steps for the SIFP single-phase problem were due to the maximum number of sequential iterations reached. For the FC and SIN two-phase cases, the wasted time steps were due to unphysical updates in the inner loop solutions.

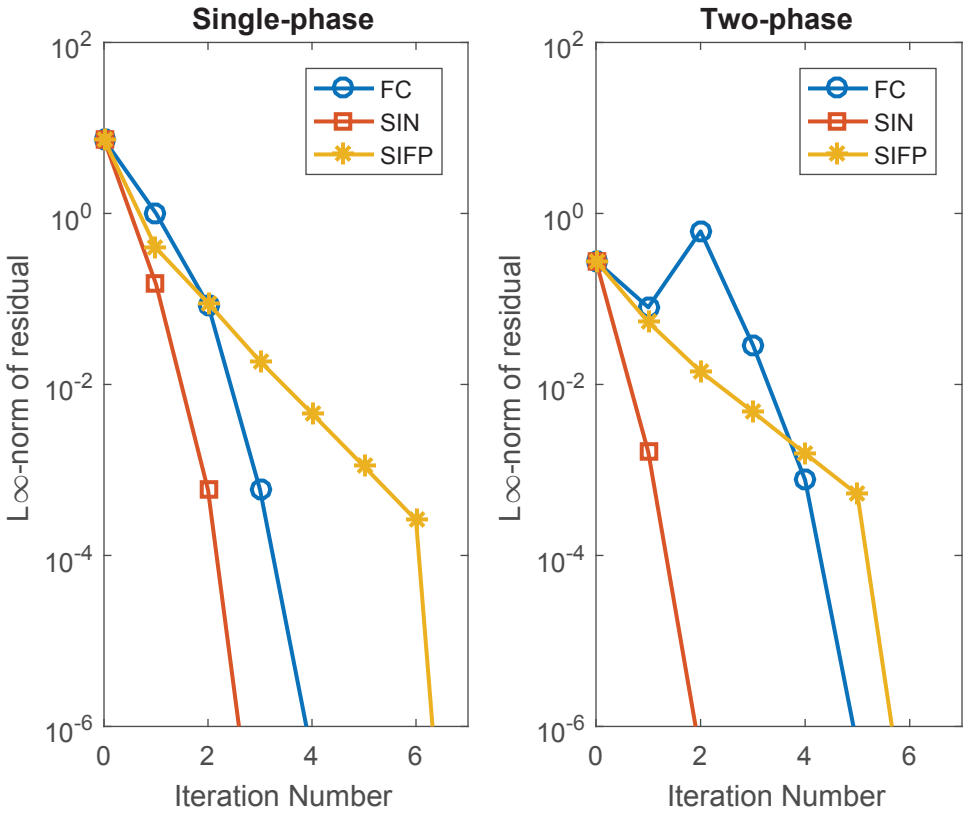


Figure 11: Plot of L_∞-norm of the flow and thermal residual after each sequential (SIFP/SIN) or Newton (FC) iteration

Figure 11 shows a plot of the L_∞-norm of the residual after each sequential or Newton iteration. These residuals were taken at $T = 10^{-2}$ days. We see that for both the single- and two-phase problems, we have the SIN method converging faster than SIFP. For the single-phase problem, we see that the fully coupled and SIN method have similar convergence profiles, with the SIN method converging in one less iteration. For the two-phase problem, we notice that the fully coupled method took only one less iteration that SIFP; this is due to the difficulty for the fully coupled method for this type of cold

water injection problem [54].

5. Conclusions

Sequential-implicit simulations currently employ a fixed-point iteration. This fixed-point approach has only a linear convergence rate and can be problematic for multiphysics problems with strong coupling between the subproblems that are split. A sequential-implicit Newton (SIN) method was presented and used to improve this convergence rate to a quadratic convergence rate.

This sequential-implicit Newton method was demonstrated for two different multiphysics problems: flow-thermal and a flow-mechanics problem. For all the test cases investigated, SIN outperformed the sequential-implicit fixed-point iteration approach in terms of the outer loop convergence. For problems that have more complexities, or stronger coupling between the subproblems, SIN can have an order of magnitude decrease in the number of sequential iterations. The sequential-implicit Newton convergence rate is also shown to be qualitatively similar to a fully coupled approach, where Newton's method is applied to the entire multiphysics problem.

Acknowledgements

The authors thank the Consortium on Reservoir Simulation Research at Stanford University (SUPRI-B) for supporting this research.

References

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

[2] J. Xu, J. Zou, Some nonoverlapping domain decomposition methods, SIAM review 40 (4) (1998) 857–914.

[3] H. Berninger, R. Kornhuber, O. Sander, A multidomain discretization of the richards equation in layered soil, Computational Geosciences 19 (1) (2015) 213–232.

[4] H. Berninger, S. Loisel, O. Sander, The 2-Lagrange multiplier method applied to nonlinear transmission problems for the richards equation in heterogeneous soil with cross points, SIAM Journal on Scientific Computing 36 (5) (2014) A2166–A2198.

[5] X.-C. Cai, D. E. Keyes, Nonlinearly preconditioned inexact Newton algorithms, SIAM Journal on Scientific Computing 24 (1) (2002) 183–200.

[6] L. Marcinkowski, X.-C. Cai, Parallel performance of some two-level ASPIN algorithms, in: Domain Decomposition Methods in Science and Engineering, Springer, 2005, pp. 639–646.

[7] F.-N. Hwang, X.-C. Cai, A class of parallel two-level nonlinear Schwarz preconditioned inexact newton algorithms, Computer methods in applied mechanics and engineering 196 (8) (2007) 1603–1611.

[8] P. R. Brune, M. G. Knepley, B. F. Smith, X. Tu, Composing scalable nonlinear algebraic solvers, SIAM Review 57 (4) (2015) 535–565.

[9] P. Jenny, S. H. Lee, H. A. Tchelepi, Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media, Journal of Computational Physics 217 (2) (2006) 627–641.

[10] A. Moncorgé, H. A. Tchelepi, P. Jenny, Modified sequential fully implicit scheme for compositional flow simulation, Journal of Computational Physics 337 (2017) 98–115.

[11] S. H. Lee, C. Wolfsteiner, H. A. Tchelepi, Multiscale finite-volume formulation for multiphase flow in porous media: black oil formulation of compressible, three-phase flow with gravity, Computational Geosciences 12 (3) (2008) 351–366.

[12] O. Møyner, K. A. Lie, A multiscale restriction-smoothed basis method for compressible black-oil models, SPE Journal 21 (06) (2016) 2–079.

[13] O. Møyner, H. A. Tchelepi, A multiscale restriction-smoothed basis method for compositional models, in: SPE Reservoir Simulation Conference, Society of Petroleum Engineers, 2017.

[14] A. Moncorgé, H. A. Tchelepi, P. Jenny, Sequential fully implicit formulation for compositional simulation using natural variables, Journal of Computational Physics 371 (2018) 690–711.

[15] J. A. White, R. I. Borja, Block-preconditioned newton–krylov solvers for fully coupled flow and geomechanics, Computational Geosciences 15 (4) (2011) 647.

[16] J. A. White, N. Castelletto, H. A. Tchelepi, Block-partitioned solvers for coupled poromechanics: A unified framework, Computer Methods in Applied Mechanics and Engineering 303 (2016) 55–74.

[17] N. Castelletto, J. White, H. A. Tchelepi, Accuracy and convergence properties of the fixed-stress iterative solution of two-way coupled poromechanics, International Journal for Numerical and Analytical Methods in Geomechanics 39 (14) (2015) 1593–1618.

[18] S. Klevtsov, N. Castelletto, J. A. White, H. A. Tchelepi, Block-preconditioned krylov methods for coupled multiphase reservoir flow and geomechanics, in: ECMOR XV-15th European Conference on the Mathematics of Oil Recovery, 2016.

[19] K. Pruess, C. Oldenburg, G. Moridis, TOUGH2 user's guide version 2, Lawrence Berkeley National Laboratory (1999).

[20] J. Rutqvist, Y. S. Wu, C. F. Tsang, G. Bodvarsson, A modeling approach for analysis of coupled multiphase fluid flow, heat transfer, and deformation in fractured porous rock, International Journal of Rock Mechanics and Mining Sciences 39 (4) (2002) 429–442.

[21] J. Kim, H. A. Tchelepi, R. Juanes, Stability, accuracy and efficiency of sequential methods for coupled flow and geomechanics, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 2009.

[22] J. Kim, H. A. Tchelepi, R. Juanes, Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits, Computer Methods in Applied Mechanics and Engineering 200 (13) (2011) 1591–1606.

[23] J. Kim, H. A. Tchelepi, R. Juanes, Stability and convergence of sequential methods for coupled flow and geomechanics: Drained and undrained splits, Computer Methods in Applied Mechanics and Engineering 200 (23) (2011) 2094–2116.

[24] A. Mikelić, M. F. Wheeler, Convergence of iterative coupling for coupled flow and geomechanics, Computational Geosciences 17 (3) (2013) 455–461.

[25] Z. Y. Wong, R. Rin, H. A. Tchelepi, R. N. Horne, Comparison of a fully implicit and sequential implicit formulation for geothermal reservoir simulations, in: 42nd Workshop on Geothermal Reservoir Engineering, 2017.

[26] Z. Y. Wong, R. Rin, H. A. Tchelepi, R. N. Horne, Modified sequential fully implicit method for geothermal reservoir simulation, in: 43rd Workshop on Geothermal Reservoir Engineering, 2018.

[27] D. G. Anderson, Iterative procedures for nonlinear integral equations, Journal of the ACM (JACM) 12 (4) (1965) 547–560.

[28] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing 7 (3) (1986) 856–869.

[29] H. F. Walker, P. Ni, Anderson acceleration for fixed-point iterations, SIAM Journal on Numerical Analysis 49 (4) (2011) 1715–1735.

[30] A. Toth, C. Kelley, Convergence analysis for Anderson acceleration, SIAM Journal on Numerical Analysis 53 (2) (2015) 805–819.

[31] J. W. Both, K. Kumar, J. M. Nordbotten, F. A. Radu, Anderson accelerated fixed-stress splitting schemes for consolidation of unsaturated porous media, arXiv preprint arXiv:1805.04211.

[32] L. Liu, D. E. Keyes, Field-split preconditioned inexact newton algorithms, SIAM Journal on Scientific Computing 37 (3) (2015) A1388–A1409.

[33] V. Dolean, M. Gander, W. Kheriji, F. Kwok, R. Masson, Nonlinear preconditioning: How to use a nonlinear schwarz method to precondition newton's method, SIAM Journal on Scientific Computing 38 (6) (2016) A3357–A3380.

[34] H. Hajibeygi, H. A. Tchelepi, Compositional multiscale finite-volume formulation, SPE Journal 19 (02) (2014) 316–326.

[35] R. Rin, P. Tomin, T. T. Garipov, D. V. Voskov, H. A. Tchelepi, General implicit coupling framework for multi-physics problems, in: SPE Reservoir Simulation Conference, Society of Petroleum Engineers, 2017.

[36] T. T. Garipov, P. Tomin, R. Rin, D. V. Voskov, H. A. Tchelepi, Unified thermo-compositional-mechanical framework for reservoir simulation, Computational Geosciences (2018) 1–19.

[37] A. St-Cyr, M. J. Gander, S. J. Thomas, Optimized multiplicative, additive, and restricted additive Schwarz preconditioning, SIAM Journal on Scientific Computing 29 (6) (2007) 2402–2425.

[38] M. J. Gander, Optimized Schwarz methods, SIAM Journal on Numerical Analysis 44 (2) (2006) 699–731.

[39] M. J. Gander, Schwarz methods over the course of time, Electron. Trans. Numer. Anal 31 (5) (2008) 228–255.

[40] X. S. Li, J. W. Demmel, Superlu_dist: A scalable distributed-memory sparse direct solver for unsymmetric linear systems, ACM Transactions on Mathematical Software (TOMS) 29 (2) (2003) 110–140.

[41] X. S. Li, An overview of SuperLU: Algorithms, implementation, and user interface, ACM Transactions on Mathematical Software (TOMS) 31 (3) (2005) 302–325.

[42] J. M. Ortega, W. C. Rheinboldt, Iterative solution of nonlinear equations in several variables, Academic Press, 1970.

[43] Y. Saad, Iterative methods for sparse linear systems, 2nd Edition, Vol. 82, SIAM, 2003.

[44] O. Coussy, Poromechanics, John Wiley & Sons, 2004.

[45] K. Aziz, A. Settari, Petroleum reservoir simulation, Chapman & Hall, 1979.

[46] O. C. Zienkiewicz, R. L. Taylor, The finite element method for solid and structural mechanics, Elsevier, 2005.

[47] T. T. Garipov, D. V. Voskov, H. A. Tchelepi, Rigorous coupling of geomechanics and thermal-compositional flow for sagd and es-sagd operations, in: SPE Canada Heavy Oil Technical Conference, Society of Petroleum Engineers, 2015.

[48] T. T. Garipov, J. A. White, A. Lapene, H. A. Tchelepi, Thermo-hydro-mechanical model for source rock thermal maturation, in: 50th US Rock Mechanics/Geomechanics Symposium, American Rock Mechanics Association, 2016.

[49] J. Mandel, Consolidation des sols (étude mathématique), Geotechnique 3 (7) (1953) 287–299.

[50] C. R. Faust, J. W. Mercer, Geothermal reservoir simulation: 2. numerical solution techniques for liquid-and vapor-dominated hydrothermal systems, Water Resources Research 15 (1) (1979) 31–46.

[51] Z. Y. Wong, R. N. Horne, D. V. Voskov, A geothermal reservoir simulator in AD-GPRS, in: Proceedings World Geothermal Congress, 2015.

[52] Z. Y. Wong, R. N. Horne, D. V. Voskov, Comparison of nonlinear formulations for geothermal reservoir simulations, in: 41st Workshop on Geothermal Reservoir Engineering, 2016.

[53] Stanford Geothermal Program (SGP), Proceedings of the special panel on geothermal model study, report SGP-TR-42, Energy Resources Engineering, Stanford University.

[54] Z. Y. Wong, R. N. Horne, H. A. Tchelepi, Sequential implicit nonlinear solver for geothermal simulation, Journal of Computational Physics 368 (2018) 236–253.