A constrained pressure-temperature residual (CPTR) method for non-isothermal multiphase flow in porous media

Thomas Roy\textsuperscript{a,*}, Tom B. Jönsthövel\textsuperscript{b}, Christopher Lemon\textsuperscript{c}, Andrew J. Wathen\textsuperscript{a}

\textsuperscript{a}Mathematical Institute, University of Oxford, Oxford, United Kingdom
\textsuperscript{b}Schlumberger Houston Production Technology Center, Houston, TX, United States of America
\textsuperscript{c}Schlumberger Abingdon Technology Center, Abingdon, United Kingdom

Abstract

In petroleum reservoir simulation, the standard preconditioner, the Constrained Pressure Residual (CPR) method, is a two-stage process which involves solving a restricted pressure system. Initially designed for isothermal models, this approach is often used in the thermal case. However, its treatment of the temperature variable does not incorporate heat diffusion, which is often dominant in thermal cases. In this paper, we present an extension of CPR: the Constrained Pressure-Temperature Residual (CPTR) method, where a restricted pressure-temperature system is solved in the first stage. In previous work, we introduced a block preconditioner with an efficient Schur complement approximation for a pressure-temperature system. Here, we extend this method for multiphase flow as the first stage of CPTR. The algorithmic performance of different two-stage preconditioners is evaluated for reservoir simulation test cases.

Keywords: preconditioning, iterative solvers, porous media, thermal reservoir simulation, multiphase flow

*Corresponding author

Email addresses: thomas.roy@maths.ox.ac.uk (Thomas Roy), tjonsthovel@slb.com (Tom B. Jönsthövel), clemon@slb.com (Christopher Lemon), andy.wathen@maths.ox.ac.uk (Andrew J. Wathen)
1. Introduction

Models of multiphase flow in porous media are used in many applications such as oil and gas recovery, geothermal energy, and carbon sequestration. In some cases, the effects of temperature changes on the fluid flow are significant enough that they are included in the model. Fluid properties affected by temperature include density, viscosity, and phase changes. In the case of heavy oil recovery, non-isothermal flow is an important part of enhanced oil recovery techniques. Under normal reservoir conditions, highly viscous oils do not flow easily to production wells. To reduce the viscosity and thus increase the fluid flow, the temperature inside the reservoir is increased, for instance with the injection a hot fluid such as steam, or with microwave heating. Decision making regarding the choice of extracting techniques can be guided by thermal reservoir simulation (see for example [1]).

In isothermal models, a pressure variable couples a number of secondary variables (saturations/concentrations) which characterize the location of different phases and hydrocarbons. The resulting PDE system is essentially elliptic with respect to pressure and hyperbolic with respect to the secondary variables. Since pressure drives the flow, successful solution techniques usually include a specific treatment of the pressure variable. This is true of classical methods such as IMPES (Implicit Pressure Explicit Saturations), and industry-standard methods such as CPR (Constrained Pressure Residual) [2] [3]. A preconditioner for the fully implicit system, CPR is a two-stage process where a restricted pressure system is solved, followed by an approximate solution of the coupled system. This method was later improved in [4] with the use of Algebraic Multigrid (AMG) [5] as a preconditioner for the pressure subsystem. The second stage is usually an incomplete factorization method such as incomplete LU factorization (ILU).

In non-isothermal models, a conservation of energy equation is added to the system along with a temperature (or enthalpy) variable. For fully implicit formulations, the industry standard preconditioner is CPR where temperature variables are grouped with the secondary variables. This is often appropriate since heat is being transported similarly to the saturations. However, heat is also diffused through rock and fluid. Diffusion can dominate in cases where the fluid flow is slow, for instance before viscous oils are properly heated, but also due to mesh refinements. In those cases, the second stage of CPR struggles to capture the heat diffusion, and so incomplete factorizations with additional fill are needed. Of course, this remedy is not
ideal in terms of scalability and memory requirements.

To avoid having to solve the fully coupled linearized system, there has been some interest in developing alternatives to the standard Newton’s method. For example, the Sequential Fully Implicit (SFI) method \cite{6} solves flow and transport problems sequentially at the nonlinear level. Although most recent work relates to coupling geomechanics, there is potential for the thermal problem \cite{7}. While the computational cost can be reduced by solving the smaller decoupled systems, considerable effort is needed for such methods to be robust.

An alternative to using AMG only in the first stage of CPR is the use of AMG for the fully coupled system. AMG methods for systems of PDEs \cite{8} have been applied successfully to several problems including reservoir simulation \cite{9}. In the context of thermal simulation, construction of the grid hierarchy using both pressure and temperature variables has been discussed in \cite{10}. However, the community’s recent focus is again in coupling flow and geomechanics, rather than thermal flow.

In this paper, we present an extension of CPR for non-isothermal flow. Instead of solving a restricted pressure system in the first stage of CPR, we solve a restricted pressure-temperature system, resulting in a Constrained Pressure-Temperature Residual method (CPTR). In \cite{11}, we presented a Schur complement approximation for the pressure-temperature system for non-isothermal single phase flow. Such a Schur complement approximation leads to an effective block preconditioner. Here, we propose an extension of that method to the multiphase flow situation and use it for the pressure-temperature subsystem in the first stage of CPTR.

In Section 2, we describe a system of PDEs for non-isothermal multiphase flow in porous media as well as its discretization. Then, in Section 3, we describe the CPR preconditioner and present our extension, CPTR, and the approximate Schur complement solver used in its first stage. Numerical results are presented in Section 4. We compare the different methods in terms of algorithmic performance and scalability. In Section 5, we conclude with a brief discussion of the results and further work.

2. Problem statement

In this section, we describe a coupled PDE system and its discretization.
2.1. Multiphase thermal flow in porous media

We describe the equations for multiphase phase flow in porous media coupled with thermal effects. The number of phases is typically two or three, for example, oil, water, and gas.

The volume fraction occupied by each phase \( \alpha \) is called the saturation of that phase, denoted \( S_\alpha \). We have the following saturation constraint:

\[
\sum_\alpha S_\alpha = 1.
\] (1)

In the absence of phase change, saturations only vary when one phase displaces the other through fluid flow. The ability of a phase \( \alpha \) to move is represented by the relative permeability \( k_{r\alpha} \), a dimensionless function of \( S_\alpha \).

2.1.1. Conservation of Mass

For each phase \( \alpha \), we have a conservation of mass equation of the following form:

\[
\phi \frac{\partial (S_\alpha \rho_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = f_\alpha \quad \text{in } \mathbb{R}_+ \times \Omega,
\] (2)

where \( \phi \) is the porosity of the rock, \( \rho_\alpha \) is the density of the fluid, \( \mathbf{u}_\alpha \) is the fluid velocity, \( f_\alpha \) is a source/sink term, and \( \Omega \) is the spatial domain. We further assume that the velocity follows Darcy’s law \[12\], i.e.

\[
\mathbf{u}_\alpha = -K \frac{k_{r\alpha}}{\mu_\alpha} \left( \nabla p - \rho_\alpha g \right),
\] (3)

where \( p \) is the pressure, \( K \) is a permeability tensor, \( \mu_\alpha \) is the viscosity, and \( g \) is gravitational acceleration. Note that we have ignored the effect of capillary pressures such that we only have one pressure \( p \) for all phases. Then, (2) becomes

\[
\phi \frac{\partial (S_\alpha \rho_\alpha)}{\partial t} - \nabla \cdot \left( \rho_\alpha \frac{K k_{r\alpha}}{\mu_\alpha} \left( \nabla p - \rho_\alpha g \right) \right) = f_\alpha \quad \text{in } \mathbb{R}_+ \times \Omega.
\] (4)

We also assume Neumann and Dirichlet boundary conditions

\[
-\frac{K k_{r\alpha}}{\mu_\alpha} \left( \nabla p - \rho_\alpha g \right) \cdot \mathbf{n} = g_{N,\alpha} \quad \text{on } \Gamma_N, \quad \text{for all } \alpha,
\] (5)

\[
S_\alpha = g_{D,\alpha}, \quad \text{for all } \alpha, \quad p = g_D \quad \text{on } \Gamma_D,
\] (6)

where \( g_{N,\alpha} \) is Neumann boundary data, \( g_D \) and \( g_{D,\alpha} \) are Dirichlet boundary data, \( \mathbf{n} \) is the unit outward normal vector on \( \partial \Omega = \Gamma_N \cup \Gamma_D \), and \( \Gamma_D \cap \Gamma_N = \emptyset \).
2.1.2. Conservation of Energy

Similarly, we have a conservation of energy equation for heat energy. Note that formulations where enthalpy is an independent variable are common, but here we consider temperature as an independent variable as in a reference commercial reservoir simulator [13]. Here, $c_{v,\alpha}$ and $c_r$ are the specific heat of the phase $\alpha$ and the rock, respectively, $\rho_r$ is the density of the rock, and $T$ is temperature. Here, $c_{v,\alpha} T$ represents the enthalpy of the phase $\alpha$, and $\rho_{\alpha} c_{v,\alpha} T$, its energy density. Heat energy is not only transported by a heat flux, but also by the fluid flux. We get the following advection-diffusion equation:

$$\frac{\partial}{\partial t} \left( \phi \sum_{\alpha} \rho_{\alpha} S_{\alpha} c_{v,\alpha} T + (1 - \phi) \rho_r c_r T \right) + \nabla \cdot \left( \sum_{\alpha} \rho_{\alpha} c_{v,\alpha} T \mathbf{u}_\alpha - \nabla \cdot (k_T \nabla T) \right) = f_T$$

in $\mathbb{R}_+ \times \Omega$, \hspace{1cm} (7)

where $f_T$ is a source/sink term, and $k_T$ is the thermal conductivity field. It is given by $k_T = (1 - \phi) k_{T,r} + \phi \sum_{\alpha} S_{\alpha} k_{T,\alpha}$, where $k_{T,r}$ and $k_{T,\alpha}$ are the conductivities of the rock and the phase $\alpha$, respectively. Assuming Darcy flow, we get

$$\frac{\partial}{\partial t} \left( \phi \sum_{\alpha} \rho_{\alpha} S_{\alpha} c_{v,\alpha} T + (1 - \phi) \rho_r c_r T \right) - \nabla \cdot \left( \sum_{\alpha} \rho_{\alpha} c_{v,\alpha} T K \frac{k_{r,\alpha}}{\mu_{\alpha}} (\nabla p - \rho_{\alpha} g) \right) - \nabla \cdot (k_T \nabla T) = f_T$$

in $\mathbb{R}_+ \times \Omega$, \hspace{1cm} (8)

We also assume Neumann and Dirichlet boundary conditions

$$- \left( \sum_{\alpha} \rho_{\alpha} c_{v,\alpha} T K \frac{k_{r,\alpha}}{\mu_{\alpha}} (\nabla p - \rho_{\alpha} g) + k_T \nabla T \right) \cdot \mathbf{n} = g_N^T \text{ on } \Gamma_N^T, \quad T = g_D^T \text{ on } \Gamma_D^T,$$

where $g_N^T$ is Neumann boundary data, $g_D^T$ is Dirichlet boundary data, $\partial \Omega = \Gamma_N^T \cup \Gamma_D^T$, and $\Gamma_D^T \cap \Gamma_N^T = \emptyset$.

Note that $c_{v,\alpha} T$ in the first term of (7) is actually the internal energy of the fluid. In the context of this paper, we consider liquids and therefore take the internal energy as equal to the enthalpy. For gases, it is more common for internal energy to also have some dependence on pressure and density.
2.1.3. Coupled problem

We assume that \( k_{ra}, \rho_{a}, \text{ and } \mu_{a} \) are empirically determined functions of saturations, pressure, and temperature. Our choices are given in Section 2.1.4.

Since the saturations satisfy the constraint (1), one of the saturation variables is explicitly replaced

\[
S_{\beta} = 1 - \sum_{\alpha \neq \beta} S_{\alpha}. \tag{10}
\]

We are interested in solving the following boundary value problem:

find \( p, T, S_{\alpha} \) for all \( \alpha \neq \beta \) such that

\[
\phi \frac{\partial (S_{\alpha} \rho_{a})}{\partial t} - \nabla \cdot \left( \rho_{a} \frac{k_{ra}}{\mu_{a}} (\nabla p - \rho_{a} g) \right) = f_{\alpha} \text{ in } \mathbb{R}_{+} \times \Omega, \text{ for all } \alpha, \tag{11}
\]

\[
\frac{\partial}{\partial t} \left( \phi \sum_{\alpha} \rho_{a} S_{\alpha} c_{v,a} T + (1 - \phi) \rho_r c_r T \right) - \nabla \cdot \sum_{\alpha} \rho_{a} c_{v,a} T K_{k_{ra}} \frac{k_{ra}}{\mu_{a}} (\nabla p - \rho_{a} g)
- \nabla \cdot (k_{T} \nabla T) = f_{T} \text{ in } \mathbb{R}_{+} \times \Omega, \tag{12}
\]

\[- \frac{K_{k_{ra}}}{\mu_{a}} (\nabla p - \rho g) \cdot n = g_{N,\alpha} \text{ on } \Gamma_{N}, \text{ for all } \alpha, \tag{13}\]

\[S_{\alpha} = g_{D,\alpha} \text{ for all } \alpha, \text{ } p = g_{D} \text{ on } \Gamma_{D}, \tag{14}\]

\[- \left( \sum_{\alpha} \rho_{a} c_{v,a} T K_{k_{ra}} \frac{k_{ra}}{\mu_{a}} (\nabla p - \rho_{a} g) + k_{T} \nabla T \right) \cdot n = g_{N}^{T} \text{ on } \Gamma_{N}^{T}, T = g_{D}^{T} \text{ on } \Gamma_{D}^{T}, \tag{15}\]

where \( \partial \Omega = \Gamma_{N} \cup \Gamma_{D} = \Gamma_{N}^{T} \cup \Gamma_{D}^{T}, \Gamma_{D} \cap \Gamma_{N} = \Gamma_{D}^{T} \cap \Gamma_{N}^{T} = \emptyset \), initial conditions for \( p, T, \) and \( S_{\alpha} \) are prescribed, and we explicitly eliminate \( S_{\beta} \) via (10).

2.1.4. Physical quantities

The relative permeability \( k_{ra} \), the densities \( \rho_{a} \), and viscosities \( \mu_{a} \) are empirically determined functions of saturations, temperature, and pressure. For the examples in this paper, we will consider the flow of two phases: a heavy oil and water, denoted \( o \) and \( w \), respectively.

For the oil viscosity \( \mu_{o} \), we choose a correlation from [14] where \( \mu_{o} \) is a function of temperature. For the oil density \( \rho_{o} \), we choose a function of
pressure and temperature given by an exponential correlation. Details and choices of parameters for both quantities can be found in Section 2.1.4 of [11].

For the water viscosity $\mu_w$, we use the following correlation [15]:

$$\mu_w(T_F) = \frac{A}{1 + BT_F + CT_F^2},$$

(16)

which takes temperature $T_F$ in °F and returns viscosity in cp (0.001 kg m$^{-1}$ s$^{-1}$). The parameters $A, B, C$ can be found in Table 1.

Table 1: Parameters for water viscosity correlation

| A    | B        | C                |
|------|----------|------------------|
| 2.1850 | 0.04012  | 5.1547×10$^{-6}$ |

For water density, we use Trangenstein’s modification of Kell’s formula [16] given by

$$\rho_w(p, T) = \frac{E_0 + E_1 T + E_2 T^2 + E_3 T^3 + E_4 T^4 + E_5 T^5}{1 + E_6 T} e^{C_w (p - E_7)}.$$  

(17)

which takes temperature $T$ in °C, and pressure $p$ in MPa. The parameters are given in Table 2.

Table 2: Parameters for the water density correlation

| $E_0$           | $E_1$         | $E_2$          | $E_3$             | $E_4$             | $E_5$           | $E_6$             | $E_7$          | $C_w$          |
|-----------------|---------------|----------------|-------------------|-------------------|-----------------|-------------------|----------------|----------------|
| 999.83952       | 16.955176     | -7.987×10$^{-3}$ | -46.170461×10$^{-6}$ | 105.56302×10$^{-9}$ | -280.54353×10$^{-12}$ | 16.87985×10$^{-3}$ | 10.2          | 3.98854×10$^{-4}$ |

For the relative permeability, we simply set it the be equal to the phase saturation. If we set $S_w = 1 - S_o$, we get $k_{ro} = S_o$, and $k_{rw} = 1 - S_o$. 

7
2.1.5. Source/sink terms

We first consider source/sink terms representing injection and production wells. A simple way to model these is by using point sources/sinks

\[
\begin{align*}
    f_\alpha(x) &= \sum_i q^i_{\alpha,\text{inj}}(p, T, S_0) \delta(x - x^i_{\text{inj}}) \rho_\alpha(p, T_{\text{inj}}) \\
    &\quad - \sum_j q^j_{\alpha,\text{prod}}(p, T, S_0) \delta(x - x^j_{\text{prod}}) \rho_\alpha(p, T),
\end{align*}
\]

(18)

\[
\begin{align*}
    f_T(x) &= \sum_\alpha \left( \sum_i q^i_{\alpha,\text{inj}}(p, T, S_0) \delta(x - x^i_{\text{inj}}) \rho_\alpha(p, T_{\text{inj}}) c_{v,\alpha} T_{\text{inj}} \\
    &\quad - \sum_j q^j_{\alpha,\text{prod}}(p, T, S_0) \delta(x - x^j_{\text{prod}}) \rho_\alpha(p, T) c_{v,\alpha} T \right),
\end{align*}
\]

(19)

where \(x_{\text{inj}}\) and \(x_{\text{prod}}\) represent the location of injection and production wells, respectively, \(\delta(x)\) is the Dirac delta function, \(q^i_{\alpha,\text{inj}}\) and \(q^j_{\alpha,\text{prod}}\) are the wells’ injection and production rates, respectively, for phase \(\alpha\).

The following details about the source/sink terms are identical to those in our previous work [11], and are included for the sake of completeness.

The production rate \(q_{\text{prod}}\) is usually given by a constant target production rate. Similarly, the injection rate \(q_{\text{inj}}\) is given by a target injection rate. These rates can only be maintained if the pressure at the production well does not drop below a minimum pressure, and the pressure at the injection well does not go above a maximum pressure. In those cases, a well model is required.

We consider the commonly used Peaceman well model [17, 18] for anisotropic media with \(K = \text{diag}(K_x, K_y, K_z)\) as the permeability tensor field. In this case the rates are given by

\[
q_\alpha = \frac{2\pi h K_e k_{r,\alpha}}{\mu_\alpha \ln(r_e/r_w)} \left( p_{\text{bh}} - p \right),
\]

(20)

where \(h\) is the height of well opening, \(K_e = \sqrt{K_x K_y}\) is the equivalent permeability, \(p_{\text{bh}}\) is the bottom-hole pressure, \(r_w\) is the well radius, and \(r_e\) is the equivalent radius which can be calculated using

\[
r_e = \frac{0.14 \left( (K_y/K_x)^{1/2} D_x^2 + (K_x/K_y)^{1/2} D_y^2 \right)^{1/2}}{0.5 \left( (K_y/K_x)^{1/4} + (K_x/K_y)^{1/4} \right)},
\]

(21)
where $D_x$ and $D_y$ are the horizontal lengths of the grid cell. Since we want to allow mesh refinements, we do not want the model to change as we vary the grid size. Therefore, we arbitrarily fix $D_x = D_y = 5$ meters, and also choose $h = 5$ meters and $r_w = 0.1$ meters.

Oil recovery techniques for heavy oils can include electromagnetic heating \[19\]. These can be expressed as source terms for the energy equation. For simplicity, we do not use an electromagnetic model and choose the simple function

\[ f_T = \sum_i U_{\text{heater}}(p, T) \delta(x - x_{\text{heater}}^i)(T_{\text{heater}} - T), \]  

(22)

where $x_{\text{heater}}$ represent the location of heaters, $U_{\text{heater}}$ is the heat transfer coefficient, and $T_{\text{heater}}$ is the target heating temperature. For our simulations, we have a heating coefficient of $5.44409 \times 10^{-6}$ Js$^{-1}$K$^{-1}$. For simplicity, we also choose $T_{\text{heater}}$ to be the same as $T_{\text{inj}}$.

2.2. DG0 discretization

In reservoir simulation, Finite Volume methods are most commonly used \[20\]. Since the flux entering a given volume is identical to that leaving an adjacent one, these methods are conservative. Additionally, upwind schemes introduce substantial numerical diffusion, which helps with stability. We use a discontinuous Galerkin (DG) method \[21\] that is equivalent to a Finite Volume method used in reservoir simulation. The resulting weak formulation allows us to implement our problem in the open source Finite Element software Firedrake \[22\]. The details of the discretization are given in Section 2.2 of \[11\].

Let $\mathcal{T} = \{E_i, i \in \mathcal{I}\}$ be a partition of $\Omega$ into open element domains $E_i$ such that union of their closure is $\overline{\Omega}$, where $\mathcal{I}$ is a set of indices. We consider a piecewise constant approximation of our solution variables, i.e. in the approximation space $\mathcal{V}_h = P^0_{DG}$ with basis $\{\phi_i = 1_{E_i} \mid i \in \mathcal{I}\}$. The DG0 approximation of a function $u$ is $u_h = \sum_{i \in \mathcal{I}} u_i \phi_i$, with coefficients $u_i \in \mathbb{R}$.

For a given ordering of the indices in $\mathcal{I}$, we denote by $u^+$ and $u^-$ the limit value of $u$ for two cells sharing an edge. We define the jump of $v$ as $[v] = v^+ - v^-$. Cell centers are denoted by $h^+$ and $h^-$.  

2.2.1. Semidiscrete Problem

We now discretize \[11\]-\[15\] in space using the semidiscrete DG0 formulation described in \[11\]. We describe the discretization for two-phase flow with an oil phase and a water phase, denoted $o$ and $w$, respectively. Here, we set
$S_w = 1 - S_o$. Assuming homogeneous Neumann boundary conditions, the variational problem is: find the approximation $(p, T, S_o) \in P^0_{DG} \times P^0_{DG} \times P^0_{DG}$ satisfying the conservation of water mass equation:

$$
\int_{\Omega} \phi \frac{\partial (\rho_w (1 - S_o))}{\partial t} q \, dx + \int_{\Gamma_{int}} [q] \left( \left\{ K \right\} \rho_{w,up} \left( \frac{[p]}{\|h^+ - h^-\|} - \left\{ \rho_w \right\} \mathbf{g} \cdot \mathbf{n}_e \right) \right) dS - \int_{\Omega} f_w q \, dx = 0,
$$

(23)

the conservation of energy equation:

$$
\int_{\Omega} \phi \frac{\partial (c_{v,w} \rho_w (1 - S_o) T)}{\partial t} r \, dx + \int_{\Omega} \phi \frac{\partial (c_{v,o} \rho_o S_o T)}{\partial t} r \, dx + \int_{\Omega} (1 - \phi) \rho_r c_r \frac{\partial T}{\partial t} r \, dx + \sum_{\alpha = o, w} \int_{\Gamma_{int}} [r] \left\{ K \right\} \frac{\rho_{\alpha,up}}{\mu_{\alpha,up}} r_{up} \left( \frac{[p]}{\|h^+ - h^-\|} - \left\{ \rho_\alpha \right\} \mathbf{g} \cdot \mathbf{n}_e \right) dS
$$

$$
+ \int_{\Gamma_{int}} [r] \left\{ k_T \right\} \left( \frac{[T]}{\|h^+ - h^-\|} \right) dS - \int_{\Omega} f_T r \, dx = 0,
$$

(24)

and the conservation of oil mass equation:

$$
\int_{\Omega} \phi \frac{\partial (\rho_o S_o)}{\partial t} s \, dx + \int_{\Gamma_{int}} [s] \left( \left\{ K \right\} \frac{\rho_{o,up}^w}{\mu_{o,up}^w} \left( \frac{[p]}{\|h^+ - h^-\|} - \left\{ \rho_o \right\} \mathbf{g} \cdot \mathbf{n}_e \right) \right) dS
$$

$$
- \int_{\Omega} f_o s \, dx = 0,
$$

(25)

for all $(q, r, s) \in P^0_{DG} \times P^0_{DG} \times P^0_{DG}$. The unit outward pointing normal of a cell is denoted by $\mathbf{n}_e$. The brackets $\{ \}$ denote the average across the facets, and the double brackets $\{\{\}$ denote the harmonic average across the facets. The upwind quantities for phase $\alpha$ are given by

$$
(u_\alpha)_{up} = \begin{cases} 
 u \mid_{E_1} & \text{if } \frac{[p]}{\|h^+ - h^-\|} - \left\{ \rho_\alpha \right\} \mathbf{g} \cdot \mathbf{n}_e \geq 0, \\
 u \mid_{E_2} & \text{if } \frac{[p]}{\|h^+ - h^-\|} - \left\{ \rho_\alpha \right\} \mathbf{g} \cdot \mathbf{n}_e < 0.
\end{cases}
$$

(26)

The delta functions in the source terms are approximated on the grid as:

$$
\delta(x) = \begin{cases} 
 1/|E_i| & \text{if } x \in E_i, \\
 0 & \text{otherwise.}
\end{cases}
$$

(27)
2.2.2. Fully Discretized Problem

For time discretization, we use the backward Euler method. We define the three mappings, which are linear with respect with their last argument. As follows, we define the conservation of water mass mapping:

\[
F_w(p^{n+1}, T^{n+1}, S_{w}^{n+1}; q) := \int_\Omega \phi \rho_w^{n+1}(1 - S_o^{n+1}) - \rho_w^n(1 - S_o^n) \frac{q}{\Delta t} \, dx \\
+ \int_{\Gamma_{\text{int}}} [q] \left( \frac{\rho_w^{n+1}}{\mu_w^{n+1}} \right) (\nabla [p^{n+1}]) \cdot (\nabla h^+ - \nabla h^-) - \{ \rho_w^{n+1} \} g \cdot n_e \right) \, dS \\
- \int_{\Omega} f_w^{n+1} \, dx = 0, \tag{28}
\]

the conservation of energy mapping:

\[
F_e(p^{n+1}, T^{n+1}, S_{o}^{n+1}; r) := \int_\Omega \phi c_v w \rho_w^{n+1} T^{n+1} - \rho_w^n T^n \frac{r}{\Delta t} \, dx \\
+ \int_\Omega \phi c_{v,o} \rho_o^{n+1} T^{n+1} - S_o^n \rho_o^n T^n \frac{r}{\Delta t} \, dx + \int_\Omega (1 - \phi) \rho_r c_r T^{n+1} - T^n \frac{r}{\Delta t} \, dx \\
+ \sum_{\alpha=o,w} \int_{\Gamma_{\text{int}}} [r] \left( \frac{\rho^{n+1}}{\mu^{n+1}} \right) (\nabla [p^{n+1}]) \cdot (\nabla h^+ - \nabla h^-) - \{ \rho^{n+1} \} g \cdot n_e \right) \, dS \\
+ \int_{\Gamma_{\text{int}}} [r] \left( \frac{T^{n+1}}{\nabla h^+ - \nabla h^-} \right) \, dS - \int_{\Gamma_{\text{int}}} f^{n+1} T \, dx, \tag{29}
\]

and the conservation of oil mass mapping:

\[
F_o(p^{n+1}, T^{n+1}, S_{o}^{n+1}; s) := \int_\Omega \phi \rho_o^{n+1} S_o^{n+1} - \rho_o^n S_o^n \frac{s}{\Delta t} \, dx \\
+ \int_{\Gamma_{\text{int}}} [s] \left( \frac{\rho_o^{n+1}}{\mu_o^{n+1}} \right) (\nabla [p^{n+1}]) \cdot (\nabla h^+ - \nabla h^-) - \{ \rho_o^{n+1} \} g \cdot n_e \right) \, dS \\
- \int_{\Omega} f_o^{n+1} s \, dx = 0. \tag{30}
\]

Let

\[
F(p, T, S_o; q, r, s) := F_w(p, T, S_o; q) + F_e(p, T, S_o; r) + F_o(p, T, S_o; s). \tag{31}
\]
which is linear in $q$, $r$, and $s$, but nonlinear in $p$, $T$, and $S_o$.

At each time-step, given the previous solution $(p^n, T^n, S_o^n)$, we search for $(p^{n+1}, T^{n+1}, S_o^{n+1}) \in \mathbb{P}_{DG}^0 \times \mathbb{P}_{DG}^0 \times \mathbb{P}_{DG}^0$ such that

$$F(p^{n+1}, T^{n+1}, S_o^{n+1}; q, r, s) = 0 \text{ for all } (q, r, s) \in \mathbb{P}_{DG}^0 \times \mathbb{P}_{DG}^0 \times \mathbb{P}_{DG}^0. \quad (32)$$

In order to use the Schur complement approximation described in Section 3.3.1, we instead solve the equivalent problem: find $(p^{n+1}, T^{n+1}, S_o^{n+1}) \in \mathbb{P}_{DG}^0 \times \mathbb{P}_{DG}^0 \times \mathbb{P}_{DG}^0$ such that

$$F^*(p^{n+1}, T^{n+1}, S_o^{n+1}; q, r, s) = 0 \text{ for all } (q, r, s) \in \mathbb{P}_{DG}^0 \times \mathbb{P}_{DG}^0 \times \mathbb{P}_{DG}^0, \quad (33)$$

where

$$F^*(p, T, S_o; q, r, s) := F_p(p, T, S_o; q) + F_e(p, T, S_o; r) + F_o(p, T, S_o; s) \quad (34)$$

is $F$ where $F_w$ has been replaced by the following “pressure equation” mapping:

$$F_p(p, T, S_o; q) := c_{v,w} F_w(p, T, S_o; q) + c_{v,o} F_o(p, T, S_o; q). \quad (35)$$

The choice of this weighted sum will be justified in Section 3.3.1.

Further scaling of the different equations might be needed for the robustness of the solution algorithms. We discuss this briefly at the end of Section 3.3.

### 3. Solution Algorithms

The system of nonlinear equations (33) can be written as a system of nonlinear equations for the real coefficients $p_i$ and $T_i$ of the DG0 functions $p^{n+1}$ and $T^{n+1}$, respectively. Let $x$ be the vector of these coefficients and $G$ the function such that $G(x) = 0$ is equivalent to (33). By linearizing this equation with Newton’s method, we must solve at each iteration

$$\frac{\partial G}{\partial x} |_{x=x_k} (x_{k+1} - x_k) = -G(x_k). \quad (36)$$

The resulting linearized systems can be written as a block system of the form

$$A \delta x = \begin{bmatrix} A_{pp} & A_{pT} & A_{ps} \\ A_{Tp} & A_{TT} & A_{Ts} \\ A_{sp} & A_{sT} & A_{ss} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta T \\ \delta s \end{bmatrix} = \begin{bmatrix} b_p \\ b_T \\ b_s \end{bmatrix} = b. \quad (37)$$
The blocks of the pressure-temperature submatrix are the discrete versions of

\[ A_{pp} \sim \phi \frac{1}{\Delta t} \sum_{\alpha} c_{v,\alpha} S_{\alpha}(\rho_{\alpha})_{p} + \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{p} - \sum_{\alpha} c_{v,\alpha} (f_{\alpha})_{p}, \quad (38) \]

\[ A_{pT} \sim \phi \frac{1}{\Delta t} \sum_{\alpha} c_{v,\alpha} S_{\alpha}(\rho_{\alpha})_{T} + \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{T} - \sum_{\alpha} c_{v,\alpha} (f_{\alpha})_{T}, \quad (39) \]

\[ A_{Tp} \sim \phi \frac{1}{\Delta t} \sum_{\alpha} c_{v,\alpha} S_{\alpha}(\rho_{\alpha})_{p} T + \sum_{\alpha} \nabla \cdot (c_{v,\alpha} T(\rho_{\alpha} u_{\alpha})_{p}) - (f_{T})_{p}, \quad (40) \]

\[ A_{TT} \sim \phi \frac{1}{\Delta t} \sum_{\alpha} c_{v,\alpha}(\rho_{\alpha} + (\rho_{\alpha})_{T} T) + (1 - \phi)\frac{\mu_{\alpha} c_{r}}{\Delta t} + \sum_{\alpha} \nabla \cdot (c_{v,\alpha} \rho_{\alpha} u_{\alpha}) \\
\quad + \sum_{\alpha} \nabla \cdot (c_{v,\alpha} T(\rho_{\alpha} u_{\alpha})_{T}) - \nabla \cdot (k_{T} \nabla) - (f_{T})_{T}, \quad (41) \]

where

\[ (\rho_{\alpha} u_{\alpha})_{p} = -\frac{K_{k_{r} \alpha}}{\mu_{\alpha}} (\rho_{\alpha}(\nabla - (\rho_{\alpha})_{p} g) + (\rho_{\alpha})_{p}(\nabla p - \rho_{\alpha} g)), \quad (42) \]

and

\[ (\rho_{\alpha} u_{\alpha})_{T} = -K_{k_{r} \alpha} \left[ \left( \frac{\rho_{\alpha}}{\mu_{\alpha}} \right) _{T} (\nabla p - \rho_{\alpha} g) - \frac{\rho_{\alpha}}{\mu_{\alpha}} (\rho_{\alpha})_{T} g \right]. \quad (43) \]

All coefficients in (38)-(43) are evaluated at the previous Newton iterate \((p_{k}, T_{k}, S_{\alpha,k})\), and \((.)_{p}\) and \((.)_{T}\) denote the partial derivatives with respect to \(p\) and \(T\), respectively.

We seek to solve the resulting linearized system \((37)\) using iterative methods \([23]\).

Krylov subspace methods approximate the solution of \(Ax = b\) by constructing a sequence of Krylov subspaces, \(K_{n} = \{b, Ab, A^{2}b, \ldots, A^{n-1}b\}\). The generalized minimal residual method (GMRES) \([24]\) is a widely-used Krylov subspace method suitable for nonsymmetric linear systems. For GMRES, the approximate solution \(x_{n}\) is formed by minimizing the Euclidean norm of the residual \(r_{n} = Ax_{n} - b\) over the subspace \(K_{n}\).

For complex linear systems such as the ones considered in this paper, an efficient preconditioner is required in order to achieve a rapid convergence.
with a Krylov method. In this section, we will detail different preconditioning techniques which take into account the specific structure of (37). We first mention some methods which are important ingredients of the preconditioning techniques.

Incomplete LU factorization (ILU) [23, 25] is a general preconditioning technique in which sparse triangular factors are used to approximate the system matrix $A$. This preconditioner requires assembling the factors, followed by solving two sparse triangular systems. A popular way to determine the sparsity pattern of the factors is to simply choose the relevant triangular parts of the sparsity pattern of $A$. This is known as ILU(0). More generally, using the sparsity pattern of $A^{k+1}$ is called ILU($k$).

Multigrid methods [26, 27, 28] use hierarchies of coarse grid approximations in order to solve differential equations. Smoothing operations (such as a Jacobi or Gauss-Seidel iterations) are combined with coarse grid corrections on increasingly coarser grids. For positive definite elliptic PDEs, it is known that multigrid methods can provide optimal solvers (in the sense of linear scalability with the dimension of the discretized problem).

Algebraic Multigrid (AMG) [5, 29] uses information from the entries of the system matrix rather than that of the geometric grid. This makes AMG an ideal black-box solver for elliptic problems. Although it can be used to solve simpler problems, AMG is often used as a preconditioner for Krylov subspace methods in problems which are essentially elliptic. Relative to preconditioners such as ILU, parallel variants of multigrid methods retain more effectiveness, as we will discuss in Section 4.

**Two-stage preconditioners.** Let $M_1$ and $M_2$ be two preconditioners for the linear system $Ax = b$ for which we have the action of their (generally approximate) inverses $M_1^{-1}$, and $M_2^{-1}$. Applying a multiplicative two-stage preconditioner can be done as follows:

1. Precondition using $M_1$: $x_1 = M_1^{-1}b$;
2. Compute the new residual: $b_1 = b - Ax_1$;
3. Precondition using $M_2$ and correct: $x = M_2^{-1}b_1 + x_1$.

The action of the two-stage preconditioner can be written as

$$M^{-1} = M_2^{-1}(I - AM_1^{-1}) + M_1^{-1}. \quad (44)$$
3.1. Constrained pressure residual (CPR)

The standard preconditioner for multiphase flow in porous media is the Constrained Pressure Residual method (CPR) [2]. Initially developed for isothermal cases, CPR is also used in the thermal case where the temperature variable is treated like the saturation variables. The linear system \( (37) \) is thus rearranged in the following form:

\[
A\delta x = \begin{bmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta s \end{bmatrix} = \begin{bmatrix} b_p \\ b_s \end{bmatrix} = b, \quad (45)
\]

where the temperature is grouped with the saturations.

The CPR preconditioner is a two-stage preconditioner where the first stage preconditioner \( M_1 \) is given by

\[
M_1^{-1} \approx \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \quad (46)
\]

where \( A_{pp}^{-1} \) is approximated using an AMG V-cycle. The second preconditioner is chosen such that \( M_2^{-1} \approx A^{-1} \), usually with an incomplete LU factorization method (ILU). Applying CPR to the right-hand side \( b = [b_p, b_s]^T \) can be done as follows:

1. Solve the pressure subsystem: \( A_{pp}x_p = b_p \);
2. Compute the new residual: \( \tilde{b} = [b_p, b_s] - [A_{pp} A_{ps}] x_p \);
3. Precondition and correct: \( \begin{bmatrix} \delta p \\ \delta s \end{bmatrix} = M_2^{-1} \tilde{b} + \begin{bmatrix} \delta p \\ 0 \end{bmatrix} \).

Thus

\[
\delta = M_2^{-1} \left( I - (A - M_2) \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right) b. \quad (47)
\]

In addition to the two-stage preconditioner, decoupling operators are often used to reduce the coupling between the pressure equation and the saturation variables. Indeed, an approximation of the pressure equation \( A_{pp}\delta p + A_{ps}\delta s = b_s \) is performed in the first stage of CPR where the saturation coupling \( A_{ps} \) is ignored. A decoupling operator is a left preconditioner applied a priori to \( \text{[45]} \) of the form

\[
\begin{bmatrix} I & -D \\ 0 & I \end{bmatrix}, \quad (48)
\]
The most often-used approximations are Quasi-IMPES (QI) and True-IMPES (TI) \cite{4, 30, 31}. These approximations are
\[
D_{QI} = \text{diag}(A_{ps}) \text{diag}(A_{ss})^{-1},
\]
\[
D_{TI} = \text{colsum}(A_{ps}) \text{colsum}(A_{ss})^{-1},
\]
respectively. Here, diag(.) and colsum(.) return block diagonal matrices.

By performing this decoupling operation on the system \((45)\) before CPR, the first stage now consists in solving a subsystem for the Schur complement
\[
S_p = A_{pp} - DA_{sp}
\]
instead of the original pressure block. However, the properties of the resulting \(S_p\) need to be amenable to the performance of AMG (e.g. M-matrix properties). While this is nearly guaranteed in the black-oil case \cite{10}, it does not necessarily follow for compositional flow or thermal flow.

In our case, we perform a weighted sum of the mass equations in \((35)\). This is equivalent to choosing \(D\) as a block diagonal matrix where the coefficients are \(c_{v,\alpha}\) for the mass conservation equation of phase \(\alpha\) and 0 for the energy conservation equation.

### 3.2. Constrained pressure-temperature residual (CPTR)

We introduce a CPR-like two-stage preconditioner where a pressure-temperature subsystem is solved approximately in the first stage using an extension of the block preconditioner from \cite{11}. We call this method Constrained Pressure-Temperature Residual (CPTR).

Let the pressure-temperature submatrix
\[
A_{00} = \begin{bmatrix} A_{pp} & A_{pT} \\ A_{Tp} & A_{TT} \end{bmatrix}, \quad \text{such that} \quad A = \begin{bmatrix} A_{00} & A_{0s} \\ A_{s0} & A_{ss} \end{bmatrix}.
\]
\((49)\)

For CPTR, The first stage preconditioner \(M_1\) is given by
\[
M_1^{-1} = \begin{bmatrix} A_{00}^{-1} & 0 \\ 0 & 0 \end{bmatrix},
\]
\((50)\)

where \(A_{00}^{-1}\) is an approximation of the action of the inverse of \(A_{00}\).

In our case, we consider an approximation given by a Schur complement factorization detailed in Section \(\ref{sec:shur}\). A likely alternative would be an AMG method for systems of PDEs. See \cite{8} for a description of such methods. Efficient implementations of such AMG methods include BoomerAMG \cite{32} and multigrid reduction (MGR) in the hypre library \cite{33}, as well as Smoothed Aggregation in the ML package \cite{34}. In particular, BoomerAMG has been shown to be effective in diffusion-dominated two-phase flow problems \cite{35}, and MGR has also had some success with multiphase flow problems \cite{36, 37}.
Such approaches are more easily generalizable than the Schur complement approach since they usually only rely on algebraic information from the system matrix.

Similarly to CPR, the second preconditioner is chosen such that \(M^{-1} \approx A^{-1}\), usually with an incomplete LU factorization method (ILU).

Let \(b_0 = [b_p \ b_T]^\top\). Applying CPTR to the right-hand side \(b = [b_0 \ b_s]^\top\) can be done as follows:

1. Solve the pressure-temperature subsystem: \(A_{00}x_0 = b_0\);
2. Compute the new residual: \(\tilde{b} = [b_0 \ b_s] - [A_{00} \ A_{0s}]x_0\);
3. Precondition and correct: \(\left[\begin{array}{c} \delta_0 \\ \delta_s \end{array}\right] = M^{-1}2\tilde{b} + \left[\begin{array}{c} \delta_0 \\ 0 \end{array}\right]\).

Thus
\[
\delta = M^{-1}_2 \left( I - (A - M_2) \begin{bmatrix} A_{00}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right) b. \tag{51}
\]

3.3. Block preconditioner for the pressure-temperature subsystem

In this section, we discuss our choice for pressure-temperature solver \(A_{00}^{-1}\). Consider the following decomposition of the pressure-temperature submatrix:

\[
A_{00} = \begin{bmatrix} I & 0 \\ [A_{Tp}A_{pp}^{-1}] & I \end{bmatrix} \begin{bmatrix} 0 & A_{pp}^{-1}A_{pT} \\ 0 & S_T \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \tag{52}
\]

where \(S_T = A_{TT} - A_{Tp}A_{pp}^{-1}A_{pT}\) is the Schur complement. The inverse of \(A_{00}\) is given by

\[
A_{00}^{-1} = \begin{bmatrix} I & -A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & S_T^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}. \tag{53}
\]

Even if \(A_{00}\) is sparse, the Schur complement \(S_T\) is generally dense. A common preconditioning technique is to use the blocks of the factorization \(52\) combined with a sparse approximation of the Schur complement \(38\). With an appropriate Schur complement approximation \(\tilde{S}_T\), our block preconditioner for \(A_{00}\) is given by

\[
\tilde{A}_{00}^{-1} = \begin{bmatrix} I & -A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & \tilde{S}_T^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}. \tag{54}
\]
Both $A_{pp}^{-1}$ and $\tilde{S}_T^{-1}$ are approximated using an AMG V-cycle.

Applying the block preconditioner to the right-hand side $[b_p \ b_T]^{\top}$ can be done as follows:

1. Solve the pressure subsystem: $A_{pp}x_p = b_p$;
2. Compute the new energy equation residual: $\hat{b}_T = b_T - A_{Tp}x_p$;
3. Solve the Schur complement subsystem: $\tilde{S}_T\delta_T = \hat{b}_T$;
4. Compute the new mass equation residual: $\hat{b}_p = x_p - A_{pT}\delta_T$;
5. Solve the pressure subsystem: $A_{pp}\delta_p = \hat{b}_p$.

Thus, $[\delta_p \ \delta_T]^{\top} = \tilde{A}_{00}^{-1}[b_p \ b_T]^{\top}$.

The CPTR method with this pressure-temperature involves strictly more work than the first stage of the CPR method described in Section 3.1. The additional work is essentially an additional application of the solver $A_{pp}^{-1}$, as well as the construction and application of the solver $\tilde{S}_T^{-1}$.

### 3.3.1. Schur complement approximation

Common sparse approximations for the Schur complement are $\tilde{S}_A = A_{TT}$ and $\tilde{S}_{\text{diag}} = A_{TT} - A_{Tp}\text{diag}(A_{pp})^{-1}A_{pT}$. Here we present a Schur complement approximation which performs significantly better than these simple approximations. This Schur complement approximation requires the construction of the pressure equation (35).

We first consider the steady-state problem for pressure and temperature: find $p$, $T$ such that

$$\sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha u_\alpha) = 0 \quad \text{in } \Omega. \quad (55)$$

$$\sum_\alpha \nabla \cdot (\rho_\alpha c_{v,\alpha} T u_\alpha) - \nabla \cdot (k_T \nabla T) = 0 \quad \text{in } \Omega, \quad (56)$$

where $u_\alpha$ is given by (3), and we have homogeneous Neumann boundary conditions. Here we will consider the linearized pressure-temperature subsystem in a infinite-dimensional setting. Applying a Newton method to (55)-(56), we obtain a block matrix of the form (49) where the blocks are

$$A_{pp} = \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha u_\alpha)_{p}, \quad (57)$$

$$A_{pT} = \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha u_\alpha)_{T}, \quad (58)$$
where we have used the product rule for the divergence operator in (59) and (60). Then, the second term of the Schur complement (which corresponds in the continuous setting to the Poincaré-Steklov operator) becomes

\[
\begin{align*}
A_{T} &= \sum_{\alpha} \nabla \cdot (\rho_{\alpha} c_{v,\alpha} T u_{\alpha})_{p} - \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{p} + \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{p}, \\
A_{TT} &= \sum_{\alpha} \nabla \cdot (\rho_{\alpha} c_{v,\alpha} T u_{\alpha})_{T} - \nabla \cdot (k_{T} \nabla) \\
&= \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{T} + \sum_{\alpha} c_{v,\alpha} \nabla T \cdot (\rho_{\alpha} u_{\alpha})_{T} \\
&+ \sum_{\alpha} c_{v,\alpha} T \nabla \cdot (\rho_{\alpha} u_{\alpha})_{T} - \nabla \cdot (k_{T} \nabla),
\end{align*}
\]

where we have used the product rule for the divergence operator in (59) and (60). Then, the second term of the Schur complement (which corresponds in the continuous setting to the Poincaré-Steklov operator) becomes

\[
A_{T} A^{-1}_{pp} A_{pT} = \left( \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{p} + \sum_{\alpha} c_{v,\alpha} T \nabla \cdot (\rho_{\alpha} u_{\alpha})_{p} \right) \left( \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{p} \right)^{-1} \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{T} \\
= T \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{T} + \sum_{\alpha} c_{v,\alpha} \nabla T \cdot (\rho_{\alpha} u_{\alpha})_{p} \\
\left( \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{p} \right)^{-1} \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{T} \\
= T \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{T} + \nabla T \cdot \sum_{\alpha} c_{v,\alpha} (\rho_{\alpha} u_{\alpha})_{p} \\
\left( \nabla \cdot \sum_{\alpha} c_{v,\alpha} (\rho_{\alpha} u_{\alpha})_{p} \right)^{-1} \nabla \cdot \sum_{\alpha} c_{v,\alpha} (\rho_{\alpha} u_{\alpha})_{T}.
\]

We notice that in \( A_{TT} - A_{T} A^{-1}_{pp} A_{pT} \), the terms \( T \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_{T} \) cancel. This is due to the construction of the pressure equation (35). We are left...
with

\[
S_T = \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha}) - \nabla \cdot (k_T \nabla) + \sum_{\alpha} c_{v,\alpha} \nabla T \cdot (\rho_{\alpha} u_{\alpha})_T
\]

\[
- \nabla T \cdot \sum_{\alpha} c_{v,\alpha} (\rho_{\alpha} u_{\alpha})_p \left( \nabla \cdot \sum_{\alpha} c_{v,\alpha} (\rho_{\alpha} u_{\alpha})_p \right)^{-1} \nabla \cdot \sum_{\alpha} c_{v,\alpha} (\rho_{\alpha} u_{\alpha})_T. \tag{61}
\]

One of the nonlinear terms has canceled, and so we consider if it is possible that the last two terms also cancel. Consider the operator

\[
\sum_{\alpha} c_{v,\alpha} (\rho_{\alpha} u_{\alpha})_p \left( \nabla \cdot \sum_{\alpha} c_{v,\alpha} (\rho_{\alpha} u_{\alpha})_p \right)^{-1} \nabla . \tag{62}
\]

We now make the same approximations as in [11], which includes that \( \rho_{\alpha} \) is constant with respect to \( p \), and that we use a two-point flux approximation for the facet integrals. (While the assumption that \( \rho_{\alpha} \) is close to being a constant with respect to \( p \) is appropriate for liquid water and hydrocarbons, it may be less applicable in the case of gases.) This allows us to replace the operator (62) by the identity, giving us the following Schur complement approximation:

\[
\tilde{S}_T = \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_T - \nabla \cdot (k_T \nabla). \tag{63}
\]

The extension to the time-dependent case and the addition of source/sink terms does not differ from the single phase case, and so we refer the reader to [11] for more details. In this case, the source/sink terms related to production wells satisfy \( f_{T,\text{prod}} = \sum_{\alpha} c_{v,\alpha} T f_{\alpha,\text{prod}} \), while those for injection wells satisfy \( f_{T,\text{inj}} = \sum_{\alpha} c_{v,\alpha} T_{\text{inj}} f_{\alpha,\text{inj}} \). Also, the source term for heaters is given by \( f_T = U(T_{\text{heater}} - T) D_{\text{heaters}} \), where \( D_{\text{heaters}} \) is the sum of delta functions for the location of heaters. The Schur complement approximation is given by

\[
\tilde{S}_T = \phi \sum_{\alpha} \frac{c_{v,\alpha} \rho_{\alpha}}{\Delta t} + (1 - \phi) \frac{\rho_{c_r} c_r}{\Delta t} + \sum_{\alpha} c_{v,\alpha} \nabla \cdot (\rho_{\alpha} u_{\alpha})_T - \nabla \cdot (k_T \nabla)
\]

\[
+ U D_{\text{heaters}} - \sum_{\alpha} c_{v,\alpha} f_{\alpha,\text{prod}}. \tag{64}
\]

We can obtain the discretized version of this operator from (29) by removing the terms depending on the previous time-step, and evaluating the nonlinear terms at the previous Newton iteration. We get the following bilinear
operator:

\[ S_e(\delta T, r) := \sum_\alpha \int_\Omega \phi c_{v,\alpha} S_\alpha \rho_\alpha \frac{\delta T}{\Delta t} r \, dx + \int_\Omega (1 - \phi) \rho_r c_r \frac{\delta T}{\Delta t} r \, dx \]
\[ + \sum_\alpha \int_{\Gamma_{int}} [r] \left\{ \begin{array}{c} k_r \end{array} \right\} (k_{\alpha})^{up} c_{v,\alpha} (\rho_\alpha)^{up} (\mu_\alpha)^{up} (\delta T)^{up} \left( \frac{[p]}{\|h^+ - h^-\|} - \{ \rho_\alpha \} \mathbf{g} \cdot \mathbf{n} \right) \, dS \]
\[ + \int_{\Gamma_{int}} [r] \left\{ \begin{array}{c} k_T \end{array} \right\} \frac{[\delta T]}{\|h^+ - h^-\|} \, dS + \int_\Omega \left( - \sum_\alpha c_{v,\alpha} f_{\alpha,\text{prod}} + U D_{\text{heaters}} \right) \delta T \, dx. \]

(65)

3.4. Discussion on multi-stage preconditioners

The first stage of CPR is given by the approximate solution of \( A_{pp} \tilde{\delta}p = b_p \), which is an approximation of the pressure equation \( A_{pp} \delta p + A_{pT} \delta T + A_{ps} \delta s = b_p \). If the coupling blocks \( A_{pT} \) and \( A_{ps} \) are significant, the approximate pressure \( \tilde{\delta}p \) might not be representative. This could be countered by reducing the coupling between the pressure equation and the non-pressure variables. Recall that we are not using algebraic decoupling operators such as Quasi-IMPES and True-IMPES described in Section 3.1, but are simply taking a weighted sum of the mass equations to create a pressure equation. Similarly, the first stage of CPTR is also currently vulnerable to a strong coupling between both the pressure and energy equations and saturation variables since it ignores both the \( A_{ps} \) and \( A_{Ts} \).

A simple way to reduce this coupling without the use of algebraic decoupling operators could be to change the order of the two-stage preconditioner. Then, the approximate solution of the fully coupled system (through ILU) acts as a decoupling operator. This has been used for example in [35].

For our test cases, we do not observe a significant difference when using this variant of CPR. On the other hand, with the variant CPTR, we originally observed a significant decrease in the number of GMRES iterations in general, and an increase in the number of Newton iterations in some cases. Further investigation revealed that during GMRES, the residual of the energy equation was being reduced faster than those of the pressure and oil equations. Then, since the energy equation has larger values when using SI units, the convergence threshold for the total residual was reached too early for the residual of the other equations to have been reduced sufficiently. Even if Newton itself is scale invariant, the Jacobian system is only being solved
to a relative tolerance. This inadequate solution to the linear system caused Newton to converge slower. An obvious remedy is to scale the equations such that they are of similar magnitude.

In our case, we multiply the pressure equation by a reference temperature, and the oil equation by the product of the reference temperature and a reference specific heat coefficient. Thus, we modify $F^*$ from (34) to be

$$F^*(p, T, S_o; q, r, s) := T_{\text{ref}} F_p(p, T, S_o; q) + F_e(p, T, S_o; r) + c_{\text{ref}} T_{\text{ref}} F_o(p, T, S_o; s).$$

This scaling has little effect on the number of iterations for the original CPR and CPTR. However, it makes variant CPTR perform essentially the same as the original CPTR. Thus, in the next section, we will only consider the original versions of CPR and CPTR. Nonetheless, we will use the scaling (66) by default.

As mentioned above, for variant CPTR, the residual for the energy equation is being reduced faster than the residuals of the mass equations. This indicates that decoupling the energy equation (in this case using ILU) can improve the solution of the restricted pressure-temperature subsystem. However, the residual of the other equations is not reduced as fast for this method. It remains to be seen if algebraic decoupling operators similar to those mentioned in Section 3.1 should be used with CPTR. If so, it is unclear if the Schur complement approximation approach described in Section 3.3 could be modified for the new pressure-temperature subsystem. Since they do not rely as much on the exact structure of the equations, AMG methods for the pressure-temperature subsystem could be a promising alternative.

Finally, extending CPR and CPTR to more stages does not appear to have a significant effect on the number of iterations. For example, one could consider a three-stage preconditioner where ILU is used before and after a pressure (or pressure-temperature) restricted solution. Even in the cases where this results in fewer iterations, the additional step does not appear to be worth the additional cost. This is also true of the case where a pressure (-temperature) restricted solve is done before and after ILU.

4. Numerical Results

In this section, we perform numerical experiments for CPR and CPTR. These are implemented on the open source Finite Element software Firedrake [22, 39]. The linear algebra backend is the PETSc library [40], allowing efficient and parallel computations. The CPR and CPTR preconditioners are
implemented by providing PETSc options. Our Schur complement approximation is implemented through Firedrake’s Python interface. Our implementation is available on GitHub\footnote{https://github.com/tlroy/thermalporous}.

For the first stage of CPTR, we use our Schur complement approximation \eqref{eq:schur}. Unless stated otherwise, both the pressure block $A_{pp}$ and the approximate Schur complement are inverted using a V-cycle of AMG. We use BoomerAMG \cite{boomeramg} from the hypre library \cite{hypre} with default parameters, i.e. a symmetric-SOR/Jacobi relaxation scheme (one sweep up, one sweep down), Falgout coarsening, classical Ruge-Stüben interpolation, and Gaussian Elimination as the coarse grid solver. This implementation has a very efficient parallel version of AMG. For the second stage of CPR and CPTR, we use ILU(k) as provided in PETSc. Unless stated otherwise, ILU(0) is used. In parallel, we use block Jacobi with ILU(k) for each block (the partition is assigned when Firedrake does the discretization).

The nonlinear solver is Newton’s method with line search, and the linear solver is right-preconditioned GMRES \cite{gmres}. The convergence tolerance of Newton’s methods is $10^{-8}$ for the relative function norm and relative step size norm. The convergence tolerance for GMRES is $10^{-8}$ for the relative residual norm.

For all cases, we consider oil and water with densities and viscosities as described in Section \ref{subsec:physical}. The other physical parameters are shown in Table \ref{tab:parameters}. These parameters are representative of those used in commercial reservoir simulators. We also only consider homogeneous Neumann boundary conditions.

\begin{table}[h]
\centering
\begin{tabular}{ll}
\hline
Initial pressure & $4.1369 \times 10^7$ Pa \\
Conductivity of oil & $0.15$ Wm$^{-1}$K$^{-1}$ \\
Conductivity of water & $0.6005638$ Wm$^{-1}$K$^{-1}$ \\
Conductivity or rock & $1.7295772056$ Wm$^{-1}$K$^{-1}$ \\
Specific heat of oil & $2093.4$ JK$^{-1}$kg$^{-1}$ \\
Specific heat of water & $4181.3$ JK$^{-1}$kg$^{-1}$ \\
Specific heat of rock & $920$ JK$^{-1}$kg$^{-1}$ \\
\hline
\end{tabular}
\caption{Physical parameters for test cases}
\end{table}

For the scaling parameters in \eqref{eq:scaling}, we choose $T_{\text{ref}} = T_{\text{prod}}$ and $c_{\text{ref}} =$ \footnote{\url{https://github.com/tlroy/thermalporous}}.
$S_o c_{v,o} + (1 - S_o) c_{v,w}$, where $S_o$ is the initial oil saturation in the reservoir, which is uniform in all the test cases here. For the preconditioners we consider, the number of iterations is not very sensitive to the choice of scaling parameters.

For all cases, we evaluate the performance of the methods by comparing the number of linear iterations per nonlinear iteration.

**Computational cost.** We note that applying the first stage of CPTR, the block preconditioner from Section 3.3 should be slightly more than three times the cost of applying the first stage of CPR. The relative cost of the second stage will depend on things such as the size and dimension of the problem, and the number of processors used. To provide one example, we look at the CPU timings for the serial case given in Table 12 (a problem with 52,728 degrees of freedom). Applying the first stage of CPR takes an average of 0.008s, while the first stage of CPTR takes an average of 0.029s. Applying the second stage of CPR/CPTR with ILU(0) takes an average of 0.02s, and 0.034s with ILU(1). These timings were performed on a Lenovo ThinkCentre M920q with Intel(R) Core(TM) i5-8500T CPU @ 2.10GHz.

### 4.1. SPE10 test case

We consider the benchmark problem SPE10 [41] and use its permeability and porosity fields. For the following tests, we consider the top 20 layers of SPE10, such that the domain has dimensions $365.76 \times 670.56 \times 12.192$ meters, and the mesh is $60 \times 220 \times 20$. This problem has a highly heterogeneous permeability field. The permeability is isotropic in the $x$-$y$ plane, and anisotropic otherwise. The permeability fields are illustrated in Figure 1.

![Permeability fields](image)

(a) Permeability in $x$ and $y$ directions.  
(b) Permeability in $z$ direction

Figure 1: Log of permeability of the SPE10 test case (mm$^2$).

For our first test case, we position an injection well in the middle of the reservoir, and production wells in each corner. The wells are completed
throughout the 20 layers. For the injection and production rates, we use the Peaceman well model. The production wells produce with a bottom-hole pressure of $2.7579 \times 10^7$ Pa. The injection well injects hot water with a maximum rate of $q = 1.8 \times 10^{-3} \text{m}^3\text{s}^{-1}$. The initial temperature in the reservoir is 288.706 K and the injection temperature is 373.15 K. For the heater case (H), heater placement is the same as for the well case, and so are the initial and heating temperatures. For the well and heater case (W+H), we combine both wells and heaters.

For each case, we simulate injection and production for 100 days where the time steps are chosen adaptively such that Newton’s method usually converges in less than 10 iterations. For each case, we look at the performance of different methods as we increase the number of processors from 1 to 16. We consider CPR and CPTR where $A^{-1}_{pp}$ and $\tilde{S}^{-1}_T$ are approximated by either and AMG V-cycle or direct LU factorization, denoted CPR-AMG or CPTR-AMG and CPR-LU and CPTR-LU, respectively. We include the latter since we believe that there are cases where the AMG V-cycles in the first stage of CPR/CPTR are not as effective as usual. Since CPTR relies on three AMG V-cycles, it is likely more susceptible to its weaknesses.

For the case with both wells and heaters, we take a total of 104 time-steps. The average linear iterations per nonlinear iteration are shown in Table 4. We observe that all methods have a small increase in iterations. At the beginning of the simulation, heating is the most significant effect. For the first time-steps, CPTR-AMG takes fewer iterations than CPR-AMG, but the total number of iterations becomes the same at the 30th time-step (around 12 days). In contrast, CPTR-LU has fewer iterations than CPR-LU. This indicates that single AMG V-cycles are not enough in this case to provide good pressure and temperature solutions in the first stage of CPTR-AMG.

Table 4: Strong scaling SPE10 3D wells and heaters case, $S_o = 0.9$. Average linear iterations per nonlinear iteration.

| Method/Num. proc. | 1   | 2   | 4   | 8   | 16  |
|-------------------|-----|-----|-----|-----|-----|
| CPR-AMG           | 11.5| 11.6| 11.6| 11.6| 11.8|
| CPTR-AMG          | 11.8| 12.2| 12.2| 12.2| 12.3|
| CPR-LU            | 10.2| 10.3| 10.2| 10.2| 10.4|
| CPTR-LU           | 10  | 9.91| 9.91| 10  | 10.1|

For the case with only wells, we take a total of 101 time-steps. The average linear iterations per nonlinear iteration are shown in Table 5. We
again observe that both CPR and CPTR have a small increase in iterations. For the few first time-steps, CPTR-AMG takes fewer iterations than CPR-AMG, but the total number of iterations becomes the same at the 10th time-step (around 4 minutes). Again, CPTR-LU has fewer iterations than CPR-LU, which indicates that the additional AMG V-cycles of CPTR-AMG are not helping in this case.

Table 5: Strong scaling SPE10 3D wells case, $S_o = 0.9$. Average linear iterations per nonlinear iteration.

| Method/Num. proc. | 1    | 2    | 4    | 8    | 16   |
|-------------------|------|------|------|------|------|
| CPR-AMG           | 11.7 | 11.9 | 11.8 | 11.8 | 12   |
| CPTR-AMG          | 12.1 | 12.4 | 12.4 | 12.4 | 12.6 |
| CPR-LU            | 10.4 | 10.5 | 10.4 | 10.3 | 10.4 |
| CPTR-LU           | 10.2 | 10   | 10   | 10.1 | 10.1 |

For the case with only heaters, we take a total of 45 time-steps, except for CPR-AMG with 8 processors, where 46 time-steps are used. In that case, GMRES fails to converge within 200 iterations so we take a smaller time-step. A large jump in GMRES iterations for that same Newton step also appears for CPR-AMG with 2 processors (177 iterations), CPR-AMG with 16 processors (94 iterations), and CPTR-AMG with 16 processors (168 iterations). Further investigation revealed that some values of the saturation variable $S_o$ were greater than one during that Newton step. Since a negative water saturation ($1-S_o$) causes the $A_{pp}$ block to become indefinite, the AMG coarse grid will not be representative and ruin the performance of CPR-AMG and CPTR-AMG. Using smaller time-steps or damping factors for Newton are standard strategies for avoiding negative saturations. As for the previous cases, we see that CPTR-LU takes slightly fewer iterations than CPR-LU.

Table 6: Strong scaling SPE10 3D heaters case, $S_o = 0.9$. Average linear iterations per nonlinear iteration.

| Method/Num. proc. | 1    | 2    | 4    | 8    | 16   |
|-------------------|------|------|------|------|------|
| CPR-AMG           | 8.55 | 9.79 | 8.8  | 8.8  | 9.28 |
| CPTR-AMG          | 8.5  | 8.66 | 8.75 | 8.75 | 9.66 |
| CPR-LU            | 8.31 | 8.37 | 8.37 | 8.39 | 8.36 |
| CPTR-LU           | 8.2  | 8.3  | 8.3  | 8.34 | 8.32 |

The SPE10 test case was not originally designed for thermal cases, but
rather for testing upscaling techniques. Its difficulty lies in its highly heterogeneous permeability field. Instead of considering cases with such a coarse grid, in Section 4.3, we will consider finer grids and thus diffusion-dominated flows.

4.2. Numerical justification of the Schur complement approximation

We now perform a numerical comparison of the action of the inverses of the different Schur complement approximations. This is done for cases similar to the single phase cases described in [11]. The only difference is that we start with \( S_0 = 0.9 \) and inject water with temperature 373.15 K. In brief, the domain is a 2D slice of SPE10 with a 60 \( \times \) 120 grid. For the well case (W), we have one production well and one injection well. These are located in regions of high permeability. For the heater case (H), heater placement is the same as for the well case, and for the well and heater case (W+H), we combine both wells and heaters. For the high permeability cases (h.p.), we increase the permeability by a factor of 1,000. While the resulting permeability values are not representative of physical ones, they give a simple example of advection-dominated heat flow.

In Table 7, we compare the different Schur complement approximations by looking at the condition number of their inverse applied to the full Schur complement. While this condition number does not directly inform us about how well the preconditioner performs, it is a good indication of the quality of the approximations. For the cases, H and W stand for heaters and wells, respectively, and h.p. stands for high permeability (increased by a factor 1,000). We observe that \( \tilde{S}_T \) is a good Schur complement approximation even for the high permeability cases where the other approximations struggle.

In terms of the performance of the solver, \( \tilde{S}_T \) always results in fewer
GMRES iterations for CPTR (results not shown here). For harder cases (for example high permeability), this difference is significant; the linear solver can even fail to converge before the prescribed maximum number of iterations. See [11] for single phase flow examples where the other Schur complements exhibit worse performance for heterogeneous or anisotropic permeability fields.

4.3. Homogeneous test cases

We now compare the performance of several methods for problems with homogeneous permeability. We begin by a brief mesh refinement study for 2D cases. Then, we look at both weak and strong scaling for 3D cases. For weak scaling, the problem size is increased proportionally with the number of processors, while for strong scaling, the problem size remains fixed. For weak scaling, the problem size is increased via uniform mesh refinement.

For traditional reservoir simulation, strong scaling is often more relevant than weak scaling. Indeed, reservoir models are typically given with geological properties on a (usually rather coarse) fixed grid, for example, the SPE10 test case. Another good reason to study strong scaling is to isolate the effects of parallelization from the effects of mesh refinement observed for weak scaling. Nonetheless, the effects of mesh refinement are still relevant in the context of reservoir simulation, for example around wells or other features of interest.

4.3.1. Mesh refinement study

We perform a mesh refinement study for 2D test cases. The domain is 50 $\times$ 50 meters with an $N \times N$ grid. Since this is a 2D case, we do not include gravity. The permeability is $3 \times 10^{-13}$ m$^2$ and the porosity is 0.2. Starting at $N = 20$, we double $N$ until we reach $N = 320$.

We consider two cases: one with heaters, and one with wells. For each case, we have 6 source/sink terms with 3 located on either side of the square domain. In the heating case, these terms are all heaters, while for the well cases, one side has injection wells while the other has production wells. For the well case, the production/injection rates are constant at $q = 3 \times 10^{-7}$ m$^3$/s$^{-1}$, and water is injected at a temperature of 373.15 K. For the heater case, we have a heating temperature of 373.15 K. The initial saturation for both cases is set to $S_o = 0.9$. We then take two time-steps of 10 days.

For each case we calculate the average number of linear iterations per nonlinear iteration and these are shown for the well case and heater case in
Tables 8 and 9, respectively. In both cases, we observe that for smaller values of \(N\), CPR and CPTR perform similarly. However, as we refine the mesh, the number of iterations for CPR increases significantly, while it seems to remain constant for CPTR.

CPR treats temperature like a saturation, essentially assuming that heat is simply being transported by the fluid flow. When this is the case, ILU is sufficient. Since heat diffusion becomes more noticeable as the mesh is refined, the CPR strategy does not hold for finer meshes, and ILU is not enough. In contrast, the treatment of temperature for CPTR does not depend on ILU, but rather AMG. Hence, CPTR tackles heat diffusion appropriately on fine meshes.

### Table 8: 2D Well case. Average linear iterations per nonlinear iteration.

| method/\(N\) | 20 | 40 | 80 | 160 | 320 |
|-------------|----|----|----|-----|-----|
| CPR         | 5.29 | 6.29 | 9.14 | 15.5 | 31.5 |
| CPTR        | 5.29 | 5.29 | 5.43 | 5.64 | 6.53 |
| CPR-ILU(1)  | 5.29 | 5.43 | 6.43 | 10  | 20  |

### Table 9: 2D Heater case. Average linear iterations per nonlinear iteration.

| method/\(N\) | 20 | 40 | 80 | 160 | 320 |
|-------------|----|----|----|-----|-----|
| CPR         | 7.71 | 7.71 | 10.8 | 19  | 34.6 |
| CPTR        | 7.57 | 7.43 | 6.83 | 7   | 6.6 |
| CPR-ILU(1)  | 6.71 | 6.86 | 7.67 | 12.2 | 21.6 |

#### 4.3.2. Weak scaling

For weak scaling, we compare the parallel performance of the methods as we increase the number of processors and problem size. The domain is \(50 \times 50 \times 50\) meters with an \(N \times N \times N\) grid. Since this is a 3D case, we include gravity. The permeability is \(3 \times 10^{-13} \text{m}^{2}\) and the porosity is 0.2. We seek to have around 50,000 degrees of freedom per processor. Thus, for the number of processors 1, 2, 4, 8, and 16, we have \(N = 26, 33, 41, 52, 65\).

We consider three cases: one with heaters, and two with wells. For each case, we have 21 source/sink terms both near the top and bottom of the domain. In the heating case, these terms are all heaters, while for the well cases, the top terms are injection wells and the bottom terms, production
wells. For the well cases, the production/injection rates are constant at \( q = 10^{-7} \text{m}^3\text{s}^{-1} \). Water is injected at a temperature of 373.15 K.

The initial saturation for the heating case is set to \( S_o = 0.5 \), and for the well cases to \( S_o = 1 \) and \( S_o = 0.99 \). For each case, we take 5 time-steps and look at the average number of linear iterations per nonlinear iteration. The size of the time-steps is \( \Delta t = 10 \) days for the heating case and the well case with \( S_o = 0.99 \), and \( \Delta t = 4 \) days for the well case with \( S_o = 1 \).

Table 10: Weak scaling: 3D Heating case, \( S_o = 0.5 \). Average linear iterations per nonlinear iteration.

| Method/Num. proc. | 1   | 2   | 4   | 8   | 16  |
|-------------------|-----|-----|-----|-----|-----|
| CPR               | 6.06| 7.47| 8.88| 10.1| 11.8|
| CPTR              | 5.47| 5.47| 5.88| 5.88| 6   |
| CPR-ILU(1)        | 5.53| 7.41| 8.71| 9.65| 10.9|

Table 11: Weak scaling: 3D Well case, \( S_o = 0.99 \). Average linear iterations per nonlinear iteration.

| Method/Num. proc. | 1   | 2   | 4   | 8   | 16  |
|-------------------|-----|-----|-----|-----|-----|
| CPR               | 6   | 7.78| 8.31| 9.54| 12.4|
| CPTR              | 5.58| 5.67| 5.85| 6.08| 6.55|
| CPR-ILU(1)        | 5.58| 7.33| 8   | 9.08| 11.8|

The results for the heating case are shown in Table 10. We observe that as we increase the number of processors from 1 to 16, the number of iterations for both versions of CPR nearly doubles, while the number of iterations for CPTR increases by less than 10%. The results are similar for the well case with \( S_o = 0.99 \), shown in Table 11. The number of iterations for both versions of CPR more than doubles, and the number of iterations for CPTR increases by less than 20%. Note that for both heating and well cases, different starting constant saturations provide similar scaling results, except when \( S_o \) is greater than 0.999 for the well case.

As shown in Table 12, the results are significantly different for the well case with \( S_o = 1 \). Indeed, the number of iterations for all methods increase by the same factor of \( 2/3 \). We observe that CPTR performs essentially the same as CPR (but at an additional cost). This could indicate that the temperature solution given by the first stage of CPTR is either wrong or
trivial. Additionally, CPTR scales better in Tables 10 and 11, which suggests again that there is a large decoupling error in the first stage of CPTR. As discussed in Section 3.4, we suspect this is due to a strong coupling between the temperature and saturation. Indeed, the energy equation solved in the first stage of CPTR uses $S_o$ from the previous

Furthermore, the well case with $S_o = 1$ is very similar to some single phase test cases in [11], except that oil is injected instead of water. In that paper, the block preconditioner scales significantly better than CPR in both weak and strong scaling. This is a clear indication that the issue with CPTR is related with the coupling of the saturation variable.

4.3.3. Strong scaling

We use the same problem as the previous section on the finest mesh ($N = 65$). We keep the problem size fixed while increasing the number of processors.

In Table 13, we observe for the heating case that as we increase the number of processors from 1 to 16, the number of iterations increases by around 30% for CPR, 60% for CPR-ILU(1) and 10% for CPTR. The bulk of the increase for CPR-ILU(1) occurs when going from 1 to 2 processors, around 40%. This showcases how CPR-ILU(1) is much more effective in serial. We observe similar results for the well case with $S_o = 0.99$ in Table 14.

| Method/Num. proc. | 1   | 2   | 4   | 8   | 16  |
|-------------------|-----|-----|-----|-----|-----|
| CPR               | 9.23| 10.8| 11.2| 11.4| 11.9|
| CPTR              | 5.77| 5.85| 5.92| 6.08| 6.38|
| CPR-ILU(1)        | 6.85| 9.69| 10.2| 10.8| 10.9|

Increasing the number of processors from 1 to 16, the number of iterations
Table 14: Strong scaling: 3D Well case, $S_o = 0.99$. Average linear iterations per nonlinear
iteration.

| Method/Num. proc. | 1    | 2    | 4    | 8    | 16   |
|-------------------|------|------|------|------|------|
| CPR               | 9.45 | 11.4 | 12   | 12.2 | 12.4 |
| CPTR              | 6.15 | 6.3  | 6.35 | 6.3  | 6.55 |
| CPR-ILU(1)        | 6.8  | 10.4 | 11   | 11.2 | 11.8 |

increases by around 31% for CPR, 73% for CPR-ILU(1) and 7% for CPTR. Again, there is a large increase in the number of iterations for CPR-ILU(1) when going from 1 to 2 processors.

Table 15: Strong scaling: 3D Well case, $S_o = 1$. Average linear iterations per nonlinear
iteration.

| Method/Num. proc. | 1    | 2    | 4    | 8    | 16   |
|-------------------|------|------|------|------|------|
| CPR               | 22   | 22.8 | 22.4 | 22.6 | 23.5 |
| CPTR              | 22.1 | 22.8 | 22.3 | 22.5 | 23.4 |
| CPR-ILU(1)        | 16.6 | 18   | 18.1 | 18.8 | 19.3 |

In Table 15, we show the results for the well case for $S_o = 1$. We observe a similar increase in the number of iterations for CPR and CPTR of 7% and 6%, respectively. For CPR-ILU(1), the increase in iterations is of 17%, and half of this increase happens when going from 1 to 2 processors.

Both CPR and CPTR use Block ILU, which becomes weaker as the number of processors increases. In contrast, AMG should scale much better. In cases other than $S_o = 1$, CPTR scales much better than CPR. For those cases, the first stage of CPTR gives an accurate temperature update, while CPR relies on ILU for its temperature update.

5. Conclusion

In this work, we have implemented a fully implicit parallel non-isothermal multiphase flow in porous media simulator including two preconditioning strategies, CPR and CPTR, which uses a block preconditioner with our own Schur complement approximation. On coarse grids, both methods exhibit a similar number of iterations, which means that CPR outperforms CPTR in terms of computational cost. In these cases, heat diffusion is not very significant, so CPTR is not necessary. In other cases, however, CPTR displays
much better scalability in terms of mesh refinement, as well as parallelization. Additionally, the first stage of CPTR is vulnerable to decoupling errors with saturation variables which can negate the scalability advantages of the method.

We believe that more work is needed to understand the decoupling error from the first stage of CPTR. Algebraic decoupling operators similar to those mentioned in Section 3.1 may be necessary. When using decoupling operators, it might be harder to apply an approximate Schur complement block preconditioner. Using an AMG method for the pressure-temperature subsystem is a promising alternative.

6. Acknowledgments

This publication is based on work partially supported by the EPSRC Centre For Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with Schlumberger.

References

[1] K. H. Coats, et al., In-situ combustion model, Society of Petroleum Engineers Journal 20 (06) (1980) 533–554.

[2] J. R. Wallis, Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 1983.

[3] J. R. Wallis, R. P. Kendall, T. E. Little, et al., Constrained residual acceleration of conjugate residual methods, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 1985.

[4] S. Lacroix, Y. V. Vassilevski, J. Wheeler, M. F. Wheeler, Iterative solution methods for modeling multiphase flow in porous media fully implicitly, SIAM Journal on Scientific Computing 25 (3) (2003) 905–926.

[5] J. Ruge, K. Stüben, Algebraic multigrid, in: Multigrid methods, Vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1987, Ch. 4, pp. 73–130.
[6] 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.

[7] Z. Y. Wong, Sequential-implicit newton’s method for geothermal reservoir simulation, Ph.D. thesis, Stanford University (2018).

[8] T. Clees, AMG strategies for PDE systems with applications in industrial semiconductor simulation, Ph.D. thesis, Universität zu Köln (2005).

[9] S. Gries, K. Stüben, G. L. Brown, D. Chen, D. A. Collins, et al., Pre-conditioning for efficiently applying algebraic multigrid in fully implicit reservoir simulations, SPE Journal 19 (04) (2014) 726–736.

[10] S. Gries, System-AMG approaches for industrial fully and adaptive implicit oil reservoir simulations, Ph.D. thesis, Universität zu Köln (2015).

[11] T. Roy, T. B. Jönsthövel, C. Lemon, A. J. Wathen, A block pre-conditioner for non-isothermal flow in porous media, Journal of Computational Physics 395 (2019) 636–652.

[12] H. Darcy, Les fontaines publiques de la ville de Dijon, Victor Dalmont, 1856.

[13] D. DeBaun, T. Byer, P. Childs, J. Chen, F. Saaf, M. Wells, J. Liu, H. Cao, L. Pianelo, V. Tilakraj, et al., An extensible architecture for next generation scalable parallel reservoir simulation, in: SPE Reservoir Simulation Symposium, 2005.

[14] T. Bennison, Prediction of heavy oil viscosity, in: Presented at the IBC Heavy Oil Field Development Conference, Vol. 2, 1998, p. 4.

[15] J. W. Grabowski, P. K. Vinsome, R. C. Lin, G. Behie, B. Rubin, et al., A fully implicit general purpose finite-difference thermal model for in situ combustion and steam, in: SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers, 1979.

[16] G. S. Kell, Density, thermal expansivity, and compressibility of liquid water from 0. deg. to 150. deg., correlations and tables for atmospheric pressure and saturation reviewed and expressed on 1968 temperature scale, Journal of Chemical and Engineering Data 20 (1) (1975) 97–105.
[17] D. W. Peaceman, et al., Interpretation of well-block pressures in numerical reservoir simulation (includes associated paper 6988), Society of Petroleum Engineers Journal 18 (03) (1978) 183–194.

[18] Z. Chen, Y. Zhang, Well flow models for various numerical methods, International Journal of Numerical Analysis & Modeling 6 (3) (2009) 375–388.

[19] A. Sahni, M. Kumar, R. B. Knapp, et al., Electromagnetic heating methods for heavy oil reservoirs, in: SPE/AAPG Western Regional Meeting, Society of Petroleum Engineers, 2000.

[20] R. J. LeVeque, Finite volume methods for hyperbolic problems, Vol. 31, Cambridge University Press, 2002.

[21] B. Riviere, Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation, SIAM, 2008.

[22] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, P. H. J. Kelly, Firedrake: automating the finite element method by composing abstractions, ACM Transactions on Mathematical Software (TOMS) 43 (3) (2016) 24.

[23] Y. Saad, Iterative methods for sparse linear systems, SIAM, 2003.

[24] 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.

[25] J. Meijerink, H. A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, Mathematics of Computation 31 (137) (1977) 148–162.

[26] A. Brandt, Multi-level adaptive solutions to boundary-value problems, Mathematics of Computation 31 (138) (1977) 333–390.

[27] W. L. Briggs, V. E. Henson, S. F. McCormick, A multigrid tutorial, SIAM, 2000.

[28] U. Trottenberg, C. W. Oosterlee, A. Schuller, Multigrid, Academic press, 2000.
[29] K. Stüben, An introduction to algebraic multigrid, in: Multigrid, Academic Press, 2001, pp. 413–532.

[30] S. Lacroix, Y. V. Vassilevski, M. F. Wheeler, Iterative solvers of the implicit parallel accurate reservoir simulator (IPARS), I: single processor case, TICAM report 00-28, The University of Texas at Austin (2000).

[31] R. Scheichl, R. Masson, J. Wendebourg, Decoupling and block preconditioning for sedimentary basin simulations, Computational Geosciences 7 (4) (2003) 295–318.

[32] V. Henson, U. Yang, BoomerAMG: A parallel algebraic multigrid solver and preconditioner, Applied Numerical Mathematics 41 (1) (2002) 155–177.

[33] R. D. Falgout, U. M. Yang, hypre: A library of high performance preconditioners, in: International Conference on Computational Science, Springer, 2002, pp. 632–641.

[34] M. Gee, C. Siefert, J. Hu, R. Tuminaro, M. Sala, ML 5.0 smoothed aggregation users guide, Tech. Rep. SAND2006-2649, Sandia National Laboratories (2006).

[35] Q. M. Bui, H. C. Elman, J. D. Moulton, Algebraic multigrid preconditioners for multiphase flow in porous media, SIAM Journal on Scientific Computing 39 (5) (2017) S662–S680.

[36] L. Wang, D. Osei-Kuffuor, R. Falgout, I. Mishev, J. Li, et al., Multigrid reduction for coupled flow problems with application to reservoir simulation, in: SPE Reservoir Simulation Conference, Society of Petroleum Engineers, 2017.

[37] Q. M. Bui, L. Wang, D. Osei-Kuffuor, Algebraic multigrid preconditioners for two-phase flow in porous media with phase transitions, Advances in water resources 114 (2018) 19–28.

[38] H. C. Elman, D. J. Silvester, A. J. Wathen, Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics, Oxford University Press, USA, 2014.
[39] R. C. Kirby, L. Mitchell, Solver composition across the PDE/linear algebra barrier, SIAM Journal on Scientific Computing 40 (1) (2018) C76–C98.

[40] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc Web page, http://www.mcs.anl.gov/petsc (2017).

[41] M. Christie, M. Blunt, et al., Tenth SPE comparative solution project: A comparison of upscaling techniques, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 2001.