A CONSTRAINED PRESSURE-TEMPERATURE RESIDUAL (CPTR) METHOD FOR NON-ISOTHERMAL MULTIPHASE FLOW IN POROUS MEDIA[*]

THOMAS ROY[†], TOM B. JÖNSTHÖVEL[‡], CHRISTOPHER LEMON[§], AND
ANDREW J. WATHEN[†]

Abstract. For both isothermal and thermal petroleum reservoir simulation, the constrained pressure residual (CPR) method is the industry-standard preconditioner. This method is a two-stage process involving the solution of a restricted pressure system. While initially designed for the isothermal case, CPR is also the standard for thermal cases. However, its treatment of the energy conservation equation 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.

Key words. preconditioning, iterative methods, porous media, thermal reservoir simulation, multiphase flow

AMS subject classifications. 65F08, 65F10, 76S05, 65M08, 65M60

DOI. 10.1137/19M1292023

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 of 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, [10]).

In isothermal models, a pressure variable couples a number of secondary variables (saturations/concentrations) that 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 the implicit pressure explicit saturation (IMPES) method and industry-standard ones such as the constrained pressure resid-

[†]Mathematical Institute, University of Oxford, Oxford OX2 6GG, United Kingdom (roy@maths.ox.ac.uk, andy.wathen@maths.ox.ac.uk).

[‡]Schlumberger Norway Technology Center, Asker 1383, Norway (tjonsthovel@slb.com).

[§]Schlumberger Abingdon Technology Center, Abingdon OX14 4RU, United Kingdom (clemon@slb.com).

ual method (CPR) [42, 43]. 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 [24] with the use of Algebraic Multigrid (AMG) [36] as a solver for the pressure subsystem. The second stage is usually an incomplete factorization method, such as incomplete LU factorization (ILU) [27].

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 advected similarly to the saturations. However, heat is also diffused through rock and fluids. 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 method. For example, the sequential fully implicit (SFI) method [21] 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 [45]. 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 [8] have been applied successfully to several problems, including reservoir simulation [19]. In the context of thermal simulation, construction of the grid hierarchy using both pressure and temperature variables has been discussed in [18]. However, the community's recent focus is again in coupling fluid 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 [34], 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. As an alternative, we also consider applying an unknown-based AMG method to the pressure-temperature subsystem. In a recent manuscript [11], promising results were shown for versions of CPTR for thermal-compositional-reactive flow.

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, as well as an 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, scalability, and robustness to strong cross-couplings. In section 5, we conclude with a brief discussion of the results and further work.

2. Problem statement. In this section, we begin by describing a coupled PDE system for thermal multiphase flow, followed by its discretization.

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

The volume fraction occupied by each phase α is called the saturation of that phase, denoted S_α. We have the following saturation constraint:

$$(2.1) \qquad \sum_\alpha S_\alpha = 1.$$

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

2.1.1. Conservation of mass. For each phase α, we have a conservation of mass equation of the following form:

$$(2.2) \qquad \phi \frac{\partial(S_\alpha \rho_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = f_\alpha \quad \text{in } \mathbb{R}_+ \times \Omega,$$

where ϕ is the porosity of the rock, ρ_α is the density of the fluid, $\mathbf{u}_\alpha$ is the fluid velocity, f_α is a source/sink term, and Ω is the spatial domain. We consider a fixed rock strata so that the porosity field ϕ is constant in time. We further assume that the velocity follows Darcy's law [12, 28], i.e.,

$$(2.3) \qquad \mathbf{u}_\alpha = -\mathbf{K} \frac{k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha \mathbf{g}),$$

where p is the pressure, $\mathbf{K}$ is a permeability tensor, μ_α is the viscosity, and $\mathbf{g}$ is gravitational acceleration. Note that we have ignored the effect of capillary pressures so that we only have one pressure p for all phases. Then (2.2) becomes

$$(2.4) \qquad \phi \frac{\partial(S_\alpha \rho_\alpha)}{\partial t} - \nabla \cdot \left(\rho_\alpha \frac{\mathbf{K} k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha \mathbf{g}) \right) = f_\alpha \quad \text{in } \mathbb{R}_+ \times \Omega.$$

For generality, we first assume mixed and Dirichlet boundary conditions

$$(2.5) \qquad -\frac{\mathbf{K} k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha \mathbf{g}) \cdot \mathbf{n} = g_{N,\alpha} \quad \text{on } \Gamma_N \quad \text{for all } \alpha,$$

$$(2.6) \qquad S_\alpha = g_{D,\alpha}, \text{ for all } \alpha, \quad p = g_D \quad \text{on } \Gamma_D,$$

where $g_{N,\alpha}$ is Neumann boundary data for the flow of phase α; g_D, and $g_{D,\alpha}$ are Dirichlet boundary data for pressure and saturation α, respectively; $\mathbf{n}$ is the unit outward normal vector on $\partial\Omega = \Gamma_N \cup \Gamma_D$; and $\Gamma_D \cap \Gamma_N = \varnothing$.

2.1.2. Conservation of energy. The above equations of fluid flow could be considered on their own as an isothermal model. We are interested in non-isothermal flow, and therefore, in addition to the mass equations, we add a conservation of energy equation for heat as well as a temperature variable T. (Note that enthalpy could also be considered as an independent variable.)

Similarly to the saturations, heat energy is transported by Darcy flow. In addition, however, temperature is also conducted through the porous rock and the different fluids. The conservation of energy is described by the following advection-diffusion

equation:

(2.7)
$$\frac{\partial}{\partial t}\left(\phi \sum_{\alpha} \rho_\alpha S_\alpha c_{v,\alpha} T + (1-\phi)\rho_r c_r T\right) + \nabla \cdot \sum_{\alpha} \rho_\alpha c_{v,\alpha} T \mathbf{u}_\alpha - \nabla \cdot (k_T \nabla T) = f_T$$
$$\text{in } \mathbb{R}_+ \times \Omega,$$

where $c_{v,\alpha}$ and c_r are the specific heat of the phase α and the rock, respectively; ρ_r is the density of the rock; k_T is the thermal conductivity field; and f_T is a source/sink term. In this model, the enthalpy of phase α and the rock are given by the simple relations $c_{v,\alpha}T$ and $c_r T$, respectively.

The conductivity depends on the properties of the rock and fluids. We consider the simple relation $k_T = (1-\phi)k_{T,r} + \phi \sum_\alpha S_\alpha k_{T,\alpha}$, where $k_{T,r}$ and $k_{T,\alpha}$ are the conductivities of the rock and the phase α, respectively. Assuming Darcy's law, we get

(2.8)
$$\frac{\partial}{\partial t}\left(\phi \sum_{\alpha} \rho_\alpha S_\alpha c_{v,\alpha} T + (1-\phi)\rho_r c_r T\right) - \nabla \cdot \sum_{\alpha} \rho_\alpha c_{v,\alpha} T \mathbf{K}\frac{k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha \mathbf{g})$$
$$- \nabla \cdot (k_T \nabla T) = f_T \quad \text{in } \mathbb{R}_+ \times \Omega.$$

We also first assume Neumann and Dirichlet boundary conditions

(2.9)
$$-\left(\sum_{\alpha} \rho_\alpha c_{v,\alpha} T \mathbf{K}\frac{k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha \mathbf{g}) + k_T \nabla T\right) \cdot \mathbf{n} = g_N^T \text{ on } \Gamma_N^T, \ T = g_D^T \text{ on } \Gamma_D^T,$$

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 $\Gamma_D^T \cap \Gamma_N^T = \varnothing$.

Note that $c_{v,\alpha}T$ in the first term of (2.7) is actually the internal energy of the fluid. For the numerical tests in 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 the relative permeability $k_{r\alpha}$, the density ρ_α, and the viscosity μ_α of phase α are empirically determined functions of saturations, pressure, and temperature. Our choices, which are representative of thermal reservoir simulation, are given in Appendix A.

Since the saturations satisfy the constraint (2.1), one of the saturation variables is explicitly replaced:

(2.10)
$$S_\beta = 1 - \sum_{\alpha \neq \beta} S_\alpha.$$

Thus, if the following system is for the flow of n_{phases} phases, it consists of $n_{\text{phases}} + 1$ equations and unknowns.

We are interested in solving the boundary value problem for p, T, S_α for all $\alpha \neq \beta$ satisfying (2.4), (2.8) together with boundary conditions (2.5), (2.6), (2.9), where initial conditions for p, T, and S_α are prescribed and we explicitly eliminate S_β via (2.10).

The numerical tests in this paper describe the flow of two phases: a heavy oil and water, denoted o and w, respectively.

2.1.4. Source/sink terms. Efficient recovery techniques for heavy oils include some way of reducing their viscosity, for example, by increasing the temperature inside the reservoir. This can be done by injecting hot fluids (for example, steam) or by electromagnetic heating [39].

We first consider the source/sink terms representing injection and production wells. These can be modeled by using point sources/sinks

$$f_\alpha(\mathbf{x}) = \sum_i q_{\alpha,\text{inj}}^i(p, T, S_o)\delta(\mathbf{x} - \mathbf{x}_{\text{inj}}^i)\rho_\alpha(p, T_{\text{inj}})$$

(2.11)
$$- \sum_j q_{\alpha,\text{prod}}^j(p, T, S_o)\delta(\mathbf{x} - \mathbf{x}_{\text{prod}}^j)\rho_\alpha(p, T),$$

$$f_T(\mathbf{x}) = \sum_\alpha \left(\sum_i q_{\alpha,\text{inj}}^i(p, T, S_o)\delta(\mathbf{x} - \mathbf{x}_{\text{inj}}^i)\rho_\alpha(p, T_{\text{inj}})c_{v,\alpha}T_{\text{inj}} \right.$$

(2.12)
$$\left. - \sum_j q_{\alpha,\text{prod}}^j(p, T, S_o)\delta(\mathbf{x} - \mathbf{x}_{\text{prod}}^j)\rho_\alpha(p, T)c_{v,\alpha}T \right),$$

where $\mathbf{x}_{\text{inj}}$ and $\mathbf{x}_{\text{prod}}$ represent the locations of injection and production wells, respectively; $\delta(\mathbf{x})$ is the Dirac delta function; and $q_{\alpha,\text{inj}}^i$ and $q_{\alpha,\text{prod}}^i$ are the injection and production rates of well i, respectively, for phase α.

For very simple cases, we use constant rates. In other cases, we consider a version of the commonly used Peaceman well model [6, 30]. Using such a model usually requires solving additional coupled equations for each well. Since we are using point source/sink wells, we do not have to solve any well equations to determine rates.

For heating source terms, instead of considering an electromagnetic model, we simply use the function

(2.13)
$$f_T = \sum_i U_{\text{heater}}(p, T)\delta(\mathbf{x} - \mathbf{x}_{\text{heater}}^i)(T_{\text{heater}} - T),$$

where $\mathbf{x}_{\text{heater}}$ represent the locations of heaters, U_{heater} is the heat transfer coefficient, and T_{heater} is the target heating temperature.

Details and parameters about well rates and heating sources terms can be found in section 2.1.5 of [34].

2.2. DG(0) discretization. Commercial reservoir simulators typically use first-order finite volume methods [26] for spatial discretization. These methods are stable, conservative, and easy to implement.

To have easy access to the different preconditioners provided by the PETSc library [2], we use the open-source finite element software Firedrake [31] to handle spatial discretization. The user must express the weak form of their PDE system in the Unified Form Language (UFL) [1]. Therefore, we want to formulate a weak version of the finite volume method used in reservoir simulation. To this end, we use a discontinuous Galerkin (DG) method [32] where the facet integrals are approximated as in reservoir simulation using two-point flux approximation. The details of the discretization are given in section 2.2 of [34], but we first provide the necessary notation.

Let $\mathcal{T} = \{E_i, i \in \mathcal{I}\}$ be a given partition of Ω into open element domains E_i such that the union of their closure is $\bar{\Omega}$, where $\mathcal{I}$ is a set of indices. The E_i are called cells. Since we want to emulate the finite volume method, we consider a piecewise constant approximation of our solution variables. The approximate solution will lie

in the approximation space $\mathcal{V}_h = \mathbb{P}^0_{\mathrm{DG}}$ with the basis of cellwise indicator functions, $\{\phi_i = \mathbb{1}_{E_i} \mid i \in \mathcal{I}\}$. The DG(0) approximation of a function u is $u_h = \sum_{i \in \mathcal{I}} u_i \phi_i$, with real coefficients u_i.

For a given ordering of the indices in $\mathcal{I}$, we denote by u^+ and u^- the limit values of u for two cells sharing an edge. On an edge, the jump of v is defined as $[v] = v^+ - v^-$. Denoting cell centers by h^+ and h^-, we get that $\|[h]\|$ is an edge quantity representing the Euclidean distance between the adjacent cell centers.

2.2.1. Semidiscrete problem. We now discretize the system (2.4)–(2.6), (2.8), (2.9) in space using the semidiscrete DG(0) formulation described in [34]. We consider 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 as follows: Find the approximation $(p, T, S_o) \in \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}}$ satisfying the conservation of water mass equation

$$
(2.14) \quad \int_\Omega \phi \frac{\partial(\rho_w(1 - S_o))}{\partial t} q \, \mathrm{d}x
$$
$$
+ \int_{\Gamma_{\mathrm{int}}} [q] \left(\{\!\{\mathbf{K}\}\!\} \frac{\rho_w^{\mathrm{up}} k_{rw}^{\mathrm{up}}}{\mu_w^{\mathrm{up}}} \left(\frac{[p]}{\|[h]\|} - \{\rho_w\}\mathbf{g} \cdot \mathbf{n}_e \right) \right) \mathrm{d}S - \int_\Omega f_w q \, \mathrm{d}x = 0,
$$

the conservation of energy equation

$$
(2.15)
$$
$$
\int_\Omega \phi \frac{\partial(c_{v,w}\rho_w(1 - S_o)T)}{\partial t} r \, \mathrm{d}x + \int_\Omega \phi \frac{\partial(c_{v,o}\rho_o S_o T)}{\partial t} r \, \mathrm{d}x + \int_\Omega (1 - \phi)\rho_r c_r \frac{\partial T}{\partial t} r \, \mathrm{d}x
$$
$$
+ \sum_{\alpha = o,w} \int_{\Gamma_{\mathrm{int}}} [r] \{\!\{\mathbf{K}\}\!\} k_{r\alpha}^{\mathrm{up}} c_{v,\alpha} \frac{\rho_\alpha^{\mathrm{up}}}{\mu_\alpha^{\mathrm{up}}} T^{\mathrm{up}} \left(\frac{[p]}{\|[h]\|} - \{\rho_\alpha\}\mathbf{g} \cdot \mathbf{n}_e \right) \mathrm{d}S
$$
$$
+ \int_{\Gamma_{\mathrm{int}}} [r] \{\!\{k_T\}\!\} \frac{[T]}{\|[h]\|} \, \mathrm{d}S - \int_\Omega f_T r \, \mathrm{d}x = 0
$$

and the conservation of oil mass equation

$$
\int_\Omega \phi \frac{\partial(\rho_o S_o)}{\partial t} s \, \mathrm{d}x + \int_{\Gamma_{\mathrm{int}}} [s] \left(\{\!\{\mathbf{K}\}\!\} \frac{\rho_o^{\mathrm{up}} k_{ro}^{\mathrm{up}}}{\mu_o^{\mathrm{up}}} \left(\frac{[p]}{\|[h]\|} - \{\rho_o\}\mathbf{g} \cdot \mathbf{n}_e \right) \right) \mathrm{d}S
$$
$$
(2.16) \hspace{7cm} - \int_\Omega f_o s \, \mathrm{d}x = 0,
$$

for all $(q, r, s) \in \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}}$. The unit outward-pointing normal of a cell is denoted by $\mathbf{n}_e$. The single brackets $\{\cdot\}$ denote the average across facets, while the double brackets $\{\!\{\cdot\}\!\}$ denote the harmonic average across facets. This choice of averaging is standard in reservoir simulation [14, 29].

Since the direction of Darcy flow is determined by the pressure gradient and gravity, upwind quantities carried by the velocity $\mathbf{u}_\alpha$, which, for a facet e shared by E_1 and E_2 and $\mathbf{n}_e$ pointing from E_1 to E_2, are given by

$$
(2.17) \hspace{3cm} (u_\alpha)^{\mathrm{up}} = \begin{cases} u \mid^e_{E_1} & \text{if } \dfrac{[p]}{\|[h]\|} - \{\rho_\alpha\}\mathbf{g} \cdot \mathbf{n}_e \geq 0, \\[2mm] u \mid^e_{E_2} & \text{if } \dfrac{[p]}{\|[h]\|} - \{\rho_\alpha\}\mathbf{g} \cdot \mathbf{n}_e < 0. \end{cases}
$$

The delta functions in the source/sink terms are approximated on coarse grids as

$$(2.18) \qquad \delta(x) = \begin{cases} 1/|E_i| & \text{if } x \in E_i, \\ 0 & \text{otherwise.} \end{cases}$$

For mesh refinement cases, the volume of the approximated delta functions are kept constant so that they will be defined over several cells.

2.2.2. Fully discretized problem. Time discretization is done using the fully implicit Euler method. We define three mappings, which are linear with respect with their last argument. The mappings are defined by the implicit Euler scheme applied to the left-hand side of the conservation equations (2.14)–(2.16). We denote the conservation of water mass mapping, the conservation of energy mapping, and conservation of oil mapping by $F_w(p, T, S_o; q)$, $F_e(p, T, S_o; r)$, and $F_o(p, T, S_o; s)$, respectively. Let

$$(2.19) \qquad F(p, T, S_o; q, r, s) := F_w(p, T, S_o; q) + F_e(p, T, S_o; r) + F_o(p, T, S_o; s),$$

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 $(p^{n+1}, T^{n+1}, S_o^{n+1}) \in \mathbb{P}_{\mathrm{DG}}^0 \times \mathbb{P}_{\mathrm{DG}}^0 \times \mathbb{P}_{\mathrm{DG}}^0$ such that

$$(2.20) \qquad F(p^{n+1}, T^{n+1}, S_o^{n+1}; q, r, s) = 0 \quad \text{for all } (q, r, s) \in \mathbb{P}_{\mathrm{DG}}^0 \times \mathbb{P}_{\mathrm{DG}}^0 \times \mathbb{P}_{\mathrm{DG}}^0.$$

Consider the equivalent problem: Find $(p^{n+1}, T^{n+1}, S_o^{n+1}) \in \mathbb{P}_{\mathrm{DG}}^0 \times \mathbb{P}_{\mathrm{DG}}^0 \times \mathbb{P}_{\mathrm{DG}}^0$ such that

$$(2.21) \qquad F^*(p^{n+1}, T^{n+1}, S_o^{n+1}; q, r, s) = 0 \quad \text{for all } (q, r, s) \in \mathbb{P}_{\mathrm{DG}}^0 \times \mathbb{P}_{\mathrm{DG}}^0 \times \mathbb{P}_{\mathrm{DG}}^0,$$

where

$$(2.22) \qquad F^*(p, T, S_o; q, r, s) := F_p(p, T, S_o; q) + F_e(p, T, S_o; r) + F_o(p, T, S_o; s)$$

is F, where F_w has been replaced by the following "pressure equation" mapping:

$$(2.23) \qquad F_p(p, T, S_o; q) := c_{v,w} F_w(p, T, S_o; q) + c_{v,o} F_o(p, T, S_o; q).$$

Recall that $c_{v,w}$ and $c_{v,o}$ are the specific heat coefficients of the water and oil phases, respectively.

3. Solution algorithms. The system (2.21) can be written as a nonlinear system of equations for the real coefficients p_i, T_i, and $S_{o,i}$ of the DG(0) functions p^{n+1}, T^{n+1}, and S_o^{n+1}, respectively. Let x be the vector of these real coefficients and $\mathcal{F}$ the function such that $\mathcal{F}(x) = 0$ is equivalent to (2.21). Solving this equation using Newton's method requires the solution of the following linearized system at each iteration $k + 1$:

$$(3.1) \qquad \frac{\partial \mathcal{F}}{\partial x}\Big|_{x=x_k} (x_{k+1} - x_k) = -\mathcal{F}(x_k).$$

This can be written as a block system of the form

$$(3.2) \qquad A\delta x = \begin{bmatrix} A_{pp} & A_{pT} & A_{ps} \\ A_{Tp} & A_{TT} & A_{Ts} \\ A_{sp} & A_{sT} & A_{ss} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta T \\ \delta s \end{bmatrix} = \begin{bmatrix} b_p \\ b_T \\ b_s \end{bmatrix} = b,$$

where $\delta x = x_{k+1} - x_k$ and $b = -\mathcal{F}(x_k)$, in the desired ordering. Looking back at the strong form of (2.4)–(2.8), we see that the blocks of the pressure-temperature submatrix are the discrete versions of

$$(3.3) \qquad A_{pp} \sim \phi \frac{1}{\Delta t} \sum_\alpha c_{v,\alpha} S_\alpha (\rho_\alpha)_p + \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p - \sum_\alpha c_{v,\alpha} (f_\alpha)_p,$$

$$(3.4) \qquad A_{pT} \sim \phi \frac{1}{\Delta t} \sum_\alpha c_{v,\alpha} S_\alpha (\rho_\alpha)_T + \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T - \sum_\alpha c_{v,\alpha} (f_\alpha)_T,$$

$$(3.5) \qquad A_{Tp} \sim \phi \frac{1}{\Delta t} \sum_\alpha c_{v,\alpha} S_\alpha (\rho_\alpha)_p T + \sum_\alpha \nabla \cdot (c_{v,\alpha} T (\rho_\alpha \mathbf{u}_\alpha)_p) - (f_T)_p,$$

$$(3.6) \quad A_{TT} \sim \frac{\phi}{\Delta t} \sum_\alpha c_{v,\alpha} (\rho_\alpha + (\rho_\alpha)_T T) + (1-\phi) \frac{\rho_r c_r}{\Delta t} + \sum_\alpha \nabla \cdot (c_{v,\alpha} \rho_\alpha \mathbf{u}_\alpha)$$
$$+ \sum_\alpha \nabla \cdot (c_{v,\alpha} T (\rho_\alpha \mathbf{u}_\alpha)_T) - \nabla \cdot (k_T \nabla) - (f_T)_T,$$

where

$$(3.7) \qquad (\rho_\alpha \mathbf{u}_\alpha)_p = -\frac{\mathbf{K} k_{r\alpha}}{\mu_\alpha} \left(\rho_\alpha (\nabla - (\rho_\alpha)_p \mathbf{g}) + (\rho_\alpha)_p (\nabla p - \rho_\alpha \mathbf{g}) \right)$$

and

$$(3.8) \qquad (\rho_\alpha \mathbf{u}_\alpha)_T = -\mathbf{K} k_{r\alpha} \left[\left(\frac{\rho_\alpha}{\mu_\alpha} \right)_T (\nabla p - \rho_\alpha \mathbf{g}) - \frac{\rho_\alpha}{\mu_\alpha} (\rho_\alpha)_T \mathbf{g} \right].$$

Here, the coefficients in (3.3)–(3.8) are evaluated at the previous Newton iterate $(p_k, T_k, S_{o,k})$, and $(.)_p$ and $(.)_T$ denote the partial derivatives with respect to p and T, respectively.

We seek to solve the resulting linearized system (3.2) using iterative methods [37]. We will consider the generalized minimal residual method (GMRES) [38], a widely used Krylov subspace method suitable for nonsymmetric linear systems.

For complicated 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 that take into account the specific structure of (3.2). Important ingredients for the preconditioning techniques are incomplete LU factorization (ILU) [27, 37] and AMG [36, 41].

Two-stage preconditioners. The methods described in sections 3.1 and 3.2 are multiplicative two-stage preconditioners, and so we first provide here a generic definition. Let M_1 and M_2 be two preconditioners approximating the system matrix A, for which we know a way of applying their (generally approximate) inverses M_1^{-1} and M_2^{-1}. Applying the resulting two-stage preconditioner for the system $Ax = b$ can be done as follows:

1. apply the first preconditioner: $x_1 = M_1^{-1} b$;
2. compute the new residual: $b_1 = b - Ax_1$;
3. apply the second preconditioner: $x = M_2^{-1} b_1 + x_1$.

The action of the two-stage preconditioner can also be written in the explicit form

$$(3.9) \qquad\qquad M^{-1} = M_2^{-1}(I - AM_1^{-1}) + M_1^{-1}.$$

3.1. CPR. The standard preconditioner for multiphase flow in porous media is the CPR [42]. 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 (3.2) is thus rearranged in the following form:

$$(3.10) \qquad A\delta x = \begin{bmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta s \end{bmatrix} = \begin{bmatrix} b_p \\ b_s \end{bmatrix} = b,$$

where the temperature is grouped with the saturations.

The CPR preconditioner is a two-stage preconditioner where the preconditioner M_1 solves a restricted system for the pressure

$$(3.11) \qquad M_1^{-1} \approx \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & 0 \end{bmatrix},$$

where A_{pp}^{-1} is generally approximated using an AMG V-cycle.

If the pressure solution given in the first stage is accurate, only transport problems for the saturations and temperature remain (assuming that heat diffusion is not significant). Therefore, a simple method such as ILU(0) or ILU(1) is adequate for the second-stage preconditioner M_2.

Applying CPR to the right-hand side $b = [\, b_p \;\; b_s \,]^\top$ can be done as follows:
1. solve the pressure subsystem: $A_{pp}x_p = b_p$;

2. compute the new residual:

$$\tilde{b} = \begin{bmatrix} b_p \\ b_s \end{bmatrix} - \begin{bmatrix} A_{pp} \\ A_{ps} \end{bmatrix} x_p;$$

3. precondition and correct:

$$\begin{bmatrix} \delta_p \\ \delta_s \end{bmatrix} = M_2^{-1}\tilde{b} + \begin{bmatrix} x_p \\ 0 \end{bmatrix}.$$

Thus, the update provided by CPR is

$$(3.12) \qquad \delta = M_2^{-1} \left(I - (A - M_2) \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right) b.$$

Note that an approximation of the pressure equation $A_{pp}\delta p + A_{ps}\delta s = b_s$ in (3.10) is performed in the first stage of CPR, where the coupling with secondary unknowns A_{ps} is ignored. To reduce the coupling between the pressure equation and the secondary unknowns, decoupling operators are often used with CPR. Applied to the linear system (3.10), decoupling operators have the form

$$(3.13) \qquad G = \begin{bmatrix} I & -D \\ 0 & I \end{bmatrix},$$

where D is a block diagonal matrix (with wide rectangular blocks if there are multiple nonpressure unknowns). Decoupling operators are often applied to the linear system as a preprocessing step, i.e., a left preconditioner. Alternatively, the decoupling operator G can be used within the first stage of CPR as follows:

$$(3.14) \qquad M_1^{-1} = R_p^\top (R_p G A R_p^\top)^{-1} R_p G,$$

where R_p is the pressure restriction operator such that $R_p A R_p^\top = A_{pp}$.

The most common approximations are quasi-IMPES (QI) and true-IMPES (TI) [24, 25, 40]. These approximations are $D_{\mathrm{QI}} = \mathrm{diag}(A_{ps}) \, \mathrm{diag}(A_{ss})^{-1}$ and $D_{\mathrm{TI}} = \mathrm{colsum}(A_{ps})\mathrm{colsum}(A_{ss})^{-1}$, respectively. Here, diag(.) and colsum(.) return block diagonal matrices. The colsum(.) operator returns the sum of the entries in each column (summing the columns of block matrices independently). If the matrices are nonsquare, the diag(.) and colsum(.) operators return block diagonal matrices with nonsquare blocks.

When using decoupling operators, the first stage of CPR consists of solving the approximate Schur complement system involving $S_p = A_{pp} - D A_{sp}$ instead of the original pressure block A_{pp}. While the elliptic-like properties of A_{pp} are clear from the PDEs, it is less so for S_p. Properties amenable to the performance of AMG (e.g., M-matrix properties) are nearly guaranteed in simpler multiphase cases [18], but compositional and thermal cases have no such guarantee.

In our case, we also perform a weighted sum of the mass equations in (2.23). 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 α and 0 for the energy conservation equation.

For our test cases, using decoupling operators can reduce the number of GMRES iterations needed. TI is observed to be superior to QI, which sometimes increases the number of GMRES iterations.

3.2. 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 [34]. We call this method CPTR.

Let the pressure-temperature submatrix

$$(3.15) \qquad A_{00} = \begin{bmatrix} A_{pp} & A_{pT} \\ A_{Tp} & A_{TT} \end{bmatrix} \quad \text{so that} \quad A = \begin{bmatrix} A_{00} & A_{0s} \\ A_{s0} & A_{ss} \end{bmatrix}.$$

For CPTR, The first stage preconditioner M_1 is given by

$$(3.16) \qquad M_1^{-1} = \begin{bmatrix} A_{00}^{-1} & 0 \\ 0 & 0 \end{bmatrix},$$

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

In our case, we consider two approaches to approximate A_{00}^{-1}: an approximation given by a Schur complement factorization detailed in section 3.3 and an unknown-based AMG method discussed in section 3.4.

Similarly to CPR, we use ILU for the second-stage preconditioner M_2.

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: $\tilde{b} = \begin{bmatrix} b_0 \\ b_s \end{bmatrix} - \begin{bmatrix} A_{00} \\ A_{0s} \end{bmatrix} x_0$;

3. precondition and correct: $\begin{bmatrix} \delta_0 \\ \delta_s \end{bmatrix} = M_2^{-1}\tilde{b} + \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$.

Thus,

$$(3.17) \qquad \delta = M_2^{-1} \left(I - (A - M_2) \begin{bmatrix} A_{00}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right) b.$$

Similarly to CPR, an approximation is made in the first stage of CPTR, where the saturation coupling block A_{0s} is ignored. A first attempt at reducing this decoupling error is to extend the idea of decoupling operators to the pressure-temperature system. In that case, the first stage of CPTR consists in solving the system with the Schur complement approximation $S_0 = A_{00} - DA_{s0}$ for D a block diagonal matrix of appropriate dimensions. The extensions of QI and TI are $D_{\mathrm{QI}} = \mathrm{diag}(A_{0s})\mathrm{diag}(A_{ss})^{-1}$ and $D_{\mathrm{TI}} = \mathrm{colsum}(A_{0s})\mathrm{colsum}(A_{ss})^{-1}$, respectively.

3.3. Block preconditioner for the pressure-temperature subsystem. In this section, we present a choice for the pressure-temperature solver A_{00}^{-1}. Consider the following decomposition of the pressure-temperature submatrix:

$$(3.18) \qquad A_{00} = \begin{bmatrix} I & 0 \\ A_{Tp}A_{pp}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{pp} & 0 \\ 0 & S_T \end{bmatrix} \begin{bmatrix} I & A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix},$$

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

$$(3.19) \qquad A_{00}^{-1} = \begin{bmatrix} I & -A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & S_T^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{Tp}A_{pp}^{-1} & I \end{bmatrix}.$$

In general, the Schur complement S_T is a dense matrix, even if A_{00} is sparse. A common preconditioning technique for PDE systems is to use the blocks of a factorization like (3.18) but with a sparse approximation of the Schur complement [13]. With an appropriate Schur complement approximation $\tilde{S}_T$, our block preconditioner for A_{00} is given by

$$(3.20) \qquad \tilde{A}_{00}^{-1} = \begin{bmatrix} I & -A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & \tilde{S}_T^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{Tp}A_{pp}^{-1} & I. \end{bmatrix}.$$

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

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

The CPTR method with this pressure-temperature solver 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 $\tilde{S}_T^{-1}$. One could also use block triangular versions of the block preconditioner, i.e., ignoring the left or right factor in (3.20).

3.3.1. Schur complement approximation. The two simple sparse Schur complement approximations $\tilde{S}_{A_{TT}} = A_{TT}$ and $\tilde{S}_{\mathrm{diag}} = A_{TT} - A_{Tp}\mathrm{diag}\,(A_{pp})^{-1}\,A_{pT}$ were shown to perform poorly in our previous work for the single-phase case [34]. Here, we extend the Schur complement approximation presented in that work to the multiphase case. This Schur complement approximation requires the construction of the pressure equation (2.23).

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

$$(3.21) \qquad \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = 0 \quad \text{in } \Omega,$$

$$(3.22) \qquad \sum_\alpha \nabla \cdot (\rho_\alpha c_{v,\alpha} T \mathbf{u}_\alpha) - \nabla \cdot (\mathbf{k}_T \nabla T) = 0 \quad \text{in } \Omega,$$

where $\mathbf{u}_\alpha$ is given by (2.3) and we have homogeneous Neumann boundary conditions. Note that (3.21) is the pressure equation obtained from the weighting (2.23). Here, we consider the linearized pressure-temperature subsystem in an infinite-dimensional setting. Applying a Newton method to (3.21)–(3.22), we obtain a block matrix of the form (3.15), where the blocks are

$$(3.23) \qquad A_{pp} = \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \,, \qquad A_{pT} = \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T \,,$$

$$(3.24) \quad A_{Tp} = \sum_\alpha \nabla \cdot (\rho_\alpha c_{v,\alpha} T \mathbf{u}_\alpha)_p = \sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_p + \sum_\alpha c_{v,\alpha} T \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \,,$$

$$A_{TT} = \sum_\alpha \nabla \cdot (\rho_\alpha c_{v,\alpha} T \mathbf{u}_\alpha)_T - \nabla \cdot (\mathbf{k}_T \nabla)$$

$$= \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) + \sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_T$$

$$(3.25) \qquad + \sum_\alpha c_{v,\alpha} T \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T - \nabla \cdot (\mathbf{k}_T \nabla),$$

where the product rule was used for the divergence operator in (3.24) and (3.25). Then, the second term of the Schur complement (which corresponds to the Poincaré-Steklov operator in the infinite-dimensional setting) becomes

$$A_{Tp} A_{pp}^{-1} A_{pT} = \left(\sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_p + \sum_\alpha c_{v,\alpha} T \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \right)$$

$$\left(\sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \right)^{-1} \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T$$

$$= T \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T + \sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_p$$

$$\left(\sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \right)^{-1} \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T$$

$$= T \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T + \nabla T \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p$$

$$(3.26) \qquad \left(\nabla \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p \right)^{-1} \nabla \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_T \,.$$

Taking the difference between (3.25) and (3.26), we obtain the Schur complement, $A_{TT} - A_{Tp} A_{pp}^{-1} A_{pT}$. We notice that the term $T \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T$ appears in both

(3.25) and (3.26) and thus cancels. This is due to the construction of the pressure equation (2.23). We are left with

$$(3.27) \quad S_T = \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) - \nabla \cdot (\mathbf{k}_T \nabla) + \sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_T$$
$$- \nabla T \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p \left(\nabla \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p \right)^{-1} \nabla \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_T .$$

Since one of the terms coming from the linearization has canceled, we investigate if it is possible that the last two terms in (3.27) can cancel under reasonable assumptions. Consider the following operator taken from the last term of (3.27):

$$(3.28) \qquad \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p \left(\nabla \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p \right)^{-1} \nabla.$$

We now make the same approximations as in [34], which assumes that ρ_α is constant with respect to p and that we use a two-point flux approximation for the facet integrals. This allows us to replace the operator (3.28) by the identity, giving us the following simplified and more practical Schur complement approximation:

$$(3.29) \qquad \tilde{S}_T = \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T - \nabla \cdot (\mathbf{k}_T \nabla).$$

While the assumption that ρ_α is close to being a constant with respect to p is appropriate for liquid oil and water, it may be less applicable in the case of gases. However, early investigations reveal that increasing the compressibility of the fluid does not reduce the effectiveness of the preconditioner.

The extension to the time-dependent case and the addition of source/sink terms does not differ from the single-phase case, so we refer the reader to [34] 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_{heaters} is the sum of delta functions for the locations of heaters. The Schur complement approximation is given by

$$(3.30) \quad \tilde{S}_T = \phi \sum_\alpha \frac{c_{v,\alpha} \rho_\alpha}{\Delta t} + (1 - \phi) \frac{\rho_r c_r}{\Delta t} + \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T - \nabla \cdot (k_T \nabla)$$
$$+ U D_{\text{heaters}} - \sum_\alpha c_{v,\alpha} f_{\alpha,\text{prod}}.$$

The discretized version of this operator can be obtained from the conservation of energy mapping F_e from section 2.2.2 by removing the terms that only depend on the previous time-step and by evaluating the coefficients with the values of the previous Newton iterate. We get the following bilinear operator:

$$(3.31) \quad S_e(\delta T, r) := \sum_\alpha \int_\Omega \phi c_{v,\alpha} \frac{S_\alpha \rho_\alpha \delta T}{\Delta t} r \, dx + \int_\Omega (1 - \phi) \rho_r c_r \frac{\delta T}{\Delta t} r \, dx$$
$$+ \sum_\alpha \int_{\Gamma_{\text{int}}} [r] \{\!\{ \mathbf{K} \}\!\} (k_{r\alpha})^{\text{up}} c_{v,\alpha} \frac{(\rho_\alpha)^{\text{up}}}{(\mu_\alpha)^{\text{up}}} (\delta T)^{\text{up}} \left(\frac{[p]}{\|[h]\|} - \{\rho_\alpha\} \mathbf{g} \cdot \mathbf{n}_e \right) dS$$
$$+ \int_{\Gamma_{\text{int}}} [r] \{\!\{ k_T \}\!\} \frac{[\delta T]}{\|[h]\|} \, dS + \int_\Omega \left(- \sum_\alpha c_{v,\alpha} f_{\alpha,\text{prod}} + U D_{\text{heaters}} \right) \delta T \, dx.$$

Since this Schur complement approximation relies heavily on the PDE structure, it cannot be combined with decoupling operators from section 3.2 in an obvious way. The following method can be used as an alternative to the block preconditioner.

3.4. AMG for the pressure-temperature subsystem. AMG methods for systems of PDEs are an obvious alternative for the first stage of CPTR. See [8] for a detailed description and a general framework for such methods. Efficient implementations of such AMG methods include BoomerAMG [20] and multigrid reduction (MGR) from the hypre library [15] and Smoothed Aggregation from the ML package [16]. For example, BoomerAMG was effective in some diffusion-dominated two-phase flow problems [4], and MGR had some success with multiphase flow problems [44, 5]. Such approaches are more easily generalizable than the Schur complement approach since they usually only rely on algebraic information from the system matrix.

The unknown-based strategy is to treat the variables corresponding to the same unknown separately. For a matrix with an unknownwise ordering, classical AMG coarsening and interpolation are applied to the individual diagonal blocks. Computationally cheap and easy to implement, the unknown-based approach will perform well if the cross-coupling between unknowns is not too strong and the diagonal blocks are amenable for the application of classical AMG.

With a typical finite volume scheme, degrees of freedom associated with the different variables are located at the same points. This means that one could consider using a nodal/point-based AMG method (initially designed for linear elasticity problems [35]). Instead of considering all degrees of freedom individually, they are grouped in a vector at each "point." The construction of the coarse problems uses similar heuristics to the scalar equivalent but applied to the "point" vectors and resulting blocks in the system matrix. However, the performance of coarsening strategies for the nodal approach is problem specific. Several strategies are discussed in [8].

Naively applying a nodal or unknown-based AMG method to the full reservoir simulation system does not work in the common case where the equations are hyperbolic with respect to the saturations. Substantial work is needed to construct meaningful coarse problems when applying AMG to the full system [9, 19]. The proposed methods include unknown-based AMG approaches and decoupling operators focused on maintaining AMG-friendly properties.

However, the equations are elliptic with respect to pressure, and the energy equation is parabolic with respect to temperature. We will consider an unknown-based AMG for the pressure-temperature subsystem as an alternative to the Schur complement approach.

Since AMG methods for systems of PDEs are purely algebraic, they can be applied to the modified pressure-temperature matrix S_0 obtained from decoupling operators mentioned in section 3.2. It is unclear, however, if the properties of S_0 are guaranteed to be amenable to the application of AMG.

3.5. Discussion on multistage preconditioners. The first stage of CPR is given by the approximate solution of $A_{pp}\widetilde{\delta p} = b_p$, which is an approximation of the pressure equation from (3.2):

$$(3.32) \qquad\qquad A_{pp}\delta p + A_{pT}\delta T + A_{ps}\delta s = b_p.$$

If the coupling blocks A_{pT} and A_{ps} are significant, the approximate pressure $\widetilde{\delta p}$ might not be representative. This could be countered by reducing the coupling between the pressure equation and the nonpressure variables. 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 A_{ps} and A_{Ts}. We will consider the use of decoupling operators for both CPR and CPTR. However, for CPTR, this cannot be done when using our block preconditioner as the first-stage solver.

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 (ILU on the full system followed by the restricted solve). Then the approximate solution of the fully coupled system (through ILU) acts as a decoupling operator. This idea has been used, for example, in [4] for the isothermal situation.

For reordered CPTR, investigations revealed that during GMRES, the residual of the energy equation was being reduced faster than that of the pressure equation and oil mass conservation equation. Therefore, to obtain an adequate solution, the equations must be scaled such that they are of similar magnitude.

In our case, we multiply the pressure equation by a reference temperature and the oil mass conservation equation by the product of the reference temperature and a reference-specific heat coefficient. Thus, we modify F^* from (2.22) to
(3.33)
$$F^*(p, T, S_o; q, r, s) := T_{\text{ref}} F_p(p, T, S_o; q) + F_e(p, T, S_o; r) + c_{\text{ref}} T_{\text{ref}} F_o(p, T, S_o; s).$$

This scaling has little effect on the number of iterations for CPR and the original ordering of CPTR. However, it is necessary for reordered CPTR, which then performs similarly to the original CPTR. Thus, in the next section, we will only consider versions of CPR and CPTR with the original ordering of the stages. Nonetheless, we will use the scaling (3.33) by default.

As mentioned above, for reordered 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) improves the solution of the restricted pressure-temperature subsystem. However, the residual of the other equations is not reduced as fast for this method. We will investigate numerically in the next section if the use of decoupling operators can reduce the number of iterations for CPTR.

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. We perform numerical experiments to evaluate the algorithmic performance of CPR and CPTR. The code used for these experiments can be found on GitHub.[1] This code is implemented using the open-source finite element software Firedrake [23, 31]. For solvers, Firedrake interfaces with the PETSc library [2], allowing for efficient and parallel computations. Our Schur complement approximation and the two-stage preconditioners are implemented through Firedrake's Python interface, including proof-of-concept implementations for the decoupling operators. Simple but less efficient versions of the two-stage preconditioners can also be implemented by simply providing PETSc options.

---

[1]https://github.com/tlroy/thermalporous.

TABLE 1
Physical parameters for test cases.

Initial pressure	4.1369×10^7 Pa
Conductivity of oil	0.15 W m^{-1} K^{-1}
Conductivity of water	0.6005638 W m^{-1} K^{-1}
Conductivity of rock	1.7295772056 W m^{-1} K^{-1}
Specific heat of oil	2093.4 J K^{-1} kg^{-1}
Specific heat of water	4181.3 J K^{-1} kg^{-1}
Specific heat of rock	920 J K^{-1} kg^{-1}
Density of rock	2650 kg m^{-1}

The first stage of CPTR is given by our Schur complement approximation (3.31) or by an unknown-based AMG. Unless stated otherwise, for both CPR and CPTR, an AMG V-cycle is used to approximate the inverse for the pressure block A_{pp}, the approximate Schur complement, and the temperature block A_{TT}. For both scalar and unknown-based AMG, BoomerAMG [20] from the hypre library [15] is used 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. BoomerAMG is very efficient in parallel. When direct LU factorization is used instead of AMG, we use PETSc's default implementation. For the second stage of CPR and CPTR, ILU(k) is used as provided in PETSc. Unless stated that we use ILU(1), ILU(0) is our default. The parallel version of ILU is block Jacobi, where ILU(k) is used for each block (the blocks are determined when Firedrake assigns different parts of the domain to different processors).

For the nonlinear solver, we use Newton's method with line search and for the linear solver right-preconditioned GMRES [38]. For Newton's method, the convergence tolerance for both the relative function norm and relative step size norm is set to 10^{-8}. For GMRES, the convergence tolerance for the relative residual norm is set to 10^{-8}.

For all cases, we consider oil and water with densities and viscosities as described in Appendix A. The other physical parameters are shown in Table 1. These parameters are representative of those used in commercial reservoir simulators. The pressures are taken from the SPE10 test case [7]. We also only consider homogeneous Neumann boundary conditions.

For the scaling parameters in (3.33), we choose $T_{\text{ref}} = T_{\text{prod}}$ and $c_{\text{ref}} = S_o c_{v,o} + (1 - S_o)c_{v,w}$, 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.

Computational cost. We note that for the first stage of CPTR, applying the block preconditioner from section 3.3 should be slightly more than three times the cost of applying the first stage of CPR, while the unknown-based AMG should be around two times the cost. 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. Looking at the CPU timings for the serial case given in Table 6 (a problem with 52,728 degrees of freedom), applying the first stage of CPR takes an average of 0.0088 s, while the first stage of CPTR with the block preconditioner takes an average of 0.033 s and 0.017 s for unknown-based AMG. Applying the second stage of CPR/CPTR with ILU(0) takes an average of 0.02 s and 0.034 s with ILU(1). These timings were performed on a Lenovo ThinkCentre M920q with an Intel Core i5-8500T CPU at 2.10 GHz. In terms of memory requirements, the first stage of CPTR requires two different AMG hierarchies. How this compares to the memory

requirements of the second stage will depend on the size of the blocks and fill-in used for ILU. Note that since we implement our Schur complement approximation in Python, the implementation of the block preconditioner is not as optimized as that of unknown-based AMG in BoomerAMG. Additionally, the full factorization (3.30) used here may not be needed in all cases. Dropping one of the factors makes the cost of applying the block preconditioner slightly more than two times the cost of the first stage of CPR, or slightly more than unknown-based AMG.

4.1. Homogeneous test cases. We compare the performance of several methods for problems with homogeneous permeability. We begin by a brief mesh refinement study for two-dimensional (2D) cases. Then we look at both weak and strong scaling for three-dimensional (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 very relevant in the context of reservoir simulation, for example, around wells or other features of interest.

4.1.1. Mesh refinement study. We perform a mesh refinement study for 2D test cases. The domain is 50×50 m with an $N \times N$ grid. Starting at $N = 20$, we double N until we reach $N = 640$. Since this is a 2D case, we do not include gravity. For the isotropic cases, the permeability is 3×10^{-13} m^2, and the porosity is 0.2. For the anisotropic case, the permeability in the x-direction is 3×10^{-13} m^2 and in the y-direction 3×10^{-10} m^2.

We consider three cases: one with heaters and two with wells (isotropic and anisotropic). For each case, we have six source/sink terms with three located on two opposing sides of the square domain. In the heating case, these terms are all heaters, while for the well cases, one side has injection wells and the other production wells. For the well cases, 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 oil saturation for both cases is set to $S_o = 0.9$. We then take two time-steps of 10 days.

For CPTR, we will compare the use of the following first-stage solvers: block preconditioner (CPTR-Block(AMG)), unknownwise AMG (CPTR-uAMG), and unknownwise AMG with the True-IMPES decoupling operator (CPTR-uAMG(TI)). ILU(0) is used in the second stage. For CPR, we will compare the use of nondecoupled CPR with ILU(0) (CPR-AMG) and with ILU(1) (CPR-AMG-ILU(1)), as well as CPR with True-IMPES and ILU(0) (CPR-AMG(TI)).

For each case, we calculate the average number of linear iterations per nonlinear iteration and computational time, obtained with an Intel Xeon Gold 6140 CPU at 2.30 GHz. We do not report timings for the methods using TI decoupling since the current implementation is not efficient (TI is very cheap if implemented appropriately). The results are shown for the isotropic well case, the heater case, and the anisotropic well case in Tables 2, 3, and 4, respectively. In all 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 does not increase much for CPTR

TABLE 2
Mesh refinement: 2D isotropic well case. Average linear iterations per nonlinear iteration (computational time (s)).

Method/N	20	40	80	160	320	640
CPR-AMG	5.29	6.29	9.14 (2.3)	15.5 (9)	31.5 (67)	65.7 (580)
CPR-AMG(TI)	4.86	5.43	8.29	14.7	30.7	65.1
CPR-AMG-ILU(1)	5.29	5.43	6.43 (2.3)	10 (9)	20 (62)	40.6 (472)
CPTR-Block(AMG)	5.29	5.29	5.43 (2.9)	5.64 (9.5)	6.53 (55)	7.7 (310)
CPTR-uAMG	5.29	5.29	6.29 (1.4)	6.82 (6.8)	8.24 (44)	10 (258)
CPTR-uAMG(TI)	4.29	5.14	5.29	5.82	7.35	9.26

TABLE 3
Mesh refinement: 2D heater case. Average linear iterations per nonlinear iteration (computational time (s)).

Method/N	20	40	80	160	320	640
CPR-AMG	5.33	5.5	8.5 (1.6)	14 (4.1)	25.8 (18)	49.3 (81)
CPR-AMG(TI)	5.14	5.29	7.83	14.4	25	48.8
CPR-AMG-ILU(1)	4.67	5.17	5.83 (1.6)	9 (4.2)	16 (17)	30.8 (68)
CPTR-Block(AMG)	4.83	5.17	5.17 (2)	4.6 (4.5)	5 (15)	4.75 (45)
CPTR-uAMG	4.67	5	5 (1.1)	5 (3)	5 (12)	4.75 (34)
CPTR-uAMG(TI)	4.86	4.71	4.5	4.6	4.6	4.25

TABLE 4
Mesh refinement: 2D anisotropic well case. Average linear iterations per nonlinear iteration (computational time (s)).

Method/N	20	40	80	160	320	640
CPR-AMG	5	6	9 (2.6)	17 (10)	32.3 (73)	67.2 (650)
CPR-AMG(TI)	4.29	5.71	8.25	16.1	31.8	65.3
CPR-AMG-ILU(1)	4.71	5.43	6.5 (2.6)	11.1 (10)	20.4 (67)	40.9 (514)
CPTR-Block(AMG)	4.43	5.14	5.75 (3.2)	6.92 (11)	7.17 (60)	7.46 (319)
CPTR-uAMG	4.86	5.43	6.25 (1.7)	8.08 (8.6)	10.1 (55)	12.1 (333)
CPTR-uAMG(TI)	4.14	5	5.5	7.33	9.44	11.6

or even decreases in the heater case. In the well cases, we observe that this increase is least significant for CPTR-Block(AMG). In the heater case, CPTR-uAMG(TI) has the smallest number of iterations. Note that the number of Newton steps is lower for the heater case and does not increase much as the mesh is refined.

It is important to note that a different variable ordering is used for CPTR-uAMG to allow the use of uAMG on the pressure-temperature system. This results in smaller computational times compared to the ordering used for CPR and CPTR-Block. This is not due to uAMG since we observe the same trend when LU is used as the first-stage solver for CPTR. Nevertheless, we observe that CPTR-Block(AMG) is only faster than CPR-AMG and CPR-AMG-ILU(1) for $N = 320, 640$. For both well cases at $N = 640$, the cost per iteration of CPTR-Block(AMG), CPTR-uAMG, and CPR-AMG-ILU(1) is around 4.5, 3, and 1.3 times more than CPR-AMG, respectively.

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 on AMG. Hence, CPTR tackles heat diffusion appropriately on fine meshes.

TABLE 5
Weak scaling: 3D well case, $S_o = 0.99$. Average linear iterations per nonlinear iteration.

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

TABLE 6
Weak Scaling: 3D well case, $S_o = 1$. Average linear iterations per nonlinear iteration.

Method/Num. Proc.	1	2	4	8	16
CPR-AMG	14.1	16	17.3	20.6	23.5
CPTR-Block(AMG)	14.1	15.9	17.3	20.5	23.4
CPR-AMG-ILU(1)	11.7	14	15.3	17.7	19.3

4.1.2. Weak scaling. For weak scaling, the performance of the methods is evaluated as we increase the number of processors and problem size simultaneously. The domain is a $50 \times 50 \times 50$-m box, discretized with an $N \times N \times N$ grid. The permeability is constant at $3 \times 10^{-13} \mathrm{m}^2$, and the porosity is 0.2. For the number of processors 1, 2, 4, 8, and 16, we have $N = 26, 33, 41, 52,$ and 65, respectively. This results in around $50,000$ degrees of freedom per processor.

For the cases here, CPR performs similarly with or without True-IMPES, and the different variants of CPTR also perform similarly. We therefore only show results for CPR-AMG and CPTR-Block(AMG). Computational timings for similar test cases are shown in [33], where the different versions of CPTR outperform CPR for large enough problems.

We consider two cases with wells. For each case, we have 21 source/sink terms both near the top and bottom of the domain. The top terms are injection wells and the bottom terms production wells. The production/injection rates are constant at $q = 10^{-7} \mathrm{\ m}^3 \mathrm{\,s}^{-1}$. Water is injected at a temperature of 373.15 K.

The initial saturation is set to $S_o = 0.99$ or $S_o = 1$. For each case, we take five 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 case with $S_o = 0.99$ and $\Delta t = 4$ days for the case with $S_o = 1$.

The results for the case with $S_o = 0.99$ are shown in Table 5. 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 fewer 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 6, the results are significantly different for the well case with $S_o = 1$. Indeed, the number of iterations for all methods increases 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 Table 5, which suggests again that there is a large decoupling error in the first stage of CPTR. As discussed in section 3.5, 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 iteration, which may be a poor approximation when injecting hot water into a reservoir filled with cold oil.

TABLE 7
Strong scaling: 3D well case, $S_o = 0.99$. Average linear iterations per nonlinear iteration.

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

TABLE 8
Strong scaling: 3D well case $S_o = 1$. Average linear iterations per nonlinear iteration.

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

Furthermore, the well case with $S_o = 1$ is very similar to some single-phase test cases in [34], 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 to the coupling of the saturation variable. However, using decoupling strategies, either CPTR-uAMG(TI) or reordered CPTR-Block mentioned in section 3.5 does not change the poor scaling behavior illustrated in Table 6.

4.1.3. 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 7, we observe for the case with $S_o = 0.99$ that as we increase the number of processors from 1 to 16, the number of iterations increases by around 31% for CPR-AMG, 73% for CPR-AMG-ILU(1), and 7% for CPTR. The bulk of the increase for CPR-AMG-ILU(1) occurs when going from 1 to 2 processors, around 40%. This showcases how CPR-AMG-ILU(1) is much more effective in serial.

In Table 8, we show the results for the well case for $S_o = 1$. We observe a similar increase in the number of iterations for CPR-AMG and CPTR of 7% and 6%, respectively. For CPR-AMG-ILU(1), the increase in iterations is 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 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 Block ILU for its temperature update.

4.1.4. Synthesis. In CPR, temperature-related effects are only treated in the second stage by ILU. On the other hand, the CPTR strategy tackles these effects in first stage using some type of AMG method, which captures diffusive effects. As the mesh is refined, heat diffusion becomes more significant, resulting in the poor scaling of CPR, while CPTR exhibits near mesh independence in most well cases and mesh independence in heater cases.

In terms of computational time, CPTR becomes competitive when diffusion is dominant, which in our experiments is when the mesh is sufficiently refined.

Block ILU becomes a weaker, albeit cheaper, method as the number of processors increase. Since CPTR does not depend on ILU for the treatment of temperature, it is more robust to a weaker second-stage. Note that the authors have observed industrial cases where a strong second stage is needed for CPR's convergence.
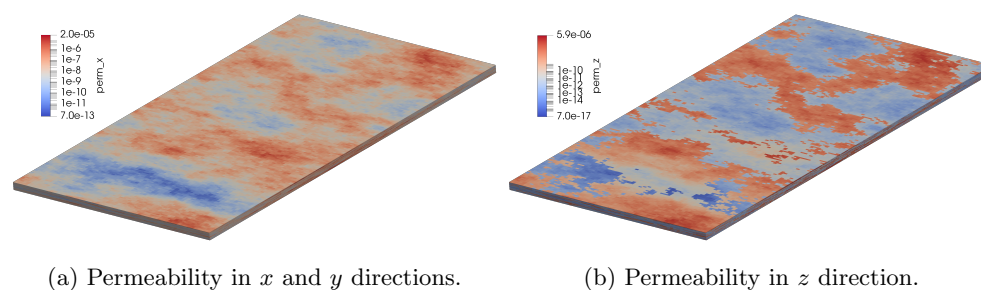
(a) Permeability in x and y directions.

(b) Permeability in z direction.

FIG. 1. *Log of permeability of the SPE10 test case (mm^2).*

TABLE 9
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-Block(AMG)	11.8	12.2	12.2	12.2	12.3
CPR-LU	10.2	10.3	10.2	10.2	10.4
CPTR-Block(LU)	10	9.91	9.91	10	10.1

In brief, CPTR is preferable when the second-stage solver is too weak for CPR, notably for diffusion-dominated flows or when many processors are available.

4.2. SPE10 test case. We consider the benchmark problem SPE10 [7], a highly heterogeneous case, 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 $365.76 \times 670.56 \times 12.192$ m, and the mesh is $60 \times 220 \times 20$. The permeability is isotropic in the x-y plane and anisotropic otherwise. The permeability fields are illustrated in Figure 1.

We consider a case with wells and heaters (W+H). 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 version of the Peaceman well model described in [34]. The production wells produce with a bottom-hole pressure of 2.7579×10^7 Pa. The injection well injects hot water with a maximum rate of $q = 1.8 \times 10^{-3}$ m^3 s^{-1}. The initial temperature is uniform at 288.706 K, and the injection temperature is 373.15 K. Heater placement is the same as for wells, and so is the heating temperature.

We simulate injection and production for 100 days where the time-steps are chosen with simple heuristics such that Newton's method usually converges in fewer 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_{pp}^{-1} and $\tilde{S}_T^{-1}$ are approximated by either AMG V-cycle or direct LU factorization, denoted CPR-AMG and CPTR-Block(AMG) or CPR-LU and CPTR-Block(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 any such weaknesses. LU is thus only used as a benchmark for testing the expected lower bound for the iteration count.

We take a total of 104 time-steps. The average linear iterations per nonlinear iteration are shown in Table 9. We observe that all methods have a small increase

TABLE 10
Condition numbers of the different matrices and Schur complement approximations for various cases ($S_o = 0.9$).

Matrix/Case	H	W	W+H	h.p. W	h.p. W+H
$\tilde{S}_{\text{diag}}^{-1}S$	1.043	43.11	122.3	8.13E+07	3.64E+07
$\tilde{S}_{A_{TT}}^{-1}S$	1.034	7.959	3.050	6.849	382.1
$\tilde{S}_T^{-1}S$	1.020	1.019	1.010	1.556	1.95
A_{TT}	5.24E+05	8.802	1188	1093	1.60E+05
S	5.36E+05	1.817	1172	241.7	1.59E+05

in iterations. At the beginning of the simulation, heating is the most significant effect. For the first time-steps, CPTR-Block(AMG) takes fewer iterations than CPR-AMG, but the total number of iterations becomes the same at the 30th time-step (around 12 days, results not shown here). In contrast, CPTR-Block(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-Block(AMG).

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. On the other hand, the cases considered in section 4.1 have finer grids and thus diffusion-dominated flows. In [11], versions of CPTR are shown to perform well for SPE10 permeability fields with laboratory length scales.

4.3. Numerical justification of the Schur complement approximation. We now compare the action of the inverses of the different Schur complement approximations. This is done for cases similar to the single-phase cases described in [34]. The only difference is that we start with $S_o = 0.9$ and inject water with a temperature of 373.15 K. In brief, the domain is a 2D slice of SPE10 with a 60×120 grid. For the well case (W), there is 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.), permeability is increased by a factor of 1,000 to emulate advection-dominated heat flow.

For different Schur complement approximations, we look at the condition number of their inverse applied to the full Schur complement (using a direct solver to invert the matrices). While the condition number is not fully indicative of the performance of preconditioned GMRES, it informs us about the quality of the approximations. In Table 10, we observe that our custom approximation $\tilde{S}_T$ approximates the Schur complement well, even for the high-permeability cases where the simpler approximations struggle.

In terms of the actual number of GMRES iterations, $\tilde{S}_T$ always performs the best for CPTR (results for the single-phase case shown in [34]). This difference is even more significant for harder cases (for example, high permeability). In some cases, GMRES does not converge before the prescribed maximum number of iterations when using other approximations. See [34] for single-phase flow examples where the other Schur complements exhibit worse performance for heterogeneous or anisotropic permeability fields.

4.4. High pressure-temperature cross-coupling. In section 4.1.1, we considered the use of unknown-based AMG (uAMG) and our block preconditioner for the

TABLE 11
Increased cross-coupling with $\Delta t = 2$ days. 2D isotropic well case. $N = 160$. Average linear iterations per nonlinear iteration.

Method/Factor	1	5	10	15	20
CPR-AMG	8.29	8.29	8.42	8.63	8.8
CPTR-Block(AMG)	5.29	5.29	5.57	6	6.27
CPTR-uAMG	5.57	6.14	7	8.5	10.6
CPTR-BD(LU)	8.29	9.14	10.1	11.1	12.8
CPTR-BD(AMG)	8.29	9.14	10.3	11.1	12.8

TABLE 12
Increased cross-coupling with $\Delta t = 0.1$ days. 2D isotropic sell case. $N = 160$. Average linear iterations per nonlinear iteration.

Method/Factor	1	15	30	35	40
CPR-AMG	5.33	6.17	5	5.17	6.33
CPTR-Block(AMG)	5	5.83	4	4.33	5.17
CPTR-uAMG	4.83	8.33	21.7	92.8	>200
CPTR-BD(LU)	5.83	7.83	12.3	19.7	>200
CPTR-BD(AMG)	6	7.5	12.3	19.7	>200

first stage of CPTR. While uAMG usually results in more iterations, it exhibits similar scalability. However, uAMG is not expected to perform well under a strong cross-coupling. Here, we artificially increase the cross-coupling of the pressure-temperature system to compare the robustness of different two-stage preconditioners. Along with uAMG and the block preconditioner, we also show results for CPTR where the first stage is a block diagonal preconditioner using either LU or AMG on the A_{pp} and A_{TT} blocks. To avoid confusion, we call these methods block-diagonal LU (BD(LU)) and block-diagonal AMG (BD(AMG)).[2] We also consider CPR with AMG in the first stage.

Recall that we have only considered the flow of two fluids: water and a heavy oil. The viscosity of the oil varies greatly with temperature and the water viscosity much less so. As for their densities, liquid water and oil are much less compressible and thermally expansive than other fluids, such as gases. To create a higher pressure-temperature cross-coupling, we increase the compressibility coefficient c and the thermal expansion coefficient β of the oil density as defined in section 2.1.4 of [34].

We consider the 2D isotropic well case from section 4.1.1 with $N = 160$. Both coefficients c and β are multiplied by the same increasing factor. Since larger coefficients result in a stiffer problem, smaller time-steps are needed for the convergence of Newton's method. Results for a time-step of 2 days and 0.1 days are shown in Tables 11 and 12, respectively. For CPTR, we observe that the block preconditioner exhibits a very small increase in iterations, while the other methods have a steady increase in iterations. It is clear that the block preconditioner is much better at dealing with a strong cross-coupling than the other versions of CPTR. For CPR, it shows similar robustness with respect to a strong cross-coupling. Note that by taking smaller time-steps, we are in a regime where CPR is competitive. Since CPR performs appropriately but CPTR (other than CPTR-Block(AMG)) does not, this may indicate that the coupling A_{Tp} is more important than A_{pT}. The temperature solution provided in

---

[2]Note that uAMG is different from BD(AMG) since the coupling blocks remain in the coarse grid correction of uAMG.

the first stage of CPTR by solvers other than the block preconditioner is not accurate. It remains to be seen if A_{pT} is more important in other cases.

In Table 12, the number of iterations eventually blows up for uAMG, BD(LU), and BD(AMG). However, in the case of uAMG, this may not necessarily be because of a strong cross-coupling but perhaps that the properties of A_{TT} are no longer amenable to the application of AMG. Indeed, the derivative of density with respect to temperature is negative. Consider the derivative of the oil enthalpy

$$(4.1) \qquad (c_{v,o}\rho_o T)_T = c_{v,o}\rho_o + c_{v,o}(\rho_o)_T T,$$

which is positive for the default parameters of the earlier sections. However, for the larger factors in Table 12, the negative influence of $(\rho_o)_T$ begins to dominate. For the largest factors, this results in A_{TT} losing diagonal dominance, becoming indefinite, and eventually having a negative diagonal. This is, of course, far from the ideal properties for the convergence of AMG. For uAMG, the coarse pressure problem is coupled through temperature using the prolongation operator for the temperature coarse grid. Since that operator is not descriptive, this may indicate why uAMG is doing worse than BD(AMG).

On the other hand, the block preconditioner does not apply AMG to A_{TT} but rather to the approximate Schur complement $\tilde{S}_T$. Since this approximation does not include $(\rho_o)_T$, it will not suffer from a larger thermal expansivity coefficient.

5. Conclusion. In this work, we have implemented a fully implicit parallel non-isothermal multiphase flow in a porous media simulator including two preconditioning strategies, CPR and CPTR. The first stage of CPTR can be a block preconditioner with our own Schur complement approximation or an unknown-based AMG method. Standard decoupling operators are available for CPR as well as their extension for CPTR.

On coarse grids, both CPR and CPTR 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. Algebraic decoupling operators do not solve this issue. To deal with a strong pressure-temperature cross-coupling, our block preconditioner is better than the less coupled alternatives in the first stage of CPTR.

We believe that more work is needed to understand the decoupling error from the first stage of CPTR. To regain the scalability that CPTR displays in other cases, an accurate energy equation is key. It is still unclear if this can be done with another CPR-like multistage preconditioner.

Appendix A. Physical quantities. For the oil viscosity μ_o, we choose a correlation from [3] where μ_o varies hugely with temperature. For the oil density ρ_o, we choose a function of pressure and temperature given by an exponential correlation. We choose an "API gravity" of 10 such that the density of oil and water are the same at surface conditions. Details and choices of parameters for μ_o and ρ_o can be found in section 2.1.4 of [34].

For the water viscosity μ_w, we use the correlation [17]

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

TABLE 13
Parameters for the water viscosity correlation.

A	B	C
2.1850	0.04012	5.1547×10^{-6}

TABLE 14
Parameters for the water density correlation.

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

which takes temperature T_F in °F and returns viscosity in cp ($0.001 \text{ kg m}^{-1} \text{ s}^{-1}$). The parameters A, B, C can be found in Table 13.

For the water density, we use Trangenstein's modification of Kell's formula [22]

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

which takes temperature T_C in °C, and pressure p in MPa. The parameters are given in Table 14.

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 $k_{rw} = 1 - S_o$.

Acknowledgments. We would like to thank the editor and the reviewers. Their comments have greatly improved the structure and readability of this paper.

REFERENCES

[1] M. S. ALNÆS, A. LOGG, K. B. ØLGAARD, M. E. ROGNES, AND G. N. WELLS, *Unified form language: A domain-specific language for weak formulations of partial differential equations*, ACM Trans. Math. Software, 40 (2014), p. 9.

[2] 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, AND H. ZHANG, *PETSc Web Page*, 2017, http://www.mcs.anl.gov/petsc.

[3] T. BENNISON, *Prediction of Heavy Oil Viscosity*, presented at the IBC Heavy Oil Field Development Conference, Vol. 2, 1998, p. 4.

[4] Q. M. BUI, H. C. ELMAN, AND J. D. MOULTON, *Algebraic multigrid preconditioners for multiphase flow in porous media*, SIAM J. Sci. Comput., 39 (2017), pp. S662–S680.

[5] Q. M. BUI, L. WANG, AND D. OSEI-KUFFUOR, *Algebraic multigrid preconditioners for two-phase flow in porous media with phase transitions*, Adv. Water Res,, 114 (2018), pp. 19–28.

[6] Z. CHEN AND Y. ZHANG, *Well flow models for various numerical methods*, Int. J. Numer. Anal. Model., 6 (2009).

[7] M. A. CHRISTIE AND M. J. BLUNT, *Tenth SPE comparative solution project: A comparison of upscaling techniques*, in SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, Richardson, TX, 2001.

[8] T. CLEES, *AMG Strategies for PDE Systems with Applications in Industrial Semiconductor Simulation*, Ph.D. thesis, University of Köln, 2005.

[9] T. CLEES AND L. GANZER, *An efficient algebraic multigrid solver strategy for adaptive implicit methods in oil-reservoir simulation*, SPE J., 15 (2010), pp. 670–681.

[10] K. H. COATS, *In-situ combustion model*, Soc. Petroleum Engrs. J., 20 (1980), pp. 533–554.

[11] M. A. CREMON, N. CASTELLETTO, AND J. A. WHITE, *Multi-Stage Preconditioners for Thermal-Compositional-Reactive Flow in Porous Media*, preprint, arXiv:1912.04385, 2019.

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

[13] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, New York, 2014.

[14] R. Eymard, T. Gallouët, and R. Herbin, *Finite volume methods*, Handb. Numer. Anal., 7 (2000), pp. 713–1018.

[15] R. D. Falgout and U. M. Yang, *HYPRE: A library of high performance preconditioners*, in International Conference on Computational Science, Springer, New York, 2002, pp. 632–641.

[16] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala, *ML 5.0 Smoothed Aggregation User's Guide*, Technical report SAND2006-2649, Sandia National Laboratories, Albuquerque, NM, 2006.

[17] J. W. Grabowski, P. K. Vinsome, R. C. Lin, G. Behie, and B. Rubin, *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, Richardson, TX, 1979.

[18] S. Gries, *System-AMG Approaches for Industrial Fully and Adaptive Implicit Oil Reservoir Simulations*, Ph.D. thesis, University of Köln, 2015.

[19] S. Gries, K. Stüben, G. L. Brown, D. Chen, and D. A. Collins, *Preconditioning for efficiently applying algebraic multigrid in fully implicit reservoir simulations*, SPE J., 19 (2014), pp. 726–736.

[20] V. E. Henson and U. M. Yang, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177.

[21] P. Jenny, S. H. Lee, and H. A. Tchelepi, *Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media*, J. Comput. Phys., 217 (2006), pp. 627–641.

[22] 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*, J. Chem. Eng. Data, 20 (1975), pp. 97–105.

[23] R. C. Kirby and L. Mitchell, *Solver composition across the PDE/linear algebra barrier*, SIAM J. Sci. Comput., 40 (2018), pp. C76–C98.

[24] S. Lacroix, Y. V. Vassilevski, J. Wheeler, and M. F. Wheeler, *Iterative solution methods for modeling multiphase flow in porous media fully implicitly*, SIAM J. Sci. Comput., 25 (2003), pp. 905–926.

[25] S. Lacroix, Y. V. Vassilevski, and M. F. Wheeler, *Iterative Solvers of the Implicit Parallel Accurate Reservoir Simulator (IPARS), I: Single Processor Case*, TICAM report 00-28, University of Texas, Austin, 2000.

[26] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Vol. 31, Cambridge University Press, Cambridge, 2002.

[27] J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[28] M. Muskat and M. W. Meres, *The flow of heterogeneous fluids through porous media*, Physics, 7 (1936), pp. 346–363.

[29] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Series on Computational Methods in Mechanics and Thermal Science, Hemisphere Publishing Corporation (CRC Press, Taylor & Francis Group), Washington, DC, 1980.

[30] D. W. Peaceman, *Interpretation of well-block pressures in numerical reservoir simulation (includes associated paper 6988)*, Soc. Petroleum Engrg. J., 18 (1978), pp. 183–194.

[31] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly, *Firedrake: Automating the finite element method by composing abstractions*, ACM Trans. Math. Software, 43 (2016), p. 24.

[32] B. Rivière, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*, SIAM, Philadelphia, 2008.

[33] T. Roy, *Preconditioning for Thermal Reservoir Simulation*, D.Phil. thesis, Oxford University, Oxford, 2019.

[34] T. Roy, T. B. Jönsthövel, C. Lemon, and A. J. Wathen, *A block preconditioner for non-isothermal flow in porous media*, J. Comput. Phys., 395 (2019), pp. 636–652.

[35] J. W. Ruge, *AMG for problems of elasticity*, Appl. Math. Comput., 19 (1986), pp. 293–309.

[36] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, Frontiers in Applied Mathematics 3, SIAM, Philadelphia, 1987, pp. 73–130.

[37] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.

[38] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[39] A. Sahni, M. Kumar, and R. B. Knapp, *Electromagnetic heating methods for heavy oil reservoirs*, in SPE/AAPG Western Regional Meeting, Society of Petroleum Engineers, Richardson, TX, 2000.

[40] R. Scheichl, R. Masson, and J. Wendebourg, *Decoupling and block preconditioning for sedimentary basin simulations*, Comput. Geosci., 7 (2003), pp. 295–318.

[41] K. Stüben, *An introduction to algebraic multigrid*, in Multigrid, U. Trottenberg, C. W. Oosterlee, and A. Schuller, eds., Academic Press, New York, 2001, pp. 413–532.

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

[43] J. R. Wallis, R. P. Kendall, and T. E. Little, *Constrained residual acceleration of conjugate residual methods*, in SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, Richardson, TX, 1985.

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

[45] Z. Y. Wong, *Sequential-Implicit Newton's Method for Geothermal Reservoir Simulation*, Ph.D. thesis, Stanford University, Stanford, CA, 2018.