Domain Decomposition Strategies for Non-linear Flow Problems in Porous Media

Jan Ole Skogestad*, Eirik Keilegavlen*, Jan M. Nordbotten

*Dept. of Mathematics, University of Bergen, Johannes Bruns gate 12, 5008 Bergen, Norway
*Dept. of Civil and Environmental Engineering, Princeton University, Princeton, NJ, USA

Abstract

Domain decomposition (DD) methods, such as the additive Schwarz method, are almost exclusively applied to a linearised equations. In the context of non-linear problems, these linear systems appear as part of a Newton iteration. However, applying DD methods directly to the original non-linear problem has some attractive features, most notably that the Newton iterations now solve local problems, and thus are expected to be simpler. Furthermore, strong, local non-linearities may to a less extent affect the numerical algorithm. For linear problems, DD can be applied both as an iterative solver or as a preconditioner. For non-linear problems, it has until recently only been understood how to use DD as a solver.

In the context of non-linear porous-medium flow problems, this article offers the first systematic study of domain decomposition strategies. The study thus compares four different approaches, which represents DD applied both as a solver and preconditioner, to both the linearised and non-linear equations. Our model equations are those obtained from a fully implicit discretisation of immiscible two-phase flow in heterogeneous porous media. In particular we emphasise the case of non-linear preconditioning, an algorithm that to our knowledge so far has not been studied nor implemented for flow in porous media. Our results show that the novel algorithm is as much as 40% faster for the most challenging problems.

Keywords: Non-linear multiphase flow, Porous media, Domain decomposition, ASPIN, Non-linear preconditioning, Non-linear solvers

1. Introduction

Simulation of flow in subsurface porous media is of crucial importance to a variety of applications, such as oil and gas recovery, CO₂ storage and geothermal energy extraction. In order to give an accurate description of the flow pattern, highly non-linear partial differential equations are needed. Both the parameter fields (and foremost permeability) and the solution may vary over several orders of magnitudes, and the variations may have different correlation lengths. To resolve all the fine scale variations, a fine discretisation is required, which leads to a large number of unknowns. This leads us to seek advanced numerical methods that can handle both high degrees of non-linearity and multiscale features, for large systems of equations.

The primary variables in reservoir simulation are usually chosen to be the pressure in one phase, and a set of phase saturations or mass variables. The governing equations are an (almost) elliptic pressure equation, and transport equations that are (depending on scale) hyperbolic or parabolic. For realistic physical processes, both the transport and the pressure equations are non-linear. In order to simulate the spatially discretised non-linear system without time-step size being limited by stability, an implicit temporal discretisation is frequently employed method for some or all equations. With a fully implicit method (FIM), a large non-linear system must be solved at each time step.

*Corresponding author. Tel.: +47 55582857, Fax: +47 55589672.

Email addresses: jan.skogestad@math.uib.no (Jan Ole Skogestad), eirik.keilegavlen@uib.no (Eirik Keilegavlen), jan.nordbotten@math.uib.no (Jan M. Nordbotten)
The standard way to handle non-linearity is by some variation of Newton iterations, in which a system of linear equations must be solved for each iteration. This consumes a considerable amount of the total simulation time, and methods have been developed both to reduce the time it takes to solve linear system, and reduce the number of Newton iterations needed. This involves amongst others two-stage preconditioners [24, 3, 22], reordering techniques [14, 15], and multiscale methods, see e.g. [11, 17, 12].

Another efficient solution strategy for linear and non-linear equations is domain decomposition (DD) [18, 21, 23], and this paper is devoted to investigation of how to best apply DD to simulate flow and transport in porous media. Domain decomposition has advantages both for elliptic problems as well as hyperbolic equations [25], and it is therefore well suited for our problems. Importantly, the decoupling of the problem into subdomains makes domain decomposition attractive for parallelisation, making efficient large scale computations possible. For non-linear systems the standard application of domain decomposition techniques is as a linear preconditioner for the solver of the linear system obtained inside the Newton iteration. Conceptually it is also possible to do DD directly to the non-linear system. The benefit of non-linear DD is evident: If the non-linearities mainly are confined to a small region of the computational domain, a spatial decomposition of the non-linear problem will make the number of Newton iterations on any sub-domain automatically adapt to the degree of non-linearity. The computational savings of this approach compared to performing Newton iterations on the whole domain in order to reduce the residual in a small region can be substantial. In reservoir simulation, non-linearities are often mostly confined to advancing fronts; hence non-linear DD should be of interest to these problems. Common practice has nevertheless been to linearise the system prior to domain decomposition. A major reason for this is that domain decomposition is far more effective as a preconditioner than as a stand-alone solver, and until recently, only linear DD preconditioners have been available. An attractive framework to overcome this bottleneck and thus facilitate non-linear preconditioning has emerged as the Additive Schwarz preconditioned inexact Newton (ASPIN) methods, quite recently introduced by Cai and Keyes [2]. ASPIN has been shown to work well for several computational fluid dynamics problems, in that it reduces the computational cost, and can handle higher non-linearities than standard Newton methods. Until now, no applications of ASPIN to porous media problems have been reported.

In this paper, we apply linear and non-linear domain decomposition both as stand-alone solvers and as preconditioners for flow and transport in porous media. We consider two-phase flow problems with a range of difficulty that spans from stable displacement in homogeneous media to highly unstable displacement in channelised systems. To highlight non-linear effects, the time stepping is fully implicit. The simulation results consistently show that the performance of ASPIN is better to that of linear preconditioning, by as much as 40%. Crucially, the performance of ASPIN is more robust than linear preconditioning in the face of increasing non-linearities, indicating that ASPIN is the domain decomposition method of choice for challenging problems. Our main conclusion is thus that contrary to industry standard, domain decomposition should be considered a preconditioner for the non-linear system, not the linear system, for optimal performance and robustness.

The rest of this paper is organised as follows. In Section 2, we describe the model problem and discretisation techniques used in this study. Section 3 gives an overview of domain decomposition methods and the specific methods considered here. In Section 4 we present numerical results from a comparative study of the different methods. The paper is concluded in Section 5.

2. Model Problem

This section reports the continuous physical model together with the discretisations employed to obtain the non-linear systems on which we base the current study.

2.1. Two-phase Flow in Porous Media

We study an idealised case of flow of two fluid phases in a porous medium. This is the prototypical system for water displacing oil, as well as CO₂ injection into brine aquifers. The phases are assumed to be immiscible and incompressible, and we neglect density differences and capillary forces. These assumptions are physically reasonable for the oil-water system, while still preserving the main non-linear terms important for this study.

The fluid potential (which we for simplicity will refer to as pressure) is then the same for both phases, and given by an elliptic equation on the form

\[ \nabla \cdot (\lambda_T K \nabla p) = q_T, \]  

1
where \( \lambda_T \) is the total mobility, \( K \) is the permeability tensor, \( p \) is pressure, and \( q_T \) represents source terms. The fluid flow is described by a hyperbolic saturation equation for each phase,

\[
\frac{\partial S_{\alpha}}{\partial t} + \nabla \cdot (f_{\alpha} \vec{v}_T) = q_{\alpha}, \quad \alpha = w, a,
\]

(2)

Here, \( \phi \) denotes the porosity, \( S_{\alpha} \) is the phase saturation, \( f_{\alpha} = \lambda_{\alpha}/\lambda_T \) is the fractional flow function, \( q_{\alpha} \) represents source terms, and the total velocity \( \vec{v}_T \) is given by Darcy’s law,

\[
\vec{v}_T = -\lambda_T K \nabla p.
\]

(3)

The phase mobility \( \lambda_{\alpha} \) is related to phase saturation by the constitutive relationship \( \lambda_{\alpha}(S_{\alpha}) = k_{\alpha}/\mu_{\alpha} \). The relative permeabilities are assumed to be a function of the phase saturation only, \( k_{\alpha} = k_{\alpha}(S_{\alpha}) \), and \( \mu_{\alpha} \) is the phase viscosity. The system is closed by requiring the saturations to fill the pore space, \( S_w + S_a = 1 \).

The primary variables in our system are \( p \) and \( S_w \). An alternative way of formulating Eqs. (1) and (2) is to define the non-linear residual

\[
\begin{align*}
F_c(u) &= \begin{bmatrix} F_p(u) \\ F_{S_w}(u) \end{bmatrix} = \begin{bmatrix} \nabla \cdot (\lambda_T K \nabla p) - q_T \\ \frac{\partial S_w}{\partial t} + \nabla \cdot (f_w \vec{v}_T) - q_w \end{bmatrix},
\end{align*}
\]

(4)

where \( u = (p, S_w)^T \), and require

\[
F_c(u) = 0.
\]

(5)

The subscript \( c \) indicates that this is the continuous form of the equations.

Eq. (5) is a coupled elliptic-hyperbolic non-linear system of equations. Since this mixed nature of the governing equations often appears in porous media simulations, results from a study of this relatively simplified problem are expected to be relevant for other, more complex problems. Sources of non-linearities in the model parameters are non-linear relative permeabilities and viscosity differences, while permeability heterogeneities instigate non-linear structures in the solution.

### 2.2. Discretisation

For simplicity, we consider only rectangular grids on rectangular domains. The spatial terms are discretised using control volume methods. The two-point flux approximation is used for the elliptic terms, while standard upstream weighting is used for hyperbolic terms. The time discretisation implicit first order (Backward Euler). With these representations of the differential operators, let \( F \) represent the discrete form of \( F_c \).

As part of the solution process, the discretised non-linear system is linearised by a Newton iteration, that is, starting from an initial guess \( u^{(0)} \), at step \( n \), solve the linear system

\[
J(u^{(n)}) \Delta u^{(n)} = -F(u^{(n)}),
\]

(6)

and obtain new solution

\[
u^{(n+1)} = u^{(n)} + \Delta u^{(n)}.
\]

(7)

Here, \( J(u) \) is the Jacobian of \( F(u) \). We calculate the derivatives contained in \( J(u) \) analytically.

In practical applications, different modifications of Newton’s method are often used. In this work, we will use the inexact Newton’s method [5, 6, 9, 13]. The primary idea is to solve the linear system (6) only as accurately as needed to provide a sufficiently good Newton search direction, using some iterative method such as GMRES. That is, find the inexact Newton direction \( s^{(n)} \) such that

\[
\|F(u^{(n)}) + J(u^{(n)}) s^{(n)}\| \leq \eta_n \|F(u^{(n)})\|,
\]

(8)

where \( \eta_n \) determines the relative accuracy to which the linear system is solved, and computing the new approximate solution by setting \( \Delta u^{(n)} = \gamma_n s^{(n)} \) in formula (7). The step length \( \gamma_n \) is determined by a line-search technique [6, 16].

The absolute accuracy needed increases as the Newton iterates \( u^{(n)} \) approach the true solution, thereby the scaling by \( \|F(u^{(n)})\| \) in the inequality above.
3. Additive Schwarz Methods

Domain decomposition methods date back to the 19th century [20], but were first brought to serious attention in the 1980’s, with the increasing availability of parallel computing [18, 21, 23]. By solving smaller subproblems on several processors, considerable savings in computational time can be made compared with single-processor computations. This has been of crucial importance for the development of faster and more accurate reservoir simulators, as well as for many other applications involving large systems of equations. However, new challenges, such as communication between the subproblems and load balancing are introduced.

Within the class of domain decomposition methods, a variety of methods are available. The additive Schwarz (AS) method [8] is among the more popular, due to its simplicity in form and its suitability for parallel computing. In its most basic form, the method consists of decomposing the problem domain into a set of possibly overlapping subdomains and solving the problem on each subdomain using data from the neighbouring subdomains at the boundaries. The global solution is then updated and the process repeated until convergence is achieved.

3.1. Additive Schwarz Methods for Linear Problems

For a linear problem, \( Ax = b, \ x \in V = \mathbb{R}^n \) (9), the additive Schwarz method may be written as

\[
x_{V_i}^{(k+1)} = x_{V_i}^{(k)} + A_{V_i}^{-1} \left( b_{V_i} - A_{V_i} x_{V_i}^{(k)} - A_{V \setminus V_i} x_{V \setminus V_i}^{(k)} \right), \quad i = 1, ..., N,
\]

for iteration step \( k \) on subdomain \( V_i \subset V \), starting from some initial guess \( x^{(0)} \). We see that on each iteration step, the AS linear solver (LS) uses only boundary data from the previous iteration step. In other words, all subdomain problems can be solved in parallel.

As the subdomains \( V_i \) may be overlapping, some components of \( x \) are solved for more than once, so we have to treat the overlap regions with special care when forming the global solution. In a porous medium setting, a natural choice is to average pressure variables and apply upstream weighting on saturation variables.

One major drawback of the AS iteration (10) is its poor convergence properties. Therefore, AS is in practice not used in the form stated above, but is instead applied as a preconditioner to the global system. This linear preconditioner (LP) can be obtained by formulating the iteration (10) globally,

\[
x^{(k+1)} = x^{(k)} + \sum_{i=1}^{N} B_i \left( b - A x^{(k)} \right).
\]

Here \( B_i = R_i^T A_{V_i}^{-1} R_i \), where \( R_i \) is the restriction from \( V \) to \( V_i \). This motivates a preconditioner,

\[
P = \sum_{i=1}^{N} B_i.
\]

In the formulas above, the overlap handling is not taken into account. In practice, each term in the sum is weighted by a diagonal matrix where each diagonal element corresponds to one cell in the subdomain. The weights are one for interior cells, and sum up to one for each cell in the overlaps. Large linear systems are typically solved by some Krylov subspace method such as GMRES [19]. The AS preconditioner (12) can speed up this iteration significantly, and has proven to be a powerful tool for a variety of applications, including flow in porous media.

3.2. Additive Schwarz methods for non-linear problems

In porous-medium flow problems, the solution typically only changes significantly in parts of the domain, e.g. around an advancing saturation front. This localisation is not exploited in linear AS, as each Newton iteration must be performed globally. That is, the AS strategy can be improved if the Newton iterations can be performed locally. See Fig. 1 for an illustration. This can be achieved by applying AS directly on the original non-linear problem (5), which will give a non-linear problem to solve on each subdomain. Hence, the Newton iterations become localised, so that
Figure 1: Localisation of Newton iterations.
if there are regions with highly non-linear problems, this can be resolved by solving small non-linear systems instead of global ones. A generalisation of the iteration (10) to non-linear problems is presented in [7]. The local non-linear problems can be written

\[ F_{\Omega_i}(u_{\Omega_i}) = 0, \quad i = 1, \ldots, N, \tag{13} \]

where \( \Omega_i \in \Omega \). The internal boundary conditions are still based on information in neighbouring subdomains from the previous iteration step. A natural choice is to use Dirichlet conditions on the pressure and let phase fluxes across boundaries be based on the saturations on the upstream side.

The convergence issues that are seen in LS are also present here in the non-linear solver (NS), which may be a major reason why it is not very widely used, despite its attractive features regarding localisation of Newton iterations. This leads us to consider AS as a preconditioner (NP) for the original non-linear problem. The additive Schwarz preconditioned inexact Newton (ASPIN) framework, developed by Cai and Keyes [2], provides the tools needed. For further analysis and a convergence proof, see [1]. It roughly consists of preconditioning (5) and solving the resulting global preconditioned non-linear system

\[ F(u) = 0, \tag{14} \]

using an inexact Newton method. The method has shown promising potential in computational fluid dynamics, but to our knowledge it has not been studied for porous media applications.

Some details are in order here. Most of the following is taken from [2] and [1], where more comprehensive presentations of ASPIN are given. The non-linearly preconditioned system (14) is obtained by defining the quantity \( T_{\Omega_i}(u) \) as the solution of the local non-linear system

\[ F_{\Omega_i}(u - T_{\Omega_i}(u)) = 0, \quad i = 1 \ldots N \tag{15} \]

for \( T_{\Omega_i}, i = 1 \ldots N \). \( T_{\Omega_i}(u) \) may be interpreted as the local correction of an initial guess \( u \), given the boundary information from the neighbouring subdomains. The non-linearly preconditioned \( F(u) \) is then defined as

\[ F(u) = \sum_{i=1}^{N} T_{\Omega_i}(u). \tag{16} \]

It can be shown that for the linear case, Eq. (14) reduces to the linearly preconditioned system presented in Section 3.1.

The non-linear system is solved using the inexact Newton’s method, which implies that we need the Jacobian of the preconditioned system, \( \mathcal{J}(u) = F'(u) \). It can be shown [2, 1] that

\[ T_{\Omega_i}'(u) = R_i^T \left( R_i J(u - T_{\Omega_i}(u)) R_i^T \right) R_i J(u - T_{\Omega_i}(u)). \tag{17} \]

The Jacobian can be written as

\[ \mathcal{J}(u) = \sum_{i=1}^{N} T_{\Omega_i}'(u), \tag{18} \]

This matrix is rather inconvenient and expensive to calculate, since the Jacobians in each term of the sum have different arguments. An approximation suggested by [2] is to assume that \( u \) is sufficiently close to the exact solution \( u^* \), which by the continuity of \( T_{\Omega_i}(u) \) implies that \( T_{\Omega_i}(u) \approx 0 \). Since the Jacobian is also continuous, we will then also have that \( J(u - T_{\Omega_i}(u)) \approx J(u) \). Then we can write

\[ T_{\Omega_i}'(u) \approx T_{\Omega_i}'(u) = R_i^T \left( R_i J(u) R_i^T \right) R_i J(u). \tag{19} \]

which leads to the approximated preconditioned Jacobian

\[ \tilde{\mathcal{J}}(u) = \sum_{i=1}^{N} \tilde{T}_{\Omega_i}(u). \tag{20} \]

We see that this gives the same preconditioner as in the linear case. It should be noted that this approximate preconditioner is only applied to the left hand side of the equation, the right hand side is still treated non-linearly. To form
the exact global Jacobian matrix after locally updating $u$, a global Jacobian matrix must be computed for the update in each of the subdomains, and matrix-matrix products are needed to obtain the full Jacobian. The computational cost of this procedure would be severe, and in practice the approximation (20) produce satisfying results.

For the global stopping conditions, we make a minor departure from the original ASPIN algorithm by using the residual of the original non-linear equation $F(u)$ instead of the preconditioned residual $F(u)$. Both of these global residuals will approach zero as $u \to u^*$ and are valid choices. Since the governing equation we are solving is obtained by requiring mass conservation, it seems reasonable to control the Newton iteration based on a mass balance error instead of a mathematically constructed preconditioned residual.

We summarise this outline of ASPIN in the form of an algorithm, starting from an initial guess $u^{(0)}$. Then $u^{(n+1)}$ is obtained from $u^{(n)}$ by performing the steps in Algorithm 1.

**Algorithm 1 ASPIN**

1. Check stopping conditions on the physical non-linear residual $F(u^{(k)})$
2. Compute the non-linear residual $g^{(n)} = F(u^{(n)})$:
   (a) Find $g_i^{(n)} = T_\Omega_i(u^{(n)})$ by solving the local non-linear systems
   
   $$ F_\Omega_i(u^{(n)} - g_i^{(n)}) = 0, \quad i = 1, 2, \ldots, N $$

   starting from $g_i^{(n)} = 0$.
   (b) Form the global residual
   
   $$ F(u^{(n)}) = g^{(n)} = \sum_{i=1}^N g_i^{(n)}. $$

3. Find the inexact Newton direction $s^{(n)}$ by solving
   
   $$ \tilde{J}(u^{(n)}) s^{(n)} = -F(u^{(n)}) $$

   such that
   
   $$ \|F(u^{(n)}) + \tilde{J}(u^{(n)}) s^{(n)}\| \leq \eta_n \|F(u^{(n)})\|, $$

   where $\eta_n \in [0, \eta_{\text{max}}]$
4. Compute the new approximate solution
   
   $$ u^{(n+1)} = u^{(n)} - \gamma^{(n)} s^{(n)}, $$

   where $\gamma^{(n)}$ is a damping parameter obtained by linesearch.

We have now presented four options for applying AS to a non-linear porous-medium flow problem, with some advantages and disadvantages. In the linear case, it is known that AS preconditioners perform better than AS solvers. It also seems to be a good idea to work directly on the non-linear rather than on the linearised problem. These features, as well as the labelling we will employ to identify the four different methods, are summarised in Tab. 1.

At this point, some comments on the form of linear preconditioning applied here is in order. For ease of comparison, we have implemented the linear analogue to ASPIN, without acknowledging that the variables are elliptic and hyperbolic, respectively; the only exception is the handling of the overlapping regions. Efficient linear preconditioning should exploit the properties of the variables, two natural options for porous media flow is two-stage preconditioning like the Constrained Pressure Residual (CPR) [24] and reordering techniques, see e.g. [14]. The CPR preconditioner first isolates the pressure part of the linear system, facilitating use of a multi-level elliptic solver, and then treats the

<table>
<thead>
<tr>
<th>LS</th>
<th>LP</th>
<th>NS</th>
<th>NP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-linear</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Preconditioner</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Summary of the four different AS strategies discussed in this paper.
updated residual, preferably by some local method like incomplete LU factorisation. This can reduce the computation time needed for one Newton iteration significantly. Reordering approaches use the fact that for a known pressure, and thereby velocity field, the saturations can be updated non-linearly, in practice reducing the number of Newton iterations needed. Both these approaches can easily be implemented in the local non-linear solvers in ASPIN. As such, the current study emphasises the implications of non-linear versus linear domain decomposition strategies applied to the typical form of the equations, recognising that further optimisation of the preconditioning strategies may be possible.

### 4. Numerical Experiments

For the comparison of the different variants of AS outlined in the previous section, we have set up a suite of problems with what we expect to be increasing difficulty, ranging from linear governing equations on a homogeneous domain to highly non-linear equations on a highly heterogeneous domain. Tab. 2 summarises the main features of each test case, ordered roughly from least to most non-linear. The permeability field used in cases C-E is generated randomly from a lognormal distribution, and then smoothed out, see Fig. 1(a). In cases F and G, we have imported permeability and porosity data from the 10th SPE Comparative Solution Project (SPE10) [4]. We have used the top (1) and bottom (85) layers, which respectively represent a near-shore sand layer and a fluvial layer with long correlation lengths, as illustrated in Fig. 4. The porosity is homogeneous, except in the SPE10 cases, where porosity is correlated to the permeability, as provided by the benchmark data.

### Table 2: Characteristic properties of the different test cases considered in this study.

<table>
<thead>
<tr>
<th>Case</th>
<th>Mobility</th>
<th>Permeability</th>
<th>Viscosity</th>
<th>Unknowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Linear</td>
<td>Homogeneous</td>
<td>1:1</td>
<td>32768</td>
</tr>
<tr>
<td>B</td>
<td>Non-linear</td>
<td>Homogeneous</td>
<td>1:1</td>
<td>32768</td>
</tr>
<tr>
<td>C</td>
<td>Non-linear</td>
<td>Random heterogeneous</td>
<td>1:1</td>
<td>32768</td>
</tr>
<tr>
<td>D</td>
<td>Non-linear</td>
<td>Random heterogeneous</td>
<td>1:10</td>
<td>32768</td>
</tr>
<tr>
<td>E</td>
<td>Non-linear</td>
<td>Random heterogeneous</td>
<td>1:100</td>
<td>32768</td>
</tr>
<tr>
<td>F</td>
<td>Non-linear</td>
<td>SPE10 layer 1</td>
<td>1:100</td>
<td>26400</td>
</tr>
<tr>
<td>G</td>
<td>Non-linear</td>
<td>SPE10 layer 85</td>
<td>1:100</td>
<td>26400</td>
</tr>
</tbody>
</table>

(a) Layer 1.  
(b) Layer 85.
In the linear mobility case A, a relative permeability of \( k_{r,A} = S_{r}^A \) is used, while in the non-linear mobility simulations B-G we have \( k_{r,A} = S_{r}^A \). The phase viscosities are equal in cases A-C, while ratios of 1:10 are used for case D and 1:100 for cases E-G. The injected fluid has the lowest viscosity, which leads to unstable displacement and fingering evolving in the saturation distribution.

In cases A-E, we use a quarter-five spot setup, that is injection in the lower left corner and production in the upper right corner, on a quadratic grid with 128 cells in each direction. In cases F-G, the domain is rectangular with 60 by 220 cells. The wells are placed in the cells with the highest permeability on each of the short sides of the rectangular domain. In all cases, we use a constant rate of injection and a pressure controlled production well. On the quadratic grid, we have used a partitioning of 4 by 4 subdomains, and on the SPE10 grids 2 by 8 subdomains is used. The overlap used is 4 cells, and the partitioning is performed such that all subdomains have the same number of cells after overlap is taken into consideration.

Tolerance values are needed in order to determine when an iteration has converged. For the global non-linear iteration, an absolute tolerance is imposed on the physical residual, that is, convergence is declared when

\[
\|F(u^{(n)})\| \leq \tau_{abs}^{global,nl}.
\]  

Using an absolute tolerance ensures that the quality of the solution is independent on the quality of the initial guess. For the local non-linear systems that arise in NS and NP, the initial guess is \( g^{(n)}_i = 0 \) and we use a relative tolerance

\[
\|F_{\Omega}(u^{(n)} - g^{(n)}_i)\| \leq \tau_{rel}^{local,nl}\|F_{\Omega}(u^{(n)})\|,
\]  

as the goal here is to provide a ‘good enough’ improvement to the global residual. However, we do not require an absolute accuracy better than \( \tau_{abs}^{global,nl} \). For the global linear system, we set \( \eta = \tau_{rel}^{global,nl} \) (see Algorithm 1), while having an absolute cut-off at \( \tau_{abs}^{global,nl} \). Typically, the relative tolerances are larger than their absolute counterparts by a few orders of magnitude. Unless otherwise stated, we have used values of \( \tau_{abs}^{global,nl} = 10^{-10}, \tau_{rel}^{global,nl} = 10^{-6} \) for the global linear system, while for the local linear system, we use \( \tau_{rel}^{local,nl} = 10^{-6} \) and \( \tau_{abs}^{global,nl} = 10^{-12} \).

As an initial guess for the first time step, the single phase solution is used. The simulations are performed in Matlab. We have used GMRES for solving the global linear systems, while the backslash operator in Matlab, which employs a direct method, is used for the local linear solves.

Since the methods used are different in nature (e.g. some methods using only global Newton iterations, some using only local, some using both), it is hard to find a good metric for comparing all methods side by side. Therefore, we start out by comparing LS and LP and then NS and NP, both for relatively short simulations for cases A-E. Finally, we study the two preconditioning methods LP and NP. Here we extend the study to include cases F and G, and longer simulations. The two domain decomposition solvers, LS and NS, are too slow for the two most complex cases.

4.1. LS versus LP

Fig. 3 shows results from a comparison of LS and LP for simulations of injection of 0.0001 pvi of water into an oil-filled reservoir with constant time step 0.00001. This comparison is made mostly for completeness, as the outcome
is rather trivial. We have used $\tau_{\text{global,lin}}^{abs} = 10^{-10}$ here in order to ensure that LS converges within a reasonable time frame without having to reduce the time step. Fig. 3(a) shows the number of AS sweeps for LS and LP, respectively. An AS sweep corresponds to one iteration in the LS case, or one assembly of the preconditioner in the LP case. The operations are the same, namely solving all the local linear systems once. We see that LP requires far fewer AS sweeps than LS for most interesting cases (C-E), about one order of magnitude less. We also note that for the most simple test cases, LS performs better than LP. This may be simply because the problems are not difficult enough for preconditioning to be superior to iterative solving: LP always has to solve at least one global linear system per time step, which gives roughly one AS sweep per GMRES iteration, whereas LS only needs to solve local linear systems. When only a few local solves are needed, this becomes an advantage for the solver. However, for the most non-linear cases, which are also the most interesting ones, many local solves are needed, and the advantage of preconditioning is dominating.

Now, LP has some extra cost in solving the global linear systems, but as Fig. 3(b) shows, the total time spent on solving linear systems is not significantly affected by this. Hence, we conclude that, as expected, using AS as a preconditioner is a far better option than using it as a solver for the linearised system. We also note that this advantage becomes more evident as the non-linearity of the problem increases, especially when going from a homogeneous to a heterogeneous permeability field.

4.2. NS versus NP

The simulation specifications are the same when comparing NS and NP as in the linear case. The outcome here is less trivial, as non-linear DD preconditioning is a less studied subject than the linear counterpart. We can also here start out by looking at the number of AS sweeps. In Fig. 4(a), we see a similar pattern, although not as pronounced, as in the linear case. However, the meaning of an AS sweep is slightly different here: One AS sweep corresponds to solving all the local non-linear systems once. That is, the amount of linear solves is not fixed, since it depends on the number of local Newton iteration steps needed during a sweep, which can vary. If we compare the number of local Newton iterations, Fig. 4(b), we see a similar pattern as for the number of AS sweeps, with a more pronounced difference between the methods. However, local Newton iterations is not a sufficiently good metric, either, since ASPIN performs linear solves that are not part of local Newton iterations when assembling the approximate preconditioned Jacobian. As for the linear case, NP requires some global solves as well, but as Fig. 4(c) shows, this does not affect the results significantly. We also see a more pronounced robustness with respect to non-linearity for NP than for NS.

The conclusion of this comparison is that the result that the AS preconditioner is superior to the AS solver, seems to carry over from the linear to the non-linear case, although the advantage is slightly smaller here.

4.3. LP versus NP

For the remainder of this section, we focus on the two AS preconditioners. Here, we have run simulations for cases A, C, E, F and G. The two latter cases, from the SPE 10 project, are expected to be more difficult and more realistic than the cases studied in the previous subsections. We have used a global time step of $\Delta t_g = 0.01$ pvi, and run the simulation up to $t = 0.5$ pvi. The global time step is halved if the Newton iteration (global or local) does not converge within 10 iteration steps, and may be doubled if the previous time step was not reduced. When the simulation reaches the point that a global time step would have reached, the global step length is attempted again.
Fig. 5 shows a comparison of LP and NP. We see that ASPIN both uses fewer global Newton iterations and GMRES iterations than the linear preconditioner, Fig. 5(a) and 5(b), with a slightly smaller contrast in GMRES iterations. These tendencies are also reflected in the time spent on solving linear systems, Fig. 5(c). Note that NP is about 40% faster than LP for the most challenging problems.

When comparing the number of time steps used, we see that NP generally needs fewer time steps than LP, and the difference increases with increasing non-linearity, see Fig. 5(d). We conclude that ASPIN seems to be more robust with respect to non-linearities than linear preconditioning.

4.4. Dependence on solver parameters

Our final investigation considers the solver parameters in ASPIN. In particular, we consider how the tolerance values for the local Newton iterations and the GMRES affect the efficiency of the method. By testing with varying tolerances we can see if computational savings can be made by solving systems more inexact, and where the disadvantage of inexactness overtakes the advantage of fewer iterations. We have run a series of simulations of case G using NP with varying relative tolerances for the GMRES iteration and for the local Newton iteration. We have used a constant time step of 0.0005 and run the simulation for 10 time steps. Fig. 6 shows a summary of the results in terms of time spent solving linear systems.

Two observations of immediate interest can be drawn from the figure. First of all, we note that the total time spent solving linear systems is completely independent on the tolerance used for local non-linear systems. This indicates that the method is robust with respect to this metric, in the sense that additional computational effort (within the thresholds investigated) in the non-linear system translates through the preconditioner to the global system. On the other hand, we clearly see some dependence on the tolerance for GMRES iterations, where the best results are obtained with a tolerance of $10^{-6}$. Still, even for this parameter the dependence is weak, with computational effort spent solving linear systems bounded within a factor of about 50%.
5. Conclusions

We have presented four different strategies for applying domain decomposition methods to a two-phase flow problem in porous media. Our model problem was chosen to capture the dominant features that make simulation of multi-phase flow in geological media challenging: A) Flux functions that have strong non-linear dependency on the solution, B) Highly spatially heterogeneous parameter fields, and C) Unstable displacements exhibiting fingering.

Our numerical experiments showed that the iterative solvers were out-performed by their preconditioner counterparts. More importantly, we document that non-linear DD preconditioning, as implemented in the ASPIN method, consistently performs better than the industry-standard approach of linearisation followed by linear domain decomposition methods. In particular, we see that the ASPIN method performs better in four key metrics: Number of global Newton iterations, number of GMRES iterations, total time spent solving linear systems, longest average time steps. Each of these metrics are important for various generalisations of the model problem, and various computational architectures. As examples, the Newton iteration becomes more costly when additional physical processes such as e.g. phase equilibrium calculations are included. Conversely, the time spent solving linear systems dominates for large computational grids.

This is a motivation for further refinement and improvement on non-linear preconditioning with the ASPIN framework as the basis. In order to take full advantage of domain decomposition, a two-level variant of ASPIN is needed [10] for the elliptic part. Furthermore, significant additional computational advantage is expected when combining ASPIN with reordering methods [14] for the hyperbolic part and CPR methods for the elliptic part.

References


