Introduction

Simulation of multi-phase flow through porous media involves the solution of governing equations for the pressure. The equations can be linearized and solved with iterative methods. The time spent in the solution of the linear system depends on the size of the problem and the variations of permeability within the media. Solution of extreme problems leads to large computing time.
Reduction of computing time for large-scale problems with Proper Orthogonal Decomposition (POD) is investigated in P. Astrid (2011) and R. Markovinović (2006).
P. Astrid (2011) propose the use of a POD-based preconditioner for the acceleration of the solution of the pressure equation. The preconditioner is constructed from a set snapshots, obtained from solutions of the pressure equation for some time steps. Once the snapshots are obtained, POD method is used to obtain a set of basis functions that capture the most relevant features of the system, and can be used to improve simulation speed of the subsequent time steps.
A similar approach is used in R. Markovinović (2006), where a set of POD-based basis vectors is obtained in a similar way. However, in this case, the acceleration is achieved by improving the initial guess.
Problems with high-contrast in the permeability coefficients are approached through the use of deflation techniques in C. Vuik and Meijerink (1999). The use of deflation techniques involves the search of good deflation vectors, that usually are problem dependent. In C. Vuik and Meijerink (1999), subdomain based deflation vectors are used for layered problems with large contrast in the permeability coefficients. However, these deflation vectors cannot be used if the distribution of the permeability coefficients is not structured, such is the case for the SPE10 benchmark.
Following the ideas of P. Astrid (2011) and R. Markovinović (2006), we propose the use of POD of many snapshots to capture the system's behaviour. The basis obtained with POD is studied as an alternative choice of deflation vectors to accelerate the convergence of the pressure solution in a porous media with high-contrast variations in the permeability coefficients.
Numerical experiments are performed with the Conjugate Gradient (CG) iterative method, in combination with preconditioning (Incomplete Cholesky, IC) and deflation techniques.
Matlab Reservoir Simulation Toolbox (MRST) to obtain the linear system related to each problem Lie (2013).
We consider incompressible single-phase flow through a porous media with large variations in the permeability coefficients for layered problems and the SPE10 benchmark model.
In the first section, we present the problem definition, together with a description of the theory and governing equations of reservoir simulation. We also present the numerical methods implemented for the solution of this system of equations.
In Section we give some theory about the linear solver used for this report and we introduce preconditioning and deflation techniques. We also demonstrate two lemmas that will help us in the choice of good deflation vectors, necessary for the deflation techniques.
In Section we present numerical experiments. We describe the problem that is studied, the solver that is used and the preconditioning and deflation techniques used for the speedup of the solver. The results are also presented in this section.
Finally, we state the conclusions.

Method and/or Theory

Flow through porous media

The equations describing single-phase flow through porous media are the mass balance equation and Darcy's law, corresponding to momentum conservation (Jansen (2013)).
Darcy's law is given by:

$$\mathbf{v} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho g \nabla d),$$

Mass balance equation is:

$$\nabla \cdot (\alpha \rho \mathbf{v}) + \alpha \frac{\partial \rho \phi}{\partial t} - \alpha \rho q = 0, \tag{1}$$

where α is the cross-sectional area A for 1D, the reservoir height h for 2D, and 1 for 3D, ρ, μ are the fluid density and viscosity, p is the pressure, $\mathbf{K}$ is the rock permeability, g is the constant of gravity, d is the reservoir depth, ϕ is the rock porosity, c_l is the liquid compressibility and q are the sources. Combining the previous equations we get:

$$-\nabla \cdot \left[\frac{\alpha\rho}{\mu}\mathbf{K}(\nabla p - \rho g \nabla d)\right] + \alpha\frac{\partial(\rho\phi)}{\partial t} - \alpha\rho q = 0. \tag{2}$$

The variables ρ, ϕ, μ and $\mathbf{K}$ may be functions of the pressure p. Usually the dependence of μ and $\mathbf{K}$ on p is small enough that they can be assumed to be pressure independent parameters.
The relation between ϕ and p is given by the rock compressibility:

$$c_r = \frac{1}{\phi}\frac{\partial \phi}{\partial p}. \tag{3}$$

The relationship between ρ and p is called the liquid compressibility, it follows from the equation of state of a liquid, and can be written as:

$$c_l = -\frac{1}{V}\frac{\partial V}{\partial p}|_{T_0} = \frac{1}{\rho}\frac{\partial \rho}{\partial p}|_{T_0}, \tag{4}$$

with T_0 a constant reference temperature.
The accumulation term $\frac{\partial(\rho\phi)}{\partial t}$ of Equation (2) can be rewritten as:

$$\phi\rho c_t \frac{\partial p}{\partial t}, \tag{5}$$

where $c_t = c_l + c_r$.
Substituting (5) into (2) we obtain:

$$-\nabla \cdot \left[\frac{\alpha}{\mu}\vec{K}(\nabla p - g\nabla d)\right] + \alpha\phi c_t \frac{\partial p}{\partial t} - \alpha q = 0, \tag{6}$$

which is a nonlinear partial differential equation (PDE) for the variable p as function of the independent variables x, y, z and t.
Solution of Equation 6 by numerical means is obtained discretizing the equation with methods such as finite differences or finite volumes. Discretization schemes result in an equation of the form

$$\mathbf{V}\dot{\mathbf{p}} + \mathbf{T}\mathbf{p} = \mathbf{q}. \tag{7}$$

Where the dot represents differentiation with respect to time, $\mathbf{T}$ and $\mathbf{V}$ are known as the *transmissibility* and *accumulation* matrices. If we have incompressible model, this equation is linear and can be solved with direct or iterative methods.
If we have a compressible model or multiphase-flow, Equation (7) can be linearized using, for example, the Newton-Raphson method.
To solve Equation (7) it is necessary to define the conditions in the boundary of the domain. These conditions can be given pressures in the boundaries, Dirichlet boundary conditions, or flow rates across the boundary, Neumann boundary conditions. A combination of the previous conditions can also be used, Robin boundary conditions.

Well Model

In reservoirs, wells are typically drilled to extract or inject fluids. Fluids are injected into a well at constant surface rate or constant bottom-hole pressure, and are produced at constant bottom-hole pressure or a constant surface rate.
A widely used model is Peacemans's model, that takes into account pressure drop. For a simple vertical well in a regular rectangular grid with blocks of length Δx and width Δy, the pressure drop between the grid-block pressure and the well-bottom-hole pressure is given by:

$$p_g - p_{well} = -\frac{q}{J_{well}} ln\left(\frac{0.14\sqrt{\Delta x^2+\Delta y^2}}{r_{well}}\right), \tag{8}$$

where J_{well} is known as the well index or productivity index (PI) for a producer, and well injectivity index (WI) for injectors, and r_{well} the well radius.

Iterative solution methods

The solution of partial differential equations PDE's can be performed with numerical methods. Some of these methods, as the finite differences method, transform our equations into a linear system of the form:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{9}$$

System (9) is solved with direct or iterative methods. Direct methods achieve a final solution, while the iterative ones are stopped if the error is less than a given value.
Some of the iterative methods are: Jacobi, Gauss Seidel, and if the matrix is Symetric Positive Definite *SPD*, the Conjugate Gradient (CG) method. Iterative methods can be preconditioned or deflated to accelerate the convergence.
In this section, we present the CG method, and we describe the Preconditioning and Deflation techniques for the acceleration of this method.

Krylov subspace Methods

If we have two subspaces $\mathscr{K}_k$, $\mathscr{L}_k$ of $\mathbb{R}^n$ and we want to solve the Equation (9), with $\mathbf{A} \in \mathbb{R}^{n\times n}$ we can use a projection method onto $\mathscr{K}_k$. This method allows us to find an approximate solution $\mathbf{x}^k$ from an arbitrary initial guess solution $\mathbf{x}^0$. This approximate solution lies in the Krylov subspace of dimension k of the matrix $\mathbf{A}$ and residual $\mathbf{r}^0$,

$$\mathbf{x}^k \in \mathbf{x}^0 + \mathscr{K}_k(\mathbf{A}, \mathbf{r}^0),$$

with $\mathscr{K}_k(\mathbf{A}, \mathbf{r}^0)$ defined as:

$$\mathscr{K}_k(\mathbf{A}, \mathbf{r}^0) = span\{\mathbf{r}^0, \mathbf{A}\mathbf{r}^0, \ldots, \mathbf{A}^{k-1}\mathbf{r}^0\}.$$

Where the residual $\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$ is orthogonal to the subspace $\mathscr{L}_k$, with

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{B}^{-1}r^k, \qquad \mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k.$$

The subspace $\mathscr{L}_k$ is chosen depending on the Krylov subspace method that is used.

Conjugate Gradient Method

The Conjugate Gradient (CG) method is a Krylov subspace method for *SPD* matrices, such that

$$||\mathbf{x} - \mathbf{x}^k||_{\mathbf{A}},^{1} \tag{10}$$

is minimal, with $\mathbf{x}$ the solution of the system and $\mathbf{x}^k$ the $k-th$ iteration.
After $k+1$ iterations of the *CG* method, the error of the iteration will be bounded by:

$$||\mathbf{x} - \mathbf{x}^{k+1}||_{\mathbf{A}} \leq 2||\mathbf{x} - \mathbf{x}^0||_{\mathbf{A}} \left(\frac{\sqrt{\kappa_2(\mathbf{A})}-1}{\sqrt{\kappa_2(\mathbf{A})}+1}\right)^{k+1}.^{2} \tag{11}$$

[1] $||\mathbf{x}||_{\mathbf{A}} = \sqrt{(\mathbf{x},\mathbf{x})_{\mathbf{A}}} = \sqrt{\mathbf{x}^T\mathbf{A}\mathbf{x}}$.

[2] The condition number $\kappa_2(\mathbf{A})$ is defined as $\kappa_2(\mathbf{A}) = \frac{\sqrt{\lambda_{max}(\mathbf{A}^T\mathbf{A})}}{\sqrt{\lambda_{min}(\mathbf{A}^T\mathbf{A})}}$. If $\mathbf{A}$ is SPD, $\kappa_2(\mathbf{A}) = \frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})}$.

Algorithm 1 Conjugate Gradient (CG) method, solving $\mathbf{Ax} = \mathbf{b}$.

```
Give an initial guess $\mathbf{x}^0$. Compute $\mathbf{r}^0 = \mathbf{b} - \mathbf{Ax}^0$ and set $\mathbf{p}^0 = \mathbf{r}^0$.

    for k=0,..., until convergence
        $\alpha^k = \frac{(\mathbf{r}^k,\mathbf{r}^k)}{(\mathbf{Ap}^k,\mathbf{p}^k)}$
        $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k$
        $\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha^k \mathbf{Ap}^k$
        $\beta^k = \frac{(\mathbf{r}^{k+1},\mathbf{r}^{k+1})}{(\mathbf{r}^k,\mathbf{r}^k)}$
        $\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta^k \mathbf{p}^k$
    end
```

Preconditioning

If we want to accelerate the convergence of an iterative method, we can transform the system into another one containing a better spectrum, i.e, a smaller condition number. This can be done by multiplying the original system (9) by a matrix $\mathbf{M}^{-1}$.

$$\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b}. \tag{12}$$

The new system has the same solution but provides a substantial improvement on the spectrum. For this preconditioned system, the convergence is given by:

$$||\mathbf{x} - \mathbf{x}^{k+1}||_{\mathbf{A}} \leq 2||\mathbf{x} - \mathbf{x}^0||_{\mathbf{A}} \left(\frac{\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})} + 1} \right)^{k+1}. \tag{13}$$

$\mathbf{M}$ is a *SPD* matrix chosen such that $\kappa(\mathbf{M}^{-1}\mathbf{A}) \leq \kappa(\mathbf{A})$, and $\mathbf{M}^{-1}b$ is easy to compute.

Deflation

Deflation is used to annihilate the effect of extreme eigenvalues on the convergence of an iterative method (C. Vuik and Meijerink (1999)).
Given an *SPD* matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the deflation matrix $\mathbf{P}$ is defined as follows (Tang (2008)):

$$\mathbf{P} = \mathbf{I} - \mathbf{AQ}, \qquad \mathbf{P} \in \mathbb{R}^{n \times n}, \qquad \mathbf{Q} \in \mathbb{R}^{n \times n},$$

where

$$\mathbf{Q} = \mathbf{ZE}^{-1}\mathbf{Z}^T, \qquad \mathbf{Z} \in \mathbb{R}^{n \times m}, \qquad \mathbf{E} \in \mathbb{R}^{m \times m},$$

with

$$\mathbf{E} = \mathbf{Z}^T\mathbf{AZ}.$$

The matrix $\mathbf{E}$ is known as the *Galerkin* or *coarse* matrix that has to be invertible, in this case $\mathbf{A}$ is *SPD* and if $\mathbf{Z}$ is full rank then $\mathbf{E}$ is invertible. The full rank matrix $\mathbf{Z}$ is called the $deflation-subspace$ matrix, and it's $l < n$ columns are the $deflation$ vectors or $projection$ vectors.
Some properties of the previous matrices are ((Tang, 2008, pag. 27)):

a) $\mathbf{P}^2 = \mathbf{P}$.

b) $\mathbf{AP}^T = \mathbf{PA}$.

c) $(\mathbf{I} - \mathbf{P}^T)\mathbf{x} = \mathbf{Qb}$.

d) $\mathbf{PAZ} = \mathbf{0}^{n \times m}$.

e) $\mathbf{PA}$ is *SPSD*[3].

[3] Symmetric Positive Semi-Definite, $(\mathbf{Ax}, \mathbf{x}) \geq 0$, for all $\mathbf{x}$.

We can split the vector $\mathbf{x}$ as:

$$\mathbf{x} = \mathbf{I}\mathbf{x} - \mathbf{P}^T\mathbf{x} + \mathbf{P}^T\mathbf{x} = (\mathbf{I} - \mathbf{P}^T)\mathbf{x} + \mathbf{P}^T\mathbf{x}. \tag{14}$$

Multiplying the expression above by $\mathbf{A}$, using the properties above, we have:

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{A}(\mathbf{I} - \mathbf{P}^T)\mathbf{b} + \mathbf{A}\mathbf{P}^T\mathbf{x}, && \mathit{Property:} \\ \mathbf{A}\mathbf{x} &= \mathbf{A}\mathbf{Q}\mathbf{b} + \mathbf{A}\mathbf{P}^T\mathbf{x}, && c) \\ \mathbf{b} &= \mathbf{A}\mathbf{Q}\mathbf{b} + \mathbf{P}\mathbf{A}\mathbf{x}, && b), \end{aligned}$$

multiplying by $\mathbf{P}$ and using the properties $\mathbf{P}\mathbf{A}\mathbf{Q} = \mathbf{0}^{n\times n}$ and $\mathbf{P}^2 = \mathbf{P}$, properties d) and e), we have:

$$\begin{aligned} \mathbf{P}\mathbf{b} &= \mathbf{P}\mathbf{A}\mathbf{Q}\mathbf{b} + \mathbf{P}^2\mathbf{A}\mathbf{x}, \\ \mathbf{P}\mathbf{b} &= \mathbf{P}\mathbf{A}\mathbf{x}, \end{aligned} \tag{15}$$

where $\mathbf{P}\mathbf{b} = \mathbf{P}\mathbf{A}\mathbf{x}$ is the deflated system. Since $\mathbf{P}\mathbf{A}$ is singular, the solution $\mathbf{x}$ can contain components of the null space of $\mathbf{P}\mathbf{A}$. A solution of this system, called deflated solution, is denoted by $\hat{\mathbf{x}}$. The equation to obtain $\hat{\mathbf{x}}$ is:

$$\mathbf{P}\mathbf{b} = \mathbf{P}\mathbf{A}\hat{\mathbf{x}}. \tag{16}$$

As mentioned above, solution of Equation (15) can contain components of $\mathscr{N}(\mathbf{P}\mathbf{A})$. Thereafter, the solution of Equation (16), $\hat{\mathbf{x}}$ can be decompose as:

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{y}, \tag{17}$$

with $\mathbf{y} \in \mathscr{R}(\mathbf{Z}) \subset \mathscr{N}(\mathbf{P}\mathbf{A})$, and $\mathbf{x}$ solution of Equation (9).
Note. If $\mathbf{y} \in \mathscr{R}(\mathbf{Z})$, then

$$\mathbf{y} = \sum_{i=1}^{m} \alpha_i \mathbf{z}_i,$$

$$\mathbf{P}\mathbf{A}y = \mathbf{P}\mathbf{A}(\mathbf{z}_1\alpha_1 + \ldots + \mathbf{z}_m\alpha_m) = \mathbf{P}\mathbf{A}\mathbf{Z}\alpha, \tag{18}$$

from h) $\mathbf{P}\mathbf{A}\mathbf{Z} = \mathbf{0}^{n\times l}$, then

$$\mathbf{P}\mathbf{A}y = 0. \tag{19}$$

Therefore $\mathscr{R}(\mathbf{Z}) \subset \mathscr{N}(\mathbf{P}\mathbf{A})$.
Multiplying Equation 17 by $\mathbf{P}^T$ we obtain:

$$\mathbf{P}^T\hat{\mathbf{x}} = \mathbf{P}^T\mathbf{x} + \mathbf{P}^T\mathbf{y},$$

combining Equation (19) with f), we have:

$$\mathbf{P}\mathbf{A}y = \mathbf{A}\mathbf{P}^T y = 0.$$

Therefore

$$\mathbf{P}^T\hat{\mathbf{x}} = \mathbf{P}^T\mathbf{x}. \tag{20}$$

Substitution of Equation (20) and g) in Equation (14) leads to:

$$\mathbf{x} = \mathbf{Q}\mathbf{b} + \mathbf{P}^T\hat{\mathbf{x}}, \tag{21}$$

that give us a relation between $\hat{\mathbf{x}}$ and $\mathbf{x}$.

Deflated CG Method

To obtain the solution of the linear system (9), we solve the deflated system:

$$\mathbf{PA}\hat{\mathbf{x}} = \mathbf{Pb}. \tag{22}$$

with the CG method, for a deflated solution $\hat{\mathbf{x}}$. Thereafter, the solution $\mathbf{x}$ of the original system is obtained from (21):

$$\mathbf{x} = \mathbf{Qb} + \mathbf{P}^T\hat{\mathbf{x}}.$$

Deflated PCG Method

The deflated linear system can also be preconditioned by an *SPD* matrix $\mathbf{M}$.
The deflated preconditioned system that will be solved with CG is ((Tang, 2008, pag. 30)):

$$\tilde{\mathbf{P}}\tilde{\mathbf{A}}\hat{\hat{\mathbf{x}}} = \tilde{\mathbf{P}}\tilde{\mathbf{b}},$$

where:

$$\tilde{\mathbf{A}} = \mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{M}^{-\frac{1}{2}}, \qquad \hat{\hat{\mathbf{x}}} = \mathbf{M}^{\frac{1}{2}}\hat{\mathbf{x}}, \qquad \tilde{\mathbf{b}} = \mathbf{M}^{-\frac{1}{2}}\mathbf{b}$$

This method is called the Deflated Preconditioned Conjugate Gradient *DPCG* method, and the error is bounded by:

$$||\mathbf{x} - \mathbf{x}^{i+1}||_{\mathbf{A}} \leq 2||\mathbf{x} - \mathbf{x}^0||_{\mathbf{A}} \left(\frac{\sqrt{\kappa_{eff}(\mathbf{M}^{-1}\mathbf{PA})} - 1}{\sqrt{\kappa_{eff}(\mathbf{M}^{-1}\mathbf{PA})} + 1} \right)^{i+1},$$

with $\kappa_{eff} = \frac{\lambda_{max}(M^{-1}PA)}{\lambda_{min}(M^{-1}PA)}$, the effective condition number and $\lambda_{min}(M^{-1}PA)$ the smallest non-zero eigenvalue of $M^{-1}PA$.

Choices of Deflation Vectors

The deflation method is used to treat the most unfavorable eigenvalues of $\mathbf{A}$. If the matrix $\mathbf{Z}$ contains eigenvectors corresponding to the unfavorable eigenvalues, the convergence of the iterative method is achieved faster. However, to obtain and to apply the eigenvectors is costly in most of the cases. Therefore, a good choice of the matrix $\mathbf{Z}$ that does not contain the eigenvectors is essential for the acceleration of the convergence.
A good choice of the deflation vectors is usually problem dependent. Available information on the system is, in general, used to obtain these vectors. Most of the techniques used to choose deflation vectors are based on approximating eigenvectors, recycling (M. Clemens and Weiland (2004)), subdomain deflation vectors (C. Vuik (2002)) or multigrid and multilevel based deflation techniques (J.M. Tang and Erlangga (2009); B. Smith (1996)). A summary of these techniques is given below.

Recycling Deflation. A set of vectors previously used is reused to build the deflation-subspace matrix (M. Clemens and Weiland (2004)). The vectors could be, for example, $q-1$ solution vectors of the linear system with different right-hand sides or of different time steps. The matrix $\mathbf{Z}$ containing this solutions will be:

$$\mathbf{Z} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(q-1)}].$$

Subdomain Deflation. The domain is divided into several subdomains, each corresponding to one or more deflation vectors. For each subdomain, there is a deflation vector that has ones for points in the subdomain and zeros for points outside (C. Vuik (2002)).

Multi Grid and Multilevel Deflation. For the multigrid and multilevel methods, there are matrices called prolongation and restriction matrices that allow us to pass from one level or grid to another. These matrices are used as the deflation-subspace matrices $\mathbf{Z}$ (J.M. Tang and Erlangga (2009)).

Examples (Optional)

This is the first sentence of the example section.

Results (Optional)

Numerical experiments.

These experiments are performed in order to understand the behavior of the deflation method and to find good deflation vectors for the given problems.
In the present section, we give a general overview of the experiments that we performed, but the specifications are presented below for each problem separately.
The reference solution is obtained with a direct solution method and plotted. This solution is visually compared with the results obtained with DICCG to see if the approximation is similar.
The experiments are performed to simulate flow through porous media with a porosity of 0.2. An incompressible single-phase model is studied for a fluid with the following properties:

- $\mu = 1cpoise$
- $\rho = 1014kg/m^3$

The rock permeability varies in each case. For the first set of problems, Dirichlet boundary conditions are used for a layered model with various contrast in permeability between layers. In the second set of problems, we used Neumann boundary conditions (no-flux). We investigate the layered problem of the previous section and the SPE 10 benchmark. For the latter problem, we take the 2^{nd} layer, where the variation of permeabilities is of 7 orders of magnitude and then we used the complete model that consists of 85 layers. For the wells, we use the Peaceman model.

The model
We model incompressible single-phase flow through a porous media. The equation describes this flow is

$$\mathbf{Tp} = \mathbf{q}.$$

In these experiments, a Cartesian grid with different grid sizes is used. Wells or sources are added to the system. Neumann and Dirichlet boundary conditions are imposed. For some problems, a pressure drop is imposed in the y direction (Dirichlet boundary condition), and for others, the no-flux (Neumann) boundary condition is used. More specifications are presented below for each problem. The matrices corresponding to the linear systems $\mathbf{A}$ and right-hand sides $\mathbf{b}$ are obtained with MRST (Lie (2013)).

The solver
The solution of the system is approximated with ICCG (Conjugate Gradient preconditioned with Incomplete Cholesky) and DICCG (Deflated Conjugate Gradient preconditioned with Incomplete Cholesky).
For the DICCG method, the deflation vectors are chosen as solutions of the system with various well configurations and boundary conditions. For each configuration, we solve the system with ICCG and we use these solutions as deflation vectors. The tolerance or stopping criterium is taken as the 2-norm of the residual for the k^{th} iteration divided by the 2-norm of the right-hand side of the preconditioned system:

$$\frac{||\mathbf{M}^{-1}r^k||_2}{||\mathbf{M}^{-1}b||_2} \leq \varepsilon.$$

The stopping criterium is varied for each problem. In particular, for the SPE10 problem the dependence on the accuracy of the snapshots for the DICCG method is studied.

Snapshots
As mentioned above, for the DICCG method we need a set of deflation vectors, that in our case are solutions of the system with various well configurations and boundary conditions. This solutions called snapshots are obtained with ICCG, the tolerance used is given for each problem. The choice of the configuration of the snapshots is related to the problem that we are solving. For each case, the configuration of the snapshots as well as the configuration of the system to solve are presented.

Case 1, Dirichlet and Neumann boundary conditions.
In the configuration of *Case 1*, four wells are positioned one-third apart from each other and from the boundaries. Two wells have a bottom hole pressure (bhp) of 5 bars, and two have a bhp of -5 bar. No-flux conditions are used in the right and left boundaries and a pressure drop in the vertical direction. The pressure in the lower boundary ($y = 1$) is 0 bars, and in the upper boundary ($y = ny$) is 3 bars. The first four snapshots ($z_1 - z_4$) are obtained setting only one well different from zero, taking no-flux conditions in the right and left boundaries and homogeneous Dirichlet conditions in the other boundaries. A fifth snapshot is obtained setting all the wells to zero and setting a pressure drop in the vertical direction used for the original system. A summary is presented below.

Configuration 1:

$System\ configuration$

W1 = W2 = -5 bars.

W3 = W4 = +5 bars.

$Boundary\ conditions$:

$P(y=1) = 0$ bars, $P(y=ny) = 3$ bars, $\frac{\partial P(x=1)}{\partial n} = \frac{\partial P(x=nx)}{\partial n} = 0.$

$Snapshots$

$\mathbf{z}_1$: W1 = -5 bars, W2 = W3 = W4 = 0,

$\mathbf{z}_2$: W2 = -5 bars, W1 = W3 = W4 = 0.

$\mathbf{z}_3$: W3 = +5 bars, W1 = W3 = W4 = 0.

$\mathbf{z}_4$: W4 = +5 bars, W1 = W2 = W3 = 0.

Boundary conditions for the first 4 snapshots:

$P(y=1) = P(y=ny) = 0$ bars, $\frac{\partial P(x=1)}{\partial n} = \frac{\partial P(x=nx)}{\partial n} = 0.$

$\mathbf{z}_5$: W1 = W2 = W3 = W4 = 0.

Boundary conditions for the 5th snapshot:

P(y=1) = 0 bars, P(y=ny) = 3 bars, $\frac{\partial P(x=1)}{\partial n} = \frac{\partial P(x=nx)}{\partial n} = 0.$

As mentioned above, we studied flow through a porous medium with *heterogeneous permeability* layers. A grid of $nx = ny = 64$ elements is studied. We use 8 layers of the same size, 4 layers with one value of permeability σ_1, followed by a layer with a different permeability value σ_2. The permeability of one set of layers is set to $\sigma_1 = 1mD$, the permeability of the other set σ_2 is changed. Therefore, the contrast in permeability between the layers ($\frac{\sigma_2}{\sigma_1} = \frac{\sigma_2}{1mD}$), depends on the value of σ_2.
We investigate the dependence on the contrast in permeability value between the layers for the ICCG

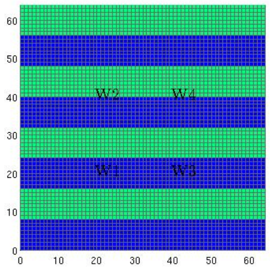


Figure 1: Heterogeneous permeability.

and DICCG methods. The permeability of one set of layers is $\sigma_1 = 1mD$ in all cases, whereas the permeability of the other set of layers varies from $\sigma_2 = 10^{-1}mD$ to $\sigma_2 = 10^{-7}mD$. Figure 1 shows these layers. The tolerance is set as 10^{-11} for the snapshots as well as for the original problem.

Table 1 shows the condition number for the matrix A, the preconditioned matrix $(M^{-1}A)$ and the deflated and preconditioned matrix $(M^{-1}PA)$ for various permeability contrasts between the layers.

σ_2 (mD)	10^{-1}	10^{-3}	10^{-5}	10^{-7}
$\kappa(A)$	2.6×10^3	2.4×10^5	2.4×10^7	2.4×10^9
$\kappa(M^{-1}A)$	206.7	8.3×10^3	8.3×10^5	8.3×10^7
$\kappa_{eff}(M^{-1}PA)$	83.27	6×10^3	1×10^6	6×10^7

Table 1: Table with the condition number for various permeability contrasts between the layers, grid size of 32 x 32, $\sigma_1 = 1mD$.

Table 2 shows the number of iterations required to achieve convergence for ICCG and DICCG, for various permeability contrasts between the layers.[4]
The solution is not reached when the value of permeability is greater than $\sigma_2 = 10^{-3}$ for both methods (ICCG and DICCG). For the ICCG method the solution is completely different from the solution obtained with a direct solver, whereas for the DICCG the solution is similar, but not the same. The plot of the residual and the solution to the problem are presented in Figures 2 and 3 for a value of permeability $\sigma_2 = 10^{-3}$.
As long as the correct solution is achieved ($\sigma_2 = 10^{-1}, 10^{-3}$), we note that the number of iterations increases when the contrast between the permeability layers increases for ICCG. For DICCG, we observe that we only need one iteration despite the change in permeability contrast between the layers.

σ_2 (mD)	10^{-1}	10^{-3}	10^{-5}	10^{-7}
ICCG	75	110	22*	22*
DICCG	1	1	10	4**

Table 2: Table with the number of iterations for different contrasts in the permeability of the layers for the ICCG and DICCG methods.

To better understand the cases when the solution has not been achieved, we study the error of the approximations. If we want the relative error $e = \frac{||\mathbf{x}-\mathbf{x}^k||_2}{||\mathbf{x}||_2}$, with $\mathbf{x}$ the true solution and $\mathbf{x}^k$ the approximation, to

[4] The * means that the solution is not achieved (from the plot of the solution), see Figure 7 for an example. Meanwhile, ** means that the solution is close to the solution obtained with a direct solver.

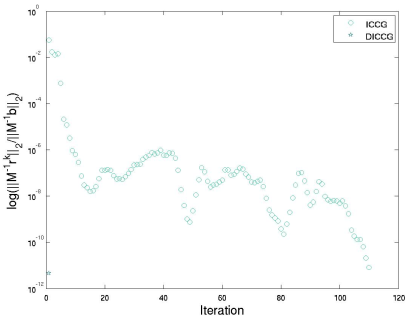


Figure 2: Convergence for the heterogeneous problem, 64 x 64 grid cells, $\sigma_2 = 10^{-3} mD$.

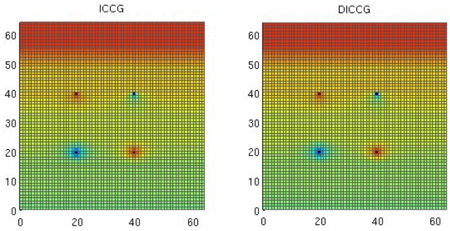


Figure 3: Solution of the heterogeneous problem, 64 x 64 grid cells,$\sigma_2 = 10^{-3} mD$.

be less than 10^{-7}, we need to choose a stopping criteria (tol) so that $tol = e/\kappa_2(M^{-1}A)$ for the preconditioned system, and $tol = e/\kappa_{eff}(M^{-1}PA)$ for the deflated and preconditoned system (see Apendix 1). The tolerance is presented in Table 3 for the ICCG and DICCG methods. We observe that the required tolerance (tol) when $\sigma_2 = 10^{-5} mD, 10^{-7} mD$ is smaller than our choice $(tol = 10^{-11})$. As a consequence, we cannot expect to find a correct solution for these cases.

σ_2 (mD)	10^{-1}	10^{-3}	10^{-5}	10^{-7}
$tol = \frac{e}{\kappa(M^{-1}A)}$	5×10^{-9}	1×10^{-10}	1×10^{-12}	1×10^{-14}
$tol = \frac{e}{\kappa_{eff}(M^{-1}PA)}$	1×10^{-8}	2×10^{-10}	1×10^{-12}	2×10^{-14}

Table 3: Table with the tolerance for various permeability contrasts between the layers, grid size of 32 x 32, $\sigma_1 = 1 mD$.

As we see from Table 3, the tolerance required when $\sigma_2 = 10^{-5} mD$ is in the order of 10^{-12}. Therefore, if we use this tolerance we can find the solution. Tolerance for the snapshots as well as for the solver is varied. We study 3 cases, in the first case, the tolerance of the snapshots and the solvers is 10^{-12}. In the second case, the tolerance of the snapshots is reduced to 10^{-14}. Finally, in case 3 the tolerance of the solvers is reduced to 10^{-13}. Table 4 shows the number of iterations necessary to achieve convergence for ICCG and DICCG for a given tolerance (Tol solver). The solution obtained with all the methods is visually compared with the solution obtained with a direct solver.
For ICCG, the solution obtained in the first two cases is similar to the expected solution but it is not the same which indicates that a tolerance of $10^{-}13$ is necessary to find the solution. For DICCG, in the first case, the solution is reached after eleven iterations. If we increase the accuracy of the snapshots (Case 2) we reach the correct solution for DICCG in one iteration (the dependence on the accuracy of the snapshots for the DICCG method is studied for the SPE10 model in the next section). If we increase the accuracy of the solvers (Case 3), the solution is also achieved, but we note that for the deflation methods the number of iterations increases.

Case	1	2	3
Tol snapshots	1×10^{-12}	1×10^{-14}	1×10^{-14}
Tol solver	1×10^{-12}	1×10^{-12}	1×10^{-13}
ICCG	55**	55**	94
DICCG	11	1	2

Table 4: Number of iterations for ICCG, DICCG for a layered problem, $\sigma_1 = 1 mD$, $\sigma_2 = 10^{-5} mD$, grid size of 64x 64.

Case 2, Neumann boundary conditions only.

In this case, four wells are positioned in the corners and have a bhp of -1 bar. One well is positioned in the center of the domain and has a bhp of +4 bars (see Figure 4). We set Neumann boundary conditions in all boundaries. The snapshots ($z_1 - z_4$) are obtained giving a value of zero to one well and non zero values to the other wells. A summary of the configurations is presented below.

Configuration 2:

System configuration

W1 = W2 = W3 = W4 = -1 bar.

W5 = +4 bars.

Boundary conditions :

$$\frac{\partial P(y=1)}{\partial n} = \frac{\partial P(y=ny)}{\partial n} = \frac{\partial P(x=1)}{\partial n} = \frac{\partial P(x=nx)}{\partial n} = 0.$$

Snapshots

$\mathbf{z}_1$: W1 = 0 bars, W2 = W3 = W4 = -1 bars, W5 = b5 = +3 bars.

$\mathbf{z}_2$: W2 = 0 bars, W1 = W3 = W4 = -1 bars, W5 = b5 = +3 bars.

$\mathbf{z}_3$: W3 = 0 bars, W1 = W3 = W4 = -1 bars, W5 = b5 = +3 bars.

$\mathbf{z}_4$: W4 = 0 bars, W1 = W2 = W3 = -1 bars, W5 = b5 = +3 bars.

Boundary conditions

$$\frac{\partial P(y=1)}{\partial n} = \frac{\partial P(y=ny)}{\partial n} = \frac{\partial P(x=1)}{\partial n} = \frac{\partial P(x=nx)}{\partial n} = 0.$$

Heterogeneous permeability layers

As in the previous case, single-phase flow through a porous medium with heterogeneous permeability layers is studied. A grid of $nx = ny = 64$ elements is investigated. The deflation vectors used in this case are 4 ($\mathbf{z}_1$-$\mathbf{z}_4$).

The snapshots and the solutions are obtained with a tolerance of 10^{-11}.

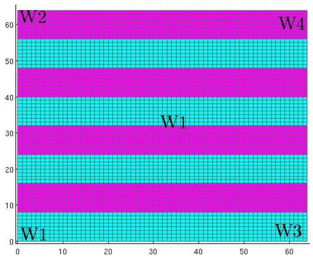


Figure 4: Heterogeneous permeability.

Table 5 shows the number of iterations required to achieve convergence for ICCG and DICCG. The plot of the residual and the solution of the problem are presented in Figure 5 and 6 for the ICCG and DICCG methods. Figure 7 shows an example when the solution is not the same with ICCG or DICCG and a direct solver.

As in Case 1, we note that the solution is not achieved when the permeability coefficient σ_2 is smaller than 10^{-3}. Studying solely the correct solutions for ICCG, the number of iterations increases as the contrast in the permeability increases. For the DICCG method, convergence is reached whithin one iteration.

σ_2	10^{-1}	10^{-3}	10^{-5}	10^{-7}
ICCG	90	131	65*	64*
DICCG	1	1	1*	1*

Table 5: Table with the number of iterations for different contrast in the permeability of the layers for the ICCG and DICCG methods, tolerance of solvers and snapshots 10^{-11}.

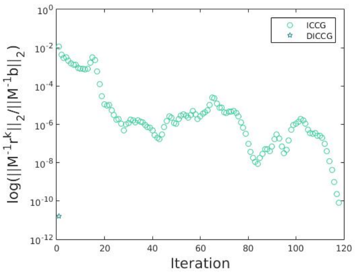


Figure 5: Convergence for the heterogeneous problem, 64 x 64 grid cells, $\sigma_2 = 10^{-3}$.

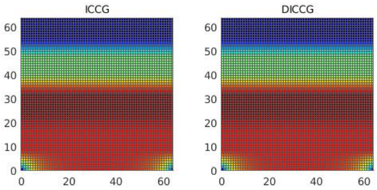


Figure 6: Solution of the heterogeneous problem, 64 x 64 grid cells, $\sigma_2 = 10^{-3}$.

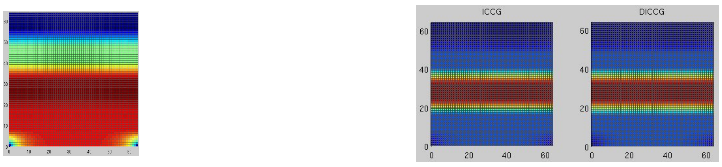


Figure 7: Example of solution not reached, heterogeneous problem $\sigma_2 = 10^{-7}$, left direct solver, right ICCG and DICCG, 64 x 64 grid cells.

SPE 10 model (1 layer)

This model has large variations in the permeability coefficients. It contains 60 x 220 x 85 cells, for these studies only one layer (2nd) is used (60 x 220 x 1 cells).

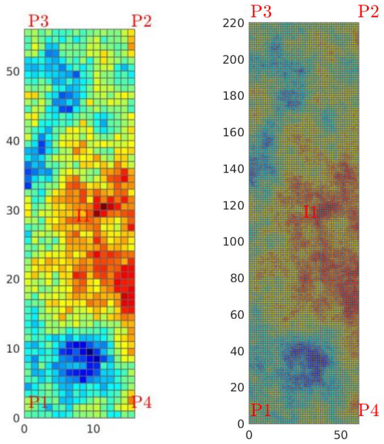
This model has 5 sources or wells, four producers in the corners (negative) and one injector in the center (positive).

The snapshots are obtained solving the system with different well configurations (*Configuration 2*). Four snapshots are used as deflation vectors. The dependence of the DICCG method on the accuracy of the snapshots is investigated. Snapshots are obtained with different accuracy, the values of tolerance used are 10^{-1}, 10^{-3}, 10^{-5}, and 10^{-7}. The original system is solved with an accuracy of 10^{-7}.

Different grid sizes are studied: 16 x 56, 30 x 110, 46 x 166 and 60 x 220.

Permeability is upscaled averaging the permeability in each grid using the harmonic-arithmetic average from MRST. The permeability of the coarser grid (16 x 56 cells) is shown in Figure 8. The permeability contrast for the diverse grid size problems is shown in Table 6. From this table, we observe that the contrast in the permeability for different grid sizes varies slightly, moreover, the order of magnitude remains the same for all the cases.

The condition number for the coarse grid problem (16 x 56) is studied in Table 7, from where we observe an important reduction in the condition number for the preconditioned the system, and a further reduction when the system is deflated.



(a) 16 x 56 grid cells (b) 60 x 220 grid cells

Figure 8: Permeability field, 16 x 56 and 60 x 220 grid cells.

Grid size	16 x 56	30 x 110	46 x 166	60 x 220
Contrast ($\times 10^7$)	1.04	2.52	2.6	2.8

Table 6: Table with the contrast of permeabilities for different grid sizes.

Condition number	value
$\kappa(A)$	2.2×10^6
$\kappa(M^{-1}A)$	377
$\kappa_{eff}(M^{-1}PA)$	82.7

Table 7: Table with the condition number of the SPE10 model, grid size of 16 x 56.

The number of iterations required to achieve convergence with the ICCG and DICCG methods for various grid sizes and various accuracy values of the snapshots are presented in Table 8. The convergence and the solution to the ICCG and DICCG methods with a tolerance of the snapshots of 10^{-7} are presented in Figure 9 and Figure 10.

For ICCG, the required iterations to reach convergence increases as the size of the grid increases. We note that the accuracy of the snapshots is important for the DICCG method. When the accuracy is low, the DICCG method beahaves almost as the ICCG method, as the accuracy increases, the number of iterations diminishes. We observe that, if we use the same accuracy for the snapshotts and the solver, 10^{-7}, the solution is reached within one iteration.

Tol	Method	16 x 56	30 x 110	46 x 166	60 x 220
	ICCG	34	73	126	159
10^{-1}	DICCG	33	72	125	158
10^{-3}	DICCG	18	38	123	151
10^{-5}	DICCG	11	21	27	55
10^{-7}	DICCG	1	1	1	1

Table 8: Table with the number of iterations for ICCG and DICCG, various tolerance for the snapshots, various grid sizes.

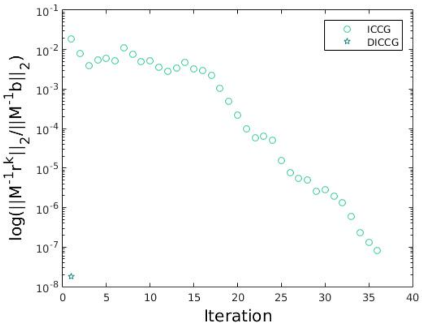


Figure 9: Convergence for the SPE10 problem, 16 x 56 grid cells, accuracy of the snapshots 10^{-7}.

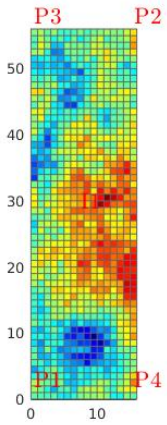


Figure 10: Solution of the SPE10 benchmark, 16 x 56 grid cells, 2nd layer, accuracy of the snapshots 10^{-7}.

SPE 10 complete

We approximate the solution for the complete SPE10 benchmark (60 x 220 x 65) with the ICCG and DICCG methods (see Figure 11). We use four deflation vectors, $z_1 - z_4$ ($Configuration$ 2). The accuracy of the snapshots is varied from 10^{-2} to 10^{-11} in steps of 10^{-3}. For the solution, accuracy is varied from 10^{-5} to 10^{-11} in steps of 10^{-3}. Results are presented in Table 9.

The convergence plot is presented in Figure 12 and the solution in Figure 13 for an accuracy of the snapshots and solvers of 10^{-11}.

We observe from Table 9, that the number of iterations necessary to achieve convergence is the same for ICCG and DICCG if the accuracy of the snapshots is low (tol 10^{-2}). When the accuracy improves, the number of iterations is reduced for DICCG. Convergence is achieved within one iteration for DICCG when the accuracy of the snapshots is 10^{-11}. Which is a significant improvement with respect to ICCG that needs 1029 iterations to achieve the solution.

We can also observe that, if the accuracy of the solver is not enough, the correct solution is not reached.

Tol. sol	10^{-5}	10^{-8}	10^{-11}
Tolerance of snapshots 10^{-2}			
ICCG	164*	604*	1029
DICCG	162*	608*	1029
Tolerance of snapshots 10^{-5}			
ICCG	165*	609*	1029
DICCG	1*	467**	879
Tolerance of snapshots 10^{-8}			
ICCG	165*	609*	1030
DICCG	1*	1*	559
Tolerance of snapshots 10^{-11}			
ICCG	165*	610*	1030
DICCG	1	1	1

Table 9: Table with the number of iterations for different accuracy in the snapshots and the solution with the ICCG and DICCG methods.

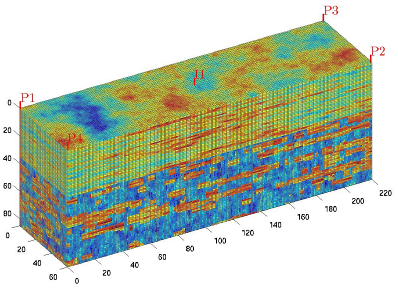


Figure 11: SPE10 benchmark, permeability field.

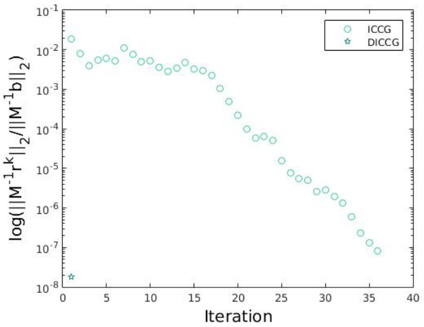


Figure 12: Convergence for ICCG and DICCG, SPE10 model, 85 layers, accuracy of the snapshots 10^{-11}.

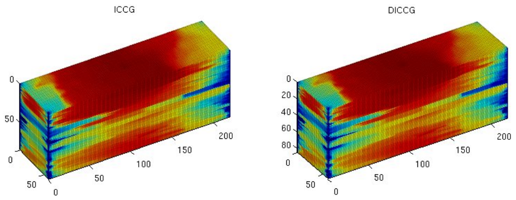


Figure 13: Solution to the SPE10 model (85 layers) with ICCG and DICCG, tolerance of the snapshots and solvers 10^{-11}.

Conclusions

A proposal to select physics-based deflation vectors for the Deflated Conjugated Gradient preconditioned with Incomplete Cholesky (DICCG) method was studied. Results show that the choice of deflation vectors is important for a good performance of the method. They also show that the accuracy of the snapshots used as deflation vectors is essential for the correct performance of the DICCG method. For the layered problem, we note that the contrast between the layers modifies the condition number and demands a higher accuracy for the solvers. When the required accuracy is not satisfied, the solution is not reached. For the cases when the correct accuracy is used, the DICCG method reaches the solution within one iteration. Results also expose that the behaviour of the solver DICCG does not depend on the grid size or the contrast when a correct accuracy for the snapshots and the solver is used.

Appendix 1. Stopping criteria

When we use an iterative method, we always want that our approximation is close enough to the exact solution. In other words, we want that the error (Saad, 2003, pag. 42):

$$||\mathbf{e}^k||_2 = ||\mathbf{x} - \mathbf{x}^k||_2,$$

or the relative error:

$$\frac{||\mathbf{x} - \mathbf{x}^k||_2}{||\mathbf{x}||_2},$$

is small.
When we want to chose a stopping criteria, we could think that the relative error is a good candidate, but is has the disadvantage that we need to know the exact solution to compute it. What we have instead is the residual

$$\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k,$$

that is actually computed in each iteration of the CG method. There is a relationship between the error and the residual that can help us with the choice of the stopping criteria.

$$\frac{||\mathbf{x} - \mathbf{x}^k||_2}{||\mathbf{x}||_2} \leq \kappa_2(A)\frac{||\mathbf{r}^k||_2}{||\mathbf{b}||_2}.$$

With this relationship in mind, we can choose the stopping criteria as an ε for which

$$\frac{||\mathbf{r}^k||_2}{||\mathbf{b}||_2} \leq \varepsilon.$$

But we should keep to have in mind the condition number of the matrix $\mathbf{A}$, because the relative error will be bounded by:

$$\frac{||\mathbf{x} - \mathbf{x}^k||_2}{||\mathbf{x}||_2} \leq \kappa_2(A)\varepsilon.$$

Acknowledgements (Optional)

This is the first sentence of the acknowledgements.

References

B. Smith, P. BjÃÿrstad, W.G. [1996] *Domain decomposition: parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press New York.

C. Vuik, A.S. and Meijerink, J.A. [1999] An Efficient Preconditioned CG Method for the Solution of a Class of Layered Problems with Extreme Contrasts in the Coefficients. *Journal of Computational Physics*, **152**, 385.

C. Vuik, A. Segal, L.Y.E.D. [2002] A comparison of various deflation vectors applied to elliptic problems with discontinuous coefficients. *Applied Numerical Mathematics*, **41**(1), 219–233.

Jansen, J.D. [2013] *A systems description of flow through porous media.* Springer.

J.M. Tang, R. Nabben, C.V. and Erlangga, Y. [2009] Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *Journal of scientific computing*, **39**(3), 340–370.

Lie, K. [2013] *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST).* SINTEF ICT.

M. Clemens, M. Wilke, R.S. and Weiland, T. [2004] Subspace projection extrapolation scheme for transient field simulations. *IEEE Transactions on Magnetics*, **40**(2), 934–937.

P. Astrid, G. Papaioannou, J.C.V.J.J. [2011] Pressure Preconditioning Using Proper Orthogonal Decomposition. In: *2011 SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA*.

R. Markovinović, J.D.J. [2006] Accelerating iterative solution methods using reduced-order models as solution predictors. *International journal for numerical methods in engineering*, **68**(5), 525–541.

Saad, Y. [2003] *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics Philadelphia, PA, USA. 2nd edn.

Tang, J. [2008] *Two-Level Preconditioned Conjugate Gradient Methods with Applications to Bubbly Flow Problems*. Ph.D. thesis, Delft University of Technology.