Introduction

In this report, we study fast and robust iterative solvers for large systems of linear equations resulting from reservoir simulation trough porous media. We propose the use of preconditioning and deflation techniques based on information obtained from the system to reduce the time spent in the solution of the linear system.

We consider incompressible single-phase flow in a layered model with large variations in the permeability coefficients 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 end with 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 **?**.

Darcy's law is given by:

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

And the mass balance equation is given by:

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

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:

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

with T_0 a constant reference temperature.

The accumulation term $\frac{\partial (\rho \phi)}{\partial t}$ of Equation (2) can be rewritten as:

$$\frac{\partial (\rho \phi)}{\partial t} = \rho \frac{\partial \phi}{\partial t} + \phi \frac{\partial \rho}{\partial t} = \rho \frac{\partial \phi}{\partial p} \frac{\partial p}{\partial t} + \phi \frac{\partial \rho}{\partial p} \frac{\partial p}{\partial t} = \phi \rho \left[\frac{1}{\phi} \frac{\partial \phi}{\partial p} + \frac{1}{\rho} \frac{\partial \rho}{\partial p} \right] \frac{\partial p}{\partial t},$$

combining the equation above with Equation (3) and (4) we have:

$$\frac{\partial(\rho\phi)}{\partial t} = \phi\rho\left[c_r + c_l\right]\frac{\partial p}{\partial t} = \phi\rho c_t\frac{\partial p}{\partial t}, \tag{5}$$

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 and t. Assuming isotropic permeability ($k = K_x = K_y = K_z$), small rock and fluid compressibilities, uniform reservoir thickness and absence of gravity forces Equation (**??**) can be rewritten as:

$$-\frac{h}{\mu}\frac{\partial}{\partial x}\left(k\frac{\partial p}{\partial x}\right) - \frac{h}{\mu}\frac{\partial}{\partial y}\left(k\frac{\partial p}{\partial y}\right) - \frac{h}{\mu}\frac{\partial}{\partial y}\left(k\frac{\partial p}{\partial z}\right) + \phi_0 c_t\frac{\partial p}{\partial t} - hq = 0. \tag{7}$$

Well Model

In reservoirs, wells are typically drilled to extract or inject fluids. If the injection or production rates are known it is necessary to know the well pressure. Meanwhile, if the well pressure is known, the objective will be to compute the flow rate in or out of the reservoir. 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.

Close to a well, the pressure presents nonlinear gradients in the radial direction. In reservoir simulation, it is possible to capture this effect with a very fine grid. It is also possible to use models that take into account an additional pressure drop based on analytical or semi-analytical solutions for the flow around the well. A widely used model is Peacemans's model, that takes into account the above-mentioned pressure drop. This is a logarithmic relationship, where the pressure drop is due to a steady-state radial flow towards a well in the center of a cell. Taking the radial distance from a production well as r operating at bottom-hole pressure in an homogeneous reservoir with permeability K, and a fluid viscosity μ, the pressure drop between the cell pressure p and the well bottom-hole pressure p_{well} in a cell of height h is given by **??**:

$$p - p_{well} = -\frac{\mu q}{2\pi Kh}ln\left(\frac{r}{r_{well}}\right) = -\frac{q}{J_{well}}, \tag{8}$$

that is obtained from Darcy's law. The negative value of the flow rate q indicates production, while, a positive value is for injection. J_{well} is known as the well index or productivity index (PI) for a producer, and well injectivity index (WI) for injectors. The model presented in Equation (8) is valid only for a restricted set of assumptions, but it has proven to be very useful basis model for simple vertical wells.

Incompressible flow.

If the fluid and rock compressibilities are small enough, they can be ignored and equation (**??**) becomes:

$$\mathbf{Tp} = \mathbf{q}. \tag{9}$$

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:

$$a_{11}x_1 + \cdots + a_{1n}x_n = b_1,$$
$$\ldots$$
$$a_{n1}x_1 + \ldots + a_{nn}x_n = b_n,$$

which is written in matrix form as:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{10}$$

Matrix $\mathbf{A}$ can have a structure that might be useful while solving the linear system, examples of structure are:

Symmetry: $\mathbf{A}^T = \mathbf{A}$.

Band: $a_{ij} \neq 0$ if $i - m_l \leq j \leq i + m_u$, with m_l, m_u two non-negative integers, and $a_{ij} = 0$ elsewhere.

Positive Definiteness: $(\mathbf{A}\mathbf{x}, \mathbf{x}) > 0, \qquad \forall \mathbf{x} \in \mathbb{R}^n, \qquad \mathbf{x} \neq 0$.

Symmetric Positive Definiteness *SPD*. If $\mathbf{A}$ is symmetric and Positive Definite.

System (10) can be 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. The convergence of an iterative method is not always guaranteed.

Some of the iterative methods that can be used are: Jacobi, Gauss Seidel, and if the matrix is *SPD* we can use the Conjugate Gradient (CG) method. This section is devoted to iterative methods, in particular CG, that is the method that we are going to use in our experiments (see Section).

In this section, we also describe the Preconditioning and Deflation techniques for the acceleration of the CG method.

Krylov subspace Methods

If we have two subspaces $\mathcal{K}_k$, $\mathcal{L}_k$ of $\mathbb{R}^n$ and we want to solve the Equation (10), with $\mathbf{A} \in \mathbb{R}^{n \times n}$ we can use a projection method onto $\mathcal{K}_k$. 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 $\mathbf{r}^0$,

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

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

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

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

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

The subspace $\mathcal{L}_k$ 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 Symmetric Positive Definite *SPD* matrices, such that

$$||\mathbf{x} - \mathbf{x}^k||_{\mathbf{A}}, ^1 \tag{11}$$

is minimal, with $\mathbf{x}$ the solution of the system and $\mathbf{x}^k$ the $k - th$ iteration.

Given an initial guess $\mathbf{x}^0$ the next approximations can be computed following the search directions $\mathbf{p}^i$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k.$$

---

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

For the CG method, the search directions $\mathbf{p}^k$ are orthogonal with respect to the $\mathbf{A}$ inner product, i.e.

$$(\mathbf{A}\mathbf{p}^k, \mathbf{p}^j) = 0, \qquad k \neq j,$$

and the residuals form an orthogonal set, i.e.

$$(\mathbf{r}^k, \mathbf{r}^j) = 0, \qquad k \neq j.$$

The constant α^k that satisfies (11) is given by:

$$\alpha^k = \frac{(\mathbf{r}^k, \mathbf{r}^k)}{(\mathbf{A}\mathbf{p}^k, \mathbf{p}^k)},$$

and the new search directions can be computed via the residuals,

$$\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta_k \mathbf{p}^k,$$

where

$$\beta_k = \frac{(\mathbf{r}^{k+1}, \mathbf{r}^{k+1})}{(\mathbf{r}^k, \mathbf{r}^k)}.$$

After $k+1$ iterations of the *CG* method, the error of the iteration will be bounded by:

$$||\mathbf{x} - \mathbf{x}^{k+1}||_{\mathbf{A}} \leq 2||\mathbf{x} - \mathbf{x}^0||_{\mathbf{A}} \left(\frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \right)^{k+1} 2. \tag{12}$$

Preconditioning

If we want to accelerate the convergence of an iterative method, we can transform the system into another one containing a better spectrum. This can be done by multiplying the original system (10) by a matrix $\mathbf{M}^{-1}$.

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}. \tag{13}$$

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

$$||\mathbf{x} - \mathbf{x}^{k+1}||_{\mathbf{A}} \leq 2||\mathbf{x} - \mathbf{x}^0||_{\mathbf{A}} \left(\frac{\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})} + 1} \right)^{k+1}. \tag{14}$$

$\mathbf{M}$ 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 **?**.

Given an *SPD* matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the deflation matrix $\mathbf{P}$ is defined as follows **?**:

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

where

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

with

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

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 (**?**, pag. 27):

---

[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, $\kappa_2(\mathbf{A}) = \frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})}$.

a) $\mathbf{E}^T = \mathbf{E}$,   *symmetric*.

b) $\mathbf{Q}^T = \mathbf{Q} = \mathbf{QAQ}$,   *symmetric*.

c) $\mathbf{QAZ} = \mathbf{Z}$.

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

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

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

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

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

i) $\mathbf{P}^T \mathbf{Z} = \mathbf{0}^{n \times m}$.

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

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

$$\mathbf{x} = \mathbf{Ix} - \mathbf{P}^T\mathbf{x} + \mathbf{P}^T\mathbf{x} = (\mathbf{I} - \mathbf{P}^T)\mathbf{x} + \mathbf{P}^T\mathbf{x}. \tag{15}$$

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

$$
\begin{aligned}
\mathbf{Ax} &= \mathbf{A}(\mathbf{I} - \mathbf{P}^T)\mathbf{b} + \mathbf{AP}^T\mathbf{x}, &\qquad Property: \\
\mathbf{Ax} &= \mathbf{AQb} + \mathbf{AP}^T\mathbf{x}, &\qquad g) \\
\mathbf{b} &= \mathbf{AQb} + \mathbf{PAx}, &\qquad f),
\end{aligned}
$$

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

$$
\begin{aligned}
\mathbf{Pb} &= \mathbf{PAQb} + \mathbf{P}^2\mathbf{Ax}, \\
\mathbf{Pb} &= \mathbf{PAx}, \tag{16}
\end{aligned}
$$

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

$$\mathbf{Pb} = \mathbf{PA}\hat{\mathbf{x}}. \tag{17}$$

As mentioned above, solution of Equation (16) can contain components of $\mathscr{N}(\mathbf{PA})$. Thereafter, the solution of Equation (17), $\hat{\mathbf{x}}$ can be decompose as:

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{y}, \tag{18}$$

with $\mathbf{y} \in \mathscr{R}(\mathbf{Z}) \subset \mathscr{N}(\mathbf{PA})$, and $\mathbf{x}$ solution of Equation (10).
Note. If $\mathbf{y} \in \mathscr{R}(\mathbf{Z})$, then

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

$$\mathbf{PA}\mathbf{y} = \mathbf{PA}(\mathbf{z}_1 \alpha_1 + ... + \mathbf{z}_m \alpha_m) = \mathbf{PAZ}\alpha, \tag{19}$$

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

$$\mathbf{PA}\mathbf{y} = 0. \tag{20}$$

---

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

Therefore $\mathscr{R}(\mathbf{Z}) \subset \mathscr{N}(\mathbf{PA})$.

Multiplying Equation 18 by $\mathbf{P}^T$ we obtain:

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

combining Equation (20) with f), we have:

$$\mathbf{PA}y = \mathbf{AP}^T y = 0.$$

Therefore

$$\mathbf{P}^T \hat{\mathbf{x}} = \mathbf{P}^T \mathbf{x}. \tag{21}$$

Substitution of Equation (21) and g) in Equation (15) leads to:

$$\mathbf{x} = \mathbf{Qb} + \mathbf{P}^T \hat{\mathbf{x}}, \tag{22}$$

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

Deflated CG Method

To obtain the solution of the linear system (10), we solve the deflated system:

$$\mathbf{PA}\hat{\mathbf{x}} = \mathbf{Pb}. \tag{23}$$

with the CG method, for a deflated solution $\hat{\mathbf{x}}$. Thereafter, the solution $\mathbf{x}$ of the original system is obtained from (22):

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

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 (**?**, pag. 30):

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

where:

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

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 **?**, subdomain deflation vectors **?** or multigrid and multilevel based deflation techniques **??**. A summary of these techniques is given below.

Recycling Deflation. A set of vectors previously used is reused to build the deflation-subspace matrix **?**. 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 **?**.

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 ($\mathbf{Z}$) **?**.

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. A single-phase model is studied for a fluid with the following properties:

- $\mu = 1 cpoise$

- $\rho = 1014 kg/m^3$

The rock permeability varies in each case. For the first set of problems, homogeneous permeability is used. Then, we use layers with different permeabilities. Finally, we use the SPE 10 model, which has large permeability variations (7-12 orders of magnitude). For our experiments, first we take the 2nd layer of this model, where the variation of permeabilities is of 7 orders of magnitude. Then we used the complete model that consists of 85 layers. For the wells, we use the Peaceman model, described in Section .

The model

This part is devoted to incompressible flow model $\mathbf{Tp} = \mathbf{q}$ (see Section **??**).

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 b are obtained with the Matlab Reservoir Simulation Toolbox (MRST) **?**.

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 kth 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 with one layer, 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 method, the tolerance used is given for each problem. For all the experiments we perform, we use one of the following configuration of wells and the snapshots described below.

Configuration 1

For the *Configuration 1*, four wells are positioned one-third apart from each other and from the boundaries. The first four snapshots ($z_1 - z_4$) are obtained setting only one well different from zero and 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 drop pressure in the vertical direction. The system to solve will be the system containing all the wells with values different from zero and the Dirichlet conditions as used for the fifth snapshot. A summary of the snapshots is presented below.

Configuration 1:

Boundary conditions for the first 4 snapshots:
$P(y = 1) = P(y = ny) = 0$ bars, $\frac{\partial P(x=1)}{\partial n} = \frac{\partial P(x=nx)}{\partial n} = 0$.

$\mathbf{z}_1$: 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 5th snapshot:
P(y=1) = 0 bars, P(y=ny) = 3 bars, $\frac{\partial P(x=1)}{\partial n} = \frac{\partial P(x=nx)}{\partial n} = 0$.

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

System configuration
Boundary conditions:
$P(y = 1) = 0 bars, P(y = ny) = 3$ bars, $\frac{\partial P(x=1)}{\partial n} = \frac{\partial P(x=nx)}{\partial n} = 0.$
W1 = -5 bars, W2 = +5 bars, W3 = +5 bar, W4 = -5 bar.

Configuration 2

Four wells are positioned in the corners and one in the center of the domain (see Figure). The snapshots $(z_1 - z_5)$ are obtained solving the problem with various well configurations. The first four snapshots $(z_1 - z_4)$ are obtained giving a value of zero to one well and non zero values to the other wells. The last snapshot (z_5) has the same configuration as the system that we want to solve. For this configuration we use Neumann boundary conditions, thereafter the sum of the values of the wells must be equal to zero for each snapshot. The set of snapshots is presented below.

Configuration 2:

$\mathbf{z}_1$: 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.

System configuration

$\mathbf{z}_5$: W1 = W2 = W3 = W4 = -1 bars, W5 = b5 = +4 bars.

Homogeneous permeability

For the first set of experiments, flow through porous medium with a homogeneous permeability is studied. The permeability is $\sigma = 1 mD$ in the whole domain.
The snapshots are obtained with ICCG method and the stopping criteria or tolerance is given for each problem. For Case 1, the snapshots are obtained as described in *Configuration 1* and for Case 2 as presented in *Configuration 2*.

Case 1
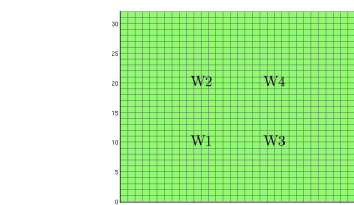
r6cm   Homogeneous permeability, *Configuration 1* for the wells. We impose no-flux boundary conditions for the vertical boundaries and a pressure difference at the horizontal boundaries. The size of the grid is $nx = ny = 32$, 64 and 128 (see Figure). The snapshots are obtained as presented in *Configuration 2*, with ICCG and a tolerance of 10^{-8}. The original system is also solved with a tolerance of 10^{-8}.

Permeability 1 mD.

Grid size $nx \times ny$ grid cells, $nx = ny = 32$, 64 and 128.

W1 = W4= -5 bars.

W2 = W3 = +5 bars.

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

Results

The condition number is computed in Table 1 for the matrix A, the preconditioned matrix $(M^{-1}A)$ and the deflated and preconditioned matrix $(M^{-1}PA)$ for the smallest problem (32 x 32 grid cells). We observe, for the preconditioned and the preconditioned and deflated systems, a significant reduction in the condition number with respect to the original system.

Table 2 shows the number of iterations necessary to achieve convergence for the ICCG and DICCG

$\kappa_2(A)$	931.6
$\kappa_2(M^{-1}A)$	37.9
$\kappa_{eff}(M^{-1}PA)$	22.9

Table 1 *Table with the condition number for the smallest problem (32 x 32 cells).*

methods for problems with various grid sizes. The plot of the residual $\left(\frac{||M^{-1}r^k||_2}{||M^{-1}b||_2} \right)$ is presented in Figure 1 and the solution in Figure 2.

We observe from Table 2 an increment in the number of iterations necessary to achieve convergence for the ICCG method when the grid size increases. On the contrary, we note that the number of iterations remains unchanged for the DICCG method. Moreover, the convergence is achieved in one iteration for this method, as expected from Lemma 2. The plot of residual $\left(\frac{||M^{-1}r^k||_2}{||M^{-1}b||_2} \right)$ is presented in Figure 1, we observe that the residual reaches a value in the order of 10^{-8} after the first iteration for the DICCG method, whereas the residual is diminishing from a value close to 10^{-2} until the desired value of 10^{-8} for ICCG.

Grid size	32 x 32	64 x 64	128 x 128
	Iterations	Iterations	Iterations
ICCG	22	41	77
DICCG	1	1	1

Table 2 *Table with the number of iterations for different grid sizes for the ICCG and DICCG methods.*
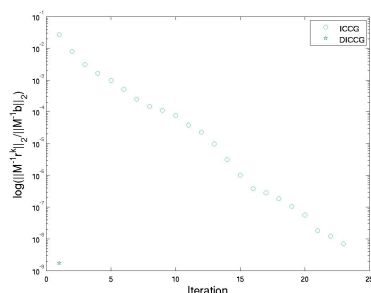
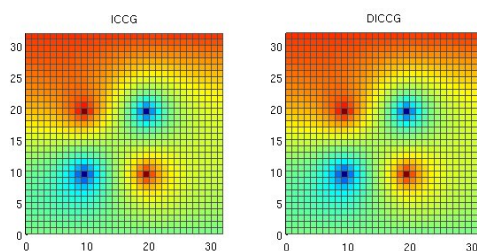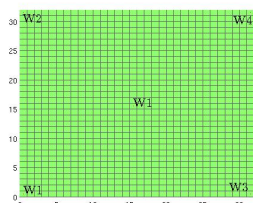**Figure 1** *Convergence for the homogeneous problem, 32 x 32 grid cells.*

Figure 2 *Solution of the homogeneous problem, 32 x 32 grid cells.*

Case 2

r5cm Homogeneous permeability, *Configuration 2* for the wells. We impose no-flux boundary conditions in all boundaries (Neumann). The snapshots are obtained solving the problem with various well configurations as presented in *Configuration 2*. The value of the wells is presented below. The tolerance used is 10^{-8} for the snapshots and the original problem.

Grid size $nx \times ny$ grid cells, $nx = ny = 32$, 64 and 128.

Permeability 1 mD.

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

W5 = +4 bars.

Neumann boundary conditions.

Results

Table 3 shows the number of iterations necessary to achieve convergence for the ICCG and DICCG methods, with a tolerance of 10^{-8}.

We observe that using solutions of the problem with various well configurations as deflation vectors (*Configuration 2*), the convergence is achieved in only one iteration with the DICCG method, as expected from theory (Lemma 2). As in the previous case we note that the number of iterations increases with the size of the problem for the ICCG method. For the smallest case (32 x 32 grid cells) the plot of the residual and the solution of the problem are presented in Figure 3 and 4.

Grid size	32 x 32	64 x 64	128 x 128
Method	Iterations	Iterations	Iterations
ICCG	26	45	77
DICCG	1	1	1

Table 3 *Table with the number of iterations for different grid sizes for the ICCG and DICCG methods.*

Examples (Optional)
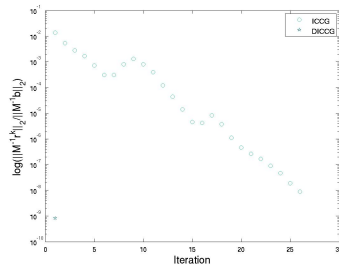
This is the first sentence of the example section.

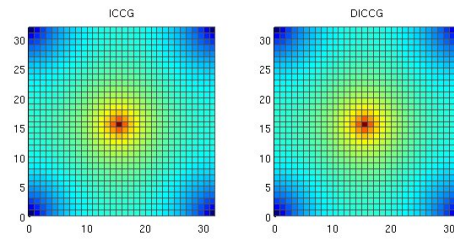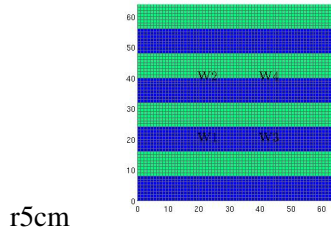**Figure 3** *Convergence for the homogeneous problem, 32 x 32 grid cells.*

Figure 4 *Solution of the homogeneous problem, 32 x 32 grid cells.*

Results (Optional)

r5cm   Heterogeneous permeability.   In the second set of experiments, we studied flow through porous media with *heterogeneous permeability*. 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 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 shows these layers. The tolerance is set as 10^{-11} for the snapshots as well as for the original problem.

Case 1

The conditions are the same as in *Case* 1 in the *homogeneous permeability* problem. In this section, only a grid of $nx = ny = 64$ elements is studied. Table 4 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 4 *Table with the condition number for various permeability contrasts between the layers, grid size of 32 x 32, $\sigma_1 = 1mD$.*

Results

Table 5 shows the number of iterations necessary 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 that $\sigma_2 = 10^{-3}$ for both methods (ICCG and DICCG). For the ICCG method the solution is completely different from the solution obtained with a linear solver, whereas for the DICCG the solution is similar, but not the same. The plot of the residual and the solution of the problem are presented in Figure 5 and 6 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 of permeability contrast between the layers, as expected from theory (Lemma 2).

To better understand the cases when the solution is not reached, 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 be less than 10^{-7}, we need to choose a stopping criteria (tol) such 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 **??**).

---

[4]The * means that the solution is not achieved (from the plot of the solution), see Figure 9 for an example. Meanwhile, ** means that the solution is close to the solution obtained with a direct solver.

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

Table 5 *Table with the number of iterations for different contrasts in the permeability of the layers for the ICCG and DICCG methods.*
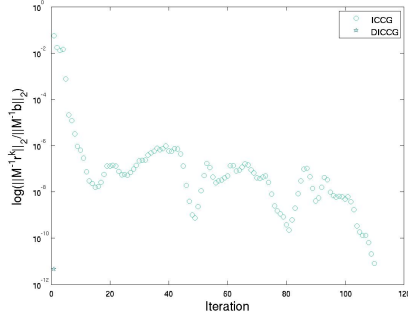

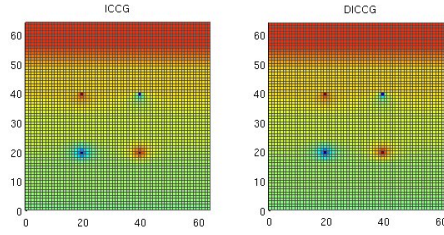**Figure 6** *Solution of the heterogeneous problem, 64 x 64 grid cells, $\sigma_2 = 10^{-3}mD$.*

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

The tolerance is presented in Table 6 for the ICCG and DICCG methods. We observe that the needed 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 6 *Table with the tolerance for various permeability contrast between the layers, grid size of 32 x 32, $\sigma_1 = 1mD$.*

As we see from Table 6, the tolerance needed when $\sigma_2 = 10^{-5}mD$ is of the order of 10^{-12}. Therefore, if we use this tolerance we can reach the solution. Table 7 shows the number of iterations necessary to achieve convergence for the ICCG, DICCG and DICCG$_d$ for a given tolerance (Tol solver). For the first deflation method (DICCG), the deflation vectors are the snapshots obtained with ICCG using the tolerance showed in the table (Tol. snapshots). For the second deflation method (DICCG$_d$), the deflation vectors are the snapshots obtained with a direct 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 is not the same. For DICCG, in the first case, the solution is reached after eleven iterations, whereas for the DICCG$_d$ it is reached within one iteration. 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 reached, 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
$DICCG_d$	1	1	2

Table 7 *Number of iterations for ICCG, DICCG for a layered problem,* $\sigma_1 = 1mD$, $\sigma_2 = 10^{-5}mD$, *grid size of 64x 64, .*
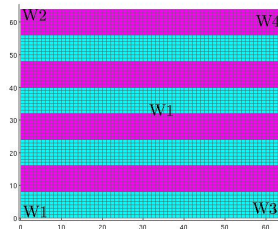
Case 2

r5cm   Heterogeneous permeability.   The conditions are the same as in the Case 2 for homogeneous permeability. For this problem only a grid of $nx = ny = 64$ elements is investigated. The deflation vectors used in this case are 4 (z_1-z_4) and 5 (z_1-z_5).
The snapshots and the solutions are obtained with a tolerance of 10^{-11}.

Results
Table 8 shows the number of iterations necessary to achieve convergence for ICCG and DICCG with 4 and 5 deflation vectors. The tolerance is 10^{-11} for the snapshots for the original problem. The plot of the residual and the solution of the problem are presented in Figure 7 and 8 for the ICCG and DICCG methods, the last one with 4 deflation vectors. Figure 9 shows an example when the solution is not the same with ICCG or DICCG and a direct solver.
As in the 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 the ICCG method, the number of iterations increases as the contrast in the permeability increases. For the DICCG method, with for deflation vectors, the convergence is achieved in one iteration as expected from theory (Lemma 2). If we use 5 deflation vectors, although the convergence is achieve in 1 iteration when the permeability coefficient is $\sigma_2 = 10^{-1}$, it is not the case when it is $\sigma_2 = 10^{-3}$, as expected. This might be caused by some errors related to the choice of deflation vectors. Therefore is important to use only the correct number of deflation vectors. Techniques such as Proper Orthogonal Decomposition (POD) **??** could help us to chose the correct set of deflation vectors.

σ_2	10^{-1}	10^{-3}	10^{-5}	10^{-7}
ICCG	90	131	65*	64*
$DICCG_4$	1	1	1*	1*
$DICCG_5$	1	500*	500*	500*

Table 8 *Table with the number of iterations for different contrast in the permeability of the layers for the ICCG and DICCG methods, tolerance of 10^{-11}, snapshots 10^{-11}. $DICCG_4$ is the solution with 4 deflation vectors and $DICCG_5$ is the solution with 5 deflation vectors.*
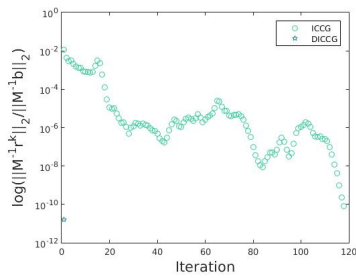

Figure 7 *Convergence for the heterogeneous problem, 64 x 64 grid cells, $\sigma_2 = 10^{-3}$.*
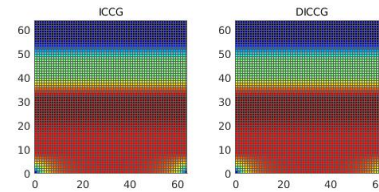
Figure 8 *Solution of the heterogeneous problem, 64 x 64 grid cells, $\sigma_2 = 10^{-3}$.*
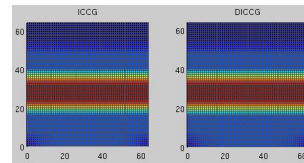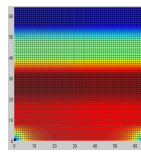

Figure 9 *Example of solution not reached, heterogeneous problem $\sigma_2 = 10^{-7}$, left direct solver, right ICCG and DICCG, 64 x 64 grid cells.*

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*). The first 4 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.

The permeability is upscaled averaging the permeability in each grid using the function *sampleFromBox* from MRST. The permeability of the coarser grid (16 x 56 cells) is shown in Figure 10. The permeability contrast for the diverse grid size problems is shown in Table 9. 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 10. We observe an important reduction in the condition number for the preconditioned the system, and a further reduction when the system is deflated.
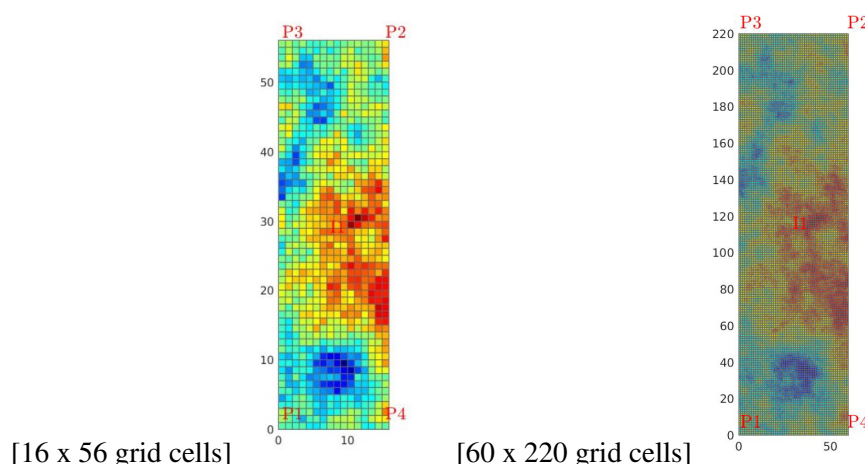

[16 x 56 grid cells]   [60 x 220 grid cells]

Figure 10 *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 9 *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 10 *Table with the condition number of the SPE10 model, grid size of 16 x 56.*

The number of iterations neccesary to achieve convergence with the ICCG and DICCG methods for different grid sizes and different accuracy for the snapshots are presented in Table 11. The convergence and the solution for the ICCG and DICCG methods with a tolerance for the snapshots of 10^{-7} are presented

in Figure 11 and Figure 12.

For the ICCG method, the necessary 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, as expected from theory (Lemma 2).

Tol	Method	16 x 56	30 x 110	46 x 166	60 x 220
	ICCG	34	73	126	159
10^{-1}	DICCG$_4$	33	72	125	158
10^{-3}	DICCG$_4$	18	38	123	151
10^{-5}	DICCG$_4$	11	21	27	55
10^{-7}	DICCG$_4$	1	1	1	1

Table 11 *Table with the number of iterations for ICCG and DICCG, various tolerance for the snapshots, various grid sizes.*
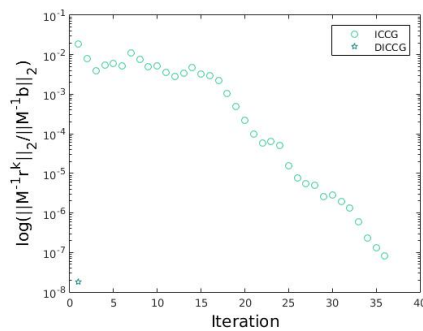

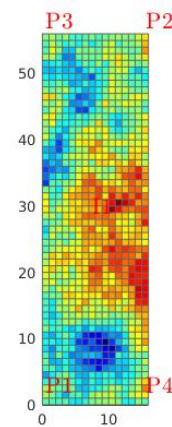**Figure 11** *Convergence for the SPE10 problem, 16 x 56 grid cells, accuracy of the snapshots 10^{-7}.*

Figure 12 *Solution of the SPE10 benchmark, 16 x 56 grid cells, 2nd layer, accuracy of the snapshots 10^{-7}.*

We approximate the solution for the complete SPE10 benchmark (60 x 220 x 65) with the ICCG and DICCG methods (see Figure 13). We use four deflation vectors, $z_1 - z_4$ from the *Configuration* 2. The accuracy for the snapshots and the solution is 10^{-11}.

Convergence is achieved within one iteration for DICCG (see Table 12), as expected from theory (Lemma 2). We note a significant improvement with respect to ICCG that needs 1029 iterations to achieve the solution.

The convergence plot is presented in Figure 14 and the solution in 15.

Method	Iterations
ICCG	1029
DICCG$_4$	1

Table 12 *Number of iterations for the SPE10 model (85 layers) for the ICCG and DICCG methods, tolerance of the snapshots and solvers* 10^{-11}.
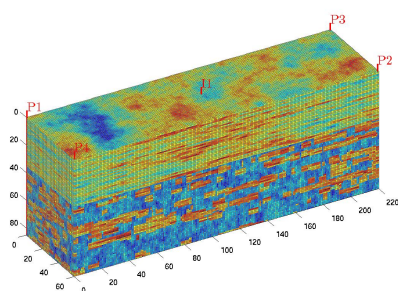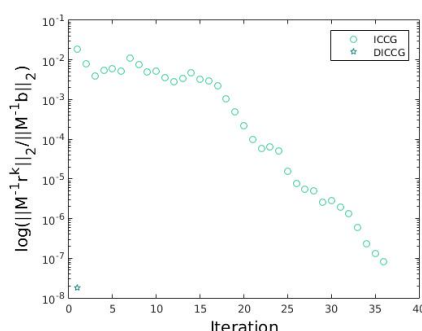

Figure 13 *SPE10 benchmark, permeability.*

Figure 14 *Convergence for the SPE10 model, 85 layers, accuracy of the snapshots* 10^{-11}.
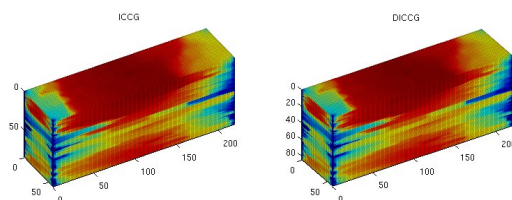

Figure 15 *Solution of the SPE10 model (85 layers) with ICCG and DICCG, tolerance of the snapshots and solvers* 10^{-11}.

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 desired 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 does not depend on the grid size (SPE 10 example) or the contrast (Heterogeneous permeability example). This is the first sentence of the conclusions.

Acknowledgements (Optional)

This is the first sentence of the acknowledgements.