



# Efficient solution techniques for two-phase flow in heterogeneous porous media using exact Jacobians

Henrik Büsing<sup>1</sup>

Received: 3 August 2019 / Accepted: 6 August 2020 / Published online: 10 October 2020  
© The Author(s) 2020

## Abstract

Two efficient and scalable numerical solution methods will be compared using exact Jacobians to solve the fully coupled Newton systems arising during fully implicit discretization of the equations for two-phase flow in porous media. These methods use algebraic multigrid (AMG) to solve the linear systems in every Newton step. The algebraic multigrid methods rely on (i) a Schur Complement Reduction (SCR-AMG) and (ii) a Constrained Pressure Residual method (CPR-AMG) to decouple elliptic and hyperbolic contributions. Both methods employ automatic differentiation (AD) to calculate exact Jacobians and are compared with finite difference (FD) approximations of the Jacobian. The superiority of AD is shown by several numerical test cases from the field of CO<sub>2</sub> geo-sequestration comprising two- and three-dimensional examples. A weak scaling test on JUQUEEN, a BlueGene/Q supercomputer, demonstrates the efficiency and scalability of both methods. To achieve maximum comparability and reproducibility, the Portable Extensible Toolkit for Scientific Computation (PETSc) is used as framework for the comparison of all solvers.

**Keywords** Algebraic multigrid (AMG) · Schur complement reduction (SCR-AMG) · Constrained pressure residual (CPR-AMG) · Multiphase flow in porous media · Automatic differentiation (AD) · CO<sub>2</sub> geo-sequestration

**Mathematics Subject Classification (2010)** 65F08 · 65M55 · 76T10

## 1 Introduction

Applications for two-phase flow in porous media are found in numerous scientific fields, e.g., geothermal energy, the remediation of groundwater contamination by non-aqueous phase liquids (NAPLs), hydrogen emission due to barrel corrosion during nuclear waste management, enhanced oil recovery where oil is produced by the injection of water, and production of oil and gas fields as well as geological storage of CO<sub>2</sub>. In geothermal energy, high-enthalpy reservoirs may contain water as liquid and vapor. Successful production

requires an assessment of the behavior of the phases over time. Remediation of NAPLs during groundwater contamination may involve the injection of hot steam. This reduces the viscosity of the NAPLs and supports their easier transport out of the reservoir. Here, groundwater, injected steam, and the NAPL are the existing phases. During nuclear waste storage, brines may corrode the storage containers, finally resulting in the emission of H<sub>2</sub>. Enhanced oil recovery makes use of water or CO<sub>2</sub> injection to increase the oil yield. In this scenario, water, oil, and possibly CO<sub>2</sub> are the existing phases. In this context, the petroleum industry created the first numerical multiphase flow simulators, described by Douglas Jr. et al. as early as 1959 [17]. Both Aziz and Settari [1] and Chavent and Jaffré [13] summarize in detail the equations for flow and transport in petroleum reservoirs.

In the test examples, the focus is on the geological sequestration of CO<sub>2</sub>. Here, brine-saturated sandstone reservoirs at depths greater than 800 m may provide a safe storage site for supercritical CO<sub>2</sub>. Safety assessments require a prediction of the propagation of the CO<sub>2</sub> plume in the subsurface. Given a safe storage site, the

---

The research leading to these results has received funding from the European Union's Horizon2020 Research and Innovation Program under grant agreement No. 640573 (Project DESCRAMBLE) and No. 676629 (Project EoCoE)

---

✉ Henrik Büsing  
hbusing@eonerc.rwth-aachen.de

<sup>1</sup> Institute for Applied Geophysics and Geothermal Energy, E.ON Energy Research Center, RWTH Aachen University, 52074 Aachen, Germany

geological sequestration of CO<sub>2</sub> is one option to mitigate anthropogenic effects of greenhouse gas emissions on our climate. Compared with the other applications, this example is particularly challenging because of the high injection rates and the fact that CO<sub>2</sub> is injected but no other fluids are produced. Both aspects result in high reservoir overpressures.

The fully implicit discretization of the fully coupled formulation for two-phase flow equations yields a system of nonlinear algebraic equations. Newton's method is used to linearize this system. This requires solving a system of linear equations in each time step and each Newton step. This linear system is often extremely ill-conditioned and asymmetric and couples strongly the different physical quantities, e.g., pressure and saturation.

Historically, ILU preconditioned methods were used to solve these systems due to their general applicability. Unfortunately, they are neither necessarily efficient nor scalable (cf. [40] Chapter 10.3 on ILU factorization preconditioners). As an alternative, direct methods are known for their robustness and reliability. However, they also require a considerable amount of computation, on the order of  $O(n^2)$ , and memory, on the order of  $O(n^{4/3})$  where  $n$  is the number of unknowns (cf. [33]). In contrast, multigrid methods promise linear complexity for certain problems and thus belong to the most efficient class of methods (cf. [24]).

Since the exact solution of the Newton system is not required, iterative methods such as GMRES [41], BiCGStab [46] or FGMRES [39] are used which approximate the solution only to a certain accuracy. The most time-consuming parts of the numerical simulation are the computation of the Jacobian and the subsequent solution of the resulting linear systems. We focus our attention on these two computational kernels to optimize the solution time.

The linear system contains both hyperbolic and almost elliptic properties. The properties of the matrix are described in, e.g., [6, 16, 29] as well as [45] and summarized by [10].

Two-stage preconditioners, such as the Constrained Pressure Residual method [47, 48], decouple these contributions from each other resulting in preconditioners implicit in pressure and explicit in saturation (IMPES). In a first stage, the pressure equation is solved. The overall solution is then updated with the result from this stage and in a second stage, the total system is solved for the remaining saturation variables. The elliptic subpart can be solved efficiently by AMG, followed by an ILU-based solution of the full system. IMPES schemes may impose severe restrictions on time step size due to the explicit handling of the saturation equation. Jenny et al. [27], propose a sequential fully implicit (SFI) multi-scale finite volume (MSFV) method to avoid these restrictions.

Kayum et al. [28], compare various CPR-AMG strategies together with different decoupling and preconditioning strategies, such as Alternate Block Factorization (ABF), Quasi IMPES (QI) and Dynamic RowSum (DRS). Other approaches making use of AMG are presented by Stüben et al. [45], who discuss strategies for solving fully implicit formulations that possess the elliptic properties required by AMG. Additionally, they present an iterative coupling scheme that is faster and also feasible for AMG as an alternative to fully implicit formulations. Mishev et al. [34], use multiplicative and additive overlapping Schwarz preconditioners together with AMG, while Gries [21] uses system-AMG with DRS preconditioning for an efficient solution of the equations in reservoir simulation. Next to these AMG-based approaches, Klíe et al. [30] present a physics-based two-stage percolation aggregation (2SPA) preconditioner and compare it with classical ILU-based preconditioning.

Recently, methods relying on a Schur Complement Reduction have also been used to solve two-phase flow equations (cf. [10]). We present a variant using AMG on the pressure field as a preconditioner and AMG as a solver on the Schur complement, preconditioned by the saturation block. In addition to AMG, geometric multigrid has also been applied to the fully coupled, fully implicit reservoir equations (cf. [5, 6]). However, we focus here on AMG due to its general applicability.

Automatic differentiation (AD) (cf. [37] and [23]) allows for an elegant, exact computation of the Jacobians during the Newton step. We use Tapenade (cf. [25]) to perform the required source-code transformation. Given a function  $\mathbf{F}$  representing the discretized PDE systems, this source-code transformation generates a function  $d\mathbf{F}$ , calculating the derivative of the function  $\mathbf{F}$ . This results from applying the chain rule: every source code can be viewed as a concatenation of basic functions, such as multiplication, addition, exponential, or sine functions. These basic functions have a simple derivative and with the chain rule the complete derivative is easily computed.

The advantage of AD over FD is the exactness of the Jacobian. No additional approximation errors are introduced as would be the case for a FD approximation. In addition, it is not necessary to choose a certain finite difference step size, which should minimize the FD error over the entire computation time. Additionally, the advantage of AD over a hand-coded exact Jacobian is its error robustness. Since the derivative of the source code is generated automatically, no errors can be introduced by manual differentiation.

Although automatic in principle, the derivative generation by AD requires some preparation of the source code, such as specifying the independent and dependent variables, as well as the derivative quantities to be computed. Depending on the code and how well-defined its interfaces are, this

may require some effort. Nevertheless, the advantages of AD outweigh these additional preparations by far.

This paper is organized as follows: In the first section, we introduce the governing equations and different parameters. Subsequently, we describe the numerical method and highlight the use of exact Jacobians by automatic differentiation (AD). The last section presents three different test cases: (i) CO<sub>2</sub> injection into a two-dimensional domain for an advection- and a diffusion-dominated test case; (ii) model 2 of the SPE10 benchmark problem, referred to as SPE10B; and (iii) CO<sub>2</sub> injection into the Sleipner reservoir. The conclusion highlights the advantages of AD, as well as the competitiveness of SCR-AMG compared with classical CPR-AMG.

In the [Appendix](#), we show the command line options for PETSc (cf. [2–4]) for selecting the different solvers. Additionally, the Buckley-Leverett problem (cf. [9]) is used for code verification. Finally, gravitational and CFL numbers are discussed for the different test examples and the influence of anisotropy is examined.

## 2 Mathematical model

The system of partial differential equations (PDEs) that governs two-phase flow in porous media consists of two mass balances, one for the wetting (*w*), water phase, and one for the non-wetting (*n*), gas phase. The choice of primary variables is typically a combination of a pressure and a saturation. We use a water pressure (*p<sub>w</sub>*), gas saturation (*S<sub>n</sub>*) formulation. Density  $\rho_\alpha$  and viscosity  $\mu_\alpha$ ,  $\alpha \in \{w, n\}$  are either constant or depend on pressure and temperature. Permeability  $\mathbb{K}$  may be either homogeneous or heterogeneous in space and either isotropic or anisotropic in different directions. Porosity  $\phi$  may be homogeneous or heterogeneous in space. Flow through the porous medium is governed by the usual extension of Darcy’s law [15] for multiphase systems by relative permeabilities  $k_{r\alpha}$ , a modification factor for the absolute permeability (cf. [7]). The volumetric flow rates  $\mathbf{v}_w$  and  $\mathbf{v}_n$  are defined as:

$$\begin{aligned} \mathbf{v}_w &= -\frac{k_{rw}}{\mu_w} \mathbb{K} (\nabla p_w - \rho_w \mathbf{g}), \\ \mathbf{v}_n &= -\frac{k_{rn}}{\mu_n} \mathbb{K} (\nabla p_n - \rho_n \mathbf{g}), \end{aligned} \tag{1}$$

with  $k_{r\alpha}$  denoting relative permeabilities,  $\mathbb{K}$  the tensor of absolute permeabilities, and  $\mathbf{g} = (0, 0, -g)^T$  gravity. The system of non-linear coupled partial differential equations then reads as such:

$$\begin{aligned} \phi \frac{\partial(\rho_w S_w)}{\partial t} + \text{div}(\rho_w \mathbf{v}_w) &= q_w \\ \phi \frac{\partial(\rho_n S_n)}{\partial t} + \text{div}(\rho_n \mathbf{v}_n) &= q_n. \end{aligned} \tag{2}$$

This system is supplemented by algebraic constraints for the saturations, which sum up to 1:

$$S_w + S_n = 1,$$

and the relation between wetting and non-wetting pressure by the capillary pressure function:

$$p_c(S_w) = p_n - p_w.$$

Applying these constraints, inserting the Darcy velocities and choosing water pressure  $p_w$  and gas saturation  $S_n$  as primary variables, we have:

$$\begin{aligned} \phi \frac{\partial(\rho_w(1 - S_n))}{\partial t} - \text{div} \left( \rho_w \frac{k_{rw}}{\mu_w} \mathbb{K} (\nabla p_w - \rho_w \mathbf{g}) \right) &= q_w \\ \phi \frac{\partial(\rho_n S_n)}{\partial t} - \text{div} \left( \rho_n \frac{k_{rn}}{\mu_n} \mathbb{K} (\nabla(p_c + p_w) - \rho_n \mathbf{g}) \right) &= q_n. \end{aligned} \tag{3}$$

### 2.1 Relative permeability and capillary pressure

The two most common approaches for modeling capillary pressure  $p_c$  were proposed by Brooks and Corey [8] and van Genuchten [20]. The Brooks-Corey model for capillary pressure:

$$p_c = p_d S_e^{-1/\lambda} \tag{4}$$

is often combined with the approach by Burdine [11] for relative permeabilities:

$$k_{rw} = S_e^{\frac{2+3\lambda}{\lambda}} \tag{5}$$

$$k_{rn} = (1 - S_e)^2 \left( 1 - S_e^{\frac{2+\lambda}{\lambda}} \right), \tag{6}$$

while the van Genuchten model for capillary pressure:

$$p_c = \frac{1}{\tau} (S_e^{-1/m} - 1)^{1/n} \tag{7}$$

is often combined with the relative permeability model after Mualem [35]:

$$k_{rw} = \sqrt{S_e} \left( 1 - (1 - S_e^{1/m})^m \right)^2 \tag{8}$$

$$k_{rn} = (1 - S_e)^{\frac{1}{3}} \left( 1 - S_e^{\frac{1}{m}} \right)^{2m}. \tag{9}$$

Here,  $S_e$  is the effective saturation:

$$S_e = \frac{S_w - S_{wr}}{1 - S_{wr} - S_{nr}}, \tag{10}$$

where  $S_{wr}$  and  $S_{nr}$  are the residual wetting and non-wetting saturations. In the Brooks-Corey model,  $p_d$  is the displacement pressure and  $\lambda$  the pore size distribution index. The displacement or entry pressure  $p_d$  is the pressure needed by the non-wetting fluid to displace the wetting fluid from the largest pore. For large and small values, the pore size distribution index  $\lambda$  corresponds to a relatively narrow or wide pore size distribution respectively.

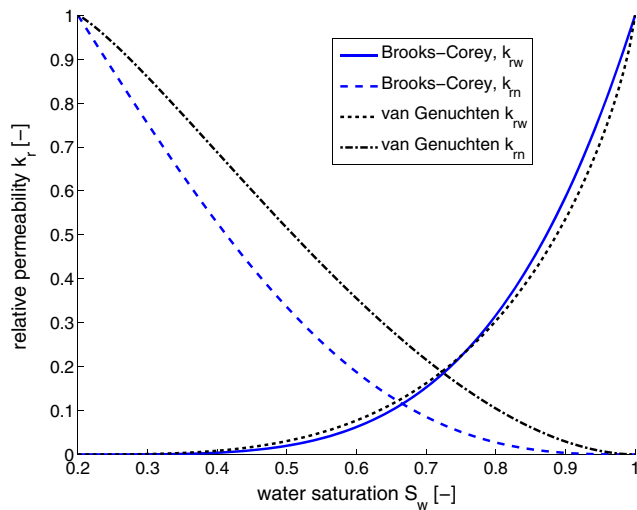


Fig. 1 Relative permeability-saturation relations after [8] and [20]

The van Genuchten model uses parameters  $\tau$ , which can be seen as an inverse entry pressure, and the parameter  $m$ , often chosen as  $m = 1 - \frac{1}{n}$ . Ideally, these parameters are determined by experiments.

Figures 1 and 2 show the relative permeability and capillary pressure functions of the two models.

### 3 Numerical method using exact Jacobians

We use the implicit Euler method as time discretization and a cell-centered finite volume method as space discretization. Let  $u_\alpha = \phi \rho_\alpha (\delta_{\alpha w} + (-1)^{\delta_{\alpha w}} S_n)$  and  $\mathbf{v}_\alpha = -\lambda_\alpha \mathbb{K} (\nabla p_w + \delta_{\alpha n} \nabla p_c - \rho_\alpha \mathbf{g})$ , where  $\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}$  is the mobility of phase  $\alpha$  and  $\delta$  is the Kronecker delta. The domain  $\Omega$  is discretized

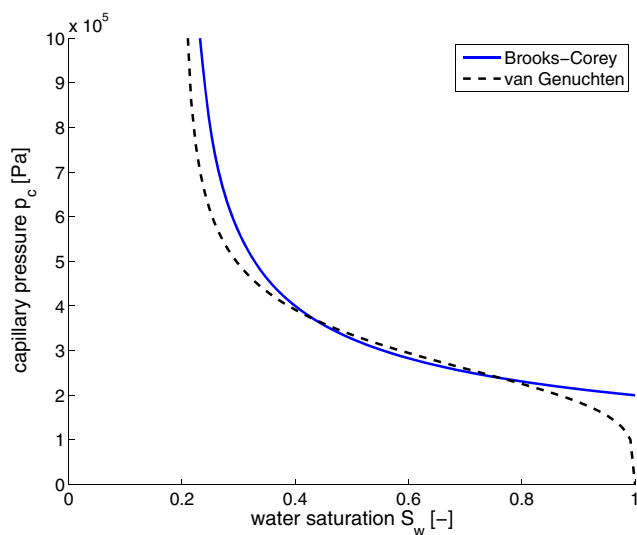


Fig. 2 Capillary pressure after [8] and [20]

into cells  $V_i$ . For a cell  $V_i$ , the integral form of Eq. 3 reads as:

$$\frac{d}{dt} \int_{V_i} u_\alpha dV + \int_{V_i} \text{div}(\rho_\alpha \mathbf{v}_\alpha) dV = \int_{V_i} q_\alpha dV \quad \alpha \in \{w, n\}, \quad (11)$$

with  $q_\alpha$  representing source and sink terms of the (non)-wetting phase. Using Green’s theorem, we transform the volume integral into a surface integral:

$$\int_{V_i} \text{div}(\rho_\alpha \mathbf{v}_\alpha) dV = \int_{\partial V_i} \rho_\alpha \mathbf{v}_\alpha \cdot \mathbf{n} d\Gamma.$$

Applying the midpoint rule for the volume integrals, a two-point flux approximation for the boundary integral, and the implicit Euler method for the time derivative, we have:

$$\begin{aligned} & \frac{u_{\alpha,i}^{n+1} - u_{\alpha,i}^n}{\Delta t} |V_i| \\ & + \sum_j \left( \rho_\alpha \frac{k_{r\alpha}}{\mu_\alpha} \mathbb{K} \right)_{ij}^{n+1} \left\{ \frac{p_{\alpha,j} - p_{\alpha,i}}{d_i + d_j} - \rho_{\alpha,ij} g_{ij} \right\}^{n+1} |A_{ij}| \\ & - q_{\alpha,i}^{n+1} |V_i| = 0, \end{aligned} \quad (12)$$

with  $\Delta t$  the time step size of time step  $n$ ,  $|V_i|$  the volume of cell  $V_i$ ,  $|A_{ij}|$  the area of the surface between cell  $V_i$  and  $V_j$ , and  $d_i$  the distance from the center of cell  $V_i$  to the interface between cell  $V_i$  and  $V_j$ . An appropriate averaging at the interface between cell  $i$  and  $j$  is needed for quantities with indices  $ij$ . The mobilities are fully upwinded:

$$\lambda_{\alpha,ij} = \begin{cases} \lambda_{\alpha,j} & \text{if } \psi_{\alpha,j} - \psi_{\alpha,i} \geq 0 \\ \lambda_{\alpha,i} & \text{if } \psi_{\alpha,j} - \psi_{\alpha,i} < 0, \end{cases} \quad (13)$$

with  $\psi_\alpha = p_w + \delta_{\alpha n} p_c - \rho_\alpha g$ . An arithmetic average  $\rho_{\alpha,ij} = \frac{\rho_{\alpha,i} + \rho_{\alpha,j}}{2}$  is used for the densities and a harmonic average for the absolute permeability:  $\mathbb{K}_{ij} = 2 \left( \frac{1}{\mathbb{K}_i} + \frac{1}{\mathbb{K}_j} \right)^{-1}$ . We end up with a system of nonlinear equations:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \quad (14)$$

where  $\mathbf{x} = (\mathbf{p}_w, \mathbf{S}_n)^T \in \mathbb{R}^{2N}$  is the vector of unknowns, and the vector function  $\mathbf{F} = (\mathbf{F}_1, \mathbf{F}_2)^T : \mathbb{R}^{2N} \rightarrow \mathbb{R}^{2N}$  consists of the two discretized equations for the water and gas phases. Newton’s method is used for linearization of the non-linear system Eq. 14. Thus, in every Newton step, the linear system:

$$\frac{\partial \mathbf{F}(\mathbf{x}_k)}{\partial \mathbf{x}_k} \Delta \mathbf{x}_k = -\mathbf{F}(\mathbf{x}_k) \quad (15)$$

is solved, where  $\Delta \mathbf{x}_k$  equals  $\mathbf{x}_{k+1} - \mathbf{x}_k$  in the  $k^{\text{th}}$  Newton step. The Jacobian  $J := \frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}}$  is of the form:

$$J = \begin{pmatrix} \frac{\partial \mathbf{F}_1}{\partial \mathbf{p}_w} & \frac{\partial \mathbf{F}_1}{\partial \mathbf{S}_n} \\ \frac{\partial \mathbf{F}_2}{\partial \mathbf{p}_w} & \frac{\partial \mathbf{F}_2}{\partial \mathbf{S}_n} \end{pmatrix}. \quad (16)$$

The sparsity pattern of the Jacobian  $J$  can be seen in Fig. 3. The  $2 \times 2$  block structure with seven diagonals

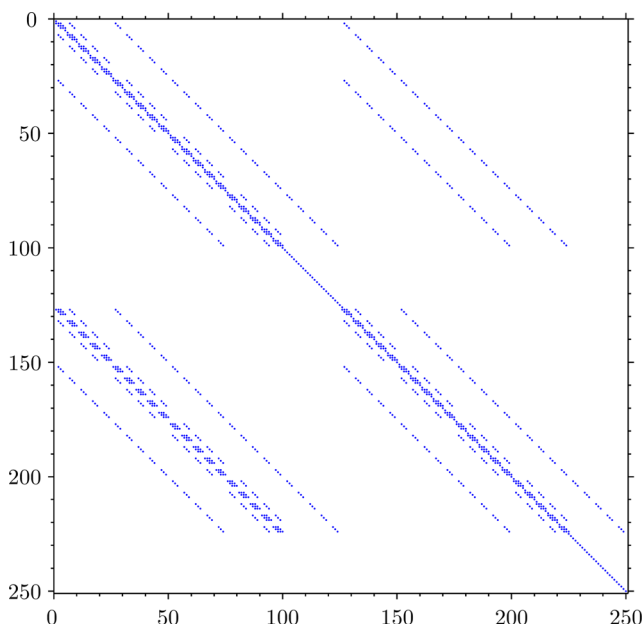


Fig. 3 Sparsity pattern of the Jacobian  $J$  with  $2 \times 2$  block structure

in each block results from the three-dimensional space discretization on a regular grid with  $5 \times 5 \times 5$  cubes and thus  $J \in \mathbb{R}^{250 \times 250}$ . The  $J_{12}$  block has only two diagonals due to the upwind scheme used. The  $J_{22}$  block has the full seven diagonals because this example has non-zero capillary pressure.

The Jacobian in Eq. 16 is modified in such a way that  $\mathbf{F}_1$  is replaced by  $\mathbf{G}_1 = \frac{1}{\rho_w} \mathbf{F}_1 + \frac{1}{\rho_n} \mathbf{F}_2$  in the first  $N$  rows. The second  $N$  rows are left as is:  $\mathbf{G}_2 = \mathbf{F}_2$ . Thus, we add Eqs. 1 and 2 of system Eq. 3. In the special case of constant densities, this eliminates the time derivatives and yields a purely elliptic pressure block  $J_{11}$ . The different entries of the Jacobian have the form:

$$\frac{\partial \mathbf{G}_1}{\partial \mathbf{p}_w} = -\text{div} \left( \left( \frac{k_{rw}(S_w)}{\mu_w} + \frac{k_{rn}(S_w)}{\mu_n} \right) \mathbb{K} \nabla \right) \tag{17}$$

$$\begin{aligned} \frac{\partial \mathbf{G}_1}{\partial \mathbf{S}_n} = & -\text{div} \left( \frac{1}{\mu_w} \frac{dk_{rw}(S_w)}{dS_n} \mathbb{K} (\nabla p_w - \rho_w \mathbf{g}) \right) \\ & -\text{div} \left( \frac{1}{\mu_n} \frac{dk_{rn}(S_w)}{dS_n} \mathbb{K} (\nabla(p_c + p_w) - \rho_n \mathbf{g}) \right) \\ & + \frac{1}{\mu_n} k_{rn}(S_w) \mathbb{K} \nabla \frac{dp_c}{dS_n} \end{aligned} \tag{18}$$

$$\frac{\partial \mathbf{G}_2}{\partial \mathbf{p}_w} = -\text{div} \left( \frac{\rho_n}{\mu_n} k_{rn}(S_w) \mathbb{K} \nabla \right) \tag{19}$$

$$\begin{aligned} \frac{\partial \mathbf{G}_2}{\partial \mathbf{S}_n} = & -\text{div} \left( \frac{\rho_n}{\mu_n} \frac{dk_{rn}(S_w)}{dS_n} \mathbb{K} (\nabla(p_c + p_w) - \rho_n \mathbf{g}) \right) \\ & + \frac{\rho_n}{\mu_n} k_{rn}(S_w) \mathbb{K} \nabla \frac{dp_c}{dS_n} \end{aligned} \tag{20}$$

To verify the correctness of the exact derivatives, we compare the Jacobian computed by finite differences (FD),

$J_{FD}$ , with the Jacobian computed by AD,  $J_{AD}$ . We vary the step size  $h$  for FD logarithmically with equidistant steps of 0.5 in the interval  $[10^{-18}, 1]$ . We compare the relative error of the Jacobian computed with AD,  $J_{AD}$ , to the finite difference (FD) Jacobian in the maximum norm for matrices  $\frac{|J_{AD} - J_{FD}|_\infty}{|J_{AD}|_\infty}$ . Figure 4 shows the relative error for the different step sizes  $h$ .

The minimal error is obtained for a step size  $h = 10^{-8}$ . This is also a very typical choice for the FD step size. Nevertheless, this step size is only optimal for the first Newton step in the first time step. Later iterations may require different optimal step sizes. Using AD, we need not worry about the choice of the step size  $h$ , as we always use the exact derivative and not an approximation.

### 3.1 Linear solvers and preconditioners

The linear solver is at the heart of the simulation code. Once the Jacobian is formed, the solution of a linear system is to be performed.

For large systems, direct methods break down due to memory consumption and runtime. Block ILU methods are in principle scalable, but the preconditioner deteriorates with the increasing number of blocks. Using ILU as a preconditioner for the full system also does not scale since the computation of the ILU decomposition is not scalable. As a consequence, we need a solver which is known to scale well, such as multigrid methods for elliptic problems. However, these methods need to be applied carefully since the coupled system is rather a degenerated parabolic/hyperbolic one than an elliptic one.

We compare two different solvers: (1) a Schur Complement Reduction method (SCR-AMG) relying on the preconditioner package Hypre [18, 19] and its AMG solver BoomerAMG [26] as preconditioner and solver and (2) a

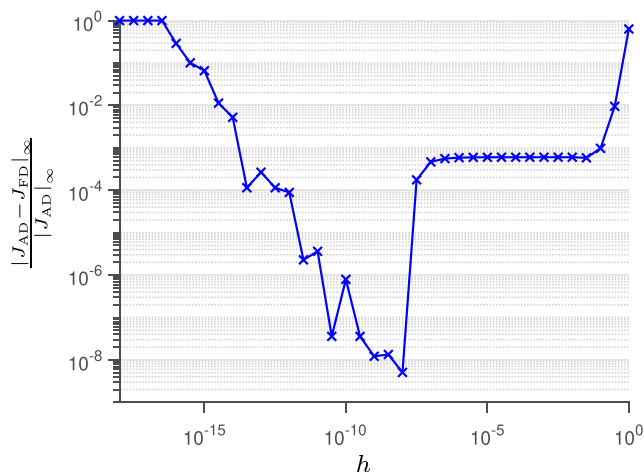


Fig. 4 Relative error between  $J_{AD}$  and  $J_{FD}$  computed with varying step sizes  $h \in [10^{-18}, 1]$

Constrained Pressure Residual method also making use of BoomerAMG for the pressure block and a block Jacobi ILU preconditioner, with no level of fill in (known as ILU(0)), for the saturation block. These two solvers are then combined with either a Jacobian formed by AD or FD.

### 3.2 Schur complement reduction

The method is implemented with the fieldsplit preconditioner from PETSc (cf. [3] Chapter 4.5 on solving block matrices). This preconditioner allows addressing the different fields, in our case pressure and saturation, of a multi-physics simulation with appropriate solvers. On the pressure field, we only apply Hypre/BoomerAMG as a preconditioner and do not use an iterative method. The Schur complement is preconditioned by the  $J_{22}$  block of the Jacobian and Hypre/BoomerAMG is used as a solver, which terminates after a maximum of 10 iterations or when the residual is decreased by two orders of magnitude.

Given a matrix  $J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \in \mathbb{R}^{2N \times 2N}$  the inverse of this matrix can be written as:

$$J^{-1} = \left[ \begin{pmatrix} I & 0 \\ J_{21} J_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} J_{11} & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I & J_{11}^{-1} J_{12} \\ 0 & I \end{pmatrix} \right]^{-1} \quad (21)$$

$$= \begin{pmatrix} I & -J_{11}^{-1} J_{12} \\ 0 & I \end{pmatrix} \begin{pmatrix} J_{11}^{-1} & 0 \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -J_{21} J_{11}^{-1} & I \end{pmatrix} \quad (22)$$

with the Schur complement  $S = J_{22} - J_{21} J_{11}^{-1} J_{12}$  of the block  $J_{11}$ . Thus, the solution of a  $2N \times 2N$  system can be reduced to the solution of two  $N \times N$  systems. For this solution method to be effective, a good preconditioner for the Schur complement matrix is needed. Note that never any of the matrices is actually inverted but rather a linear system is solved, since we do not need the actual matrix, but rather its action on a vector.

We use PETSc option “a11” and consequently  $J_{22}$  to construct the preconditioner for the Schur complement. This option is justified since all the derivatives in the Jacobian reduce to the Laplacian in case we assume all parameters to be constant. This yields an effective preconditioner for the Schur complement and consequently an overall very effective solution method. Appendix A.1 describes the different PETSc options to allow for maximum comparability and reproducibility.

### 3.3 Constrained pressure residual

In reservoir simulation, the Constrained Pressure Residual method (CPR-AMG) is often used as a solver for the occurring linear systems [22, 31, 32, 42, 45]. The idea is to apply AMG to the elliptic contributions and an ILU method to the hyperbolic part. CPR-AMG is implemented by a

composite preconditioner consisting of the fieldsplit method and a block Jacobi ILU method. The fieldsplit method is again used to separate the pressure and the saturation fields. But this time, we use it only to select the pressure field. For this field, we apply Hypre’s BoomerAMG as a solver. The saturation field is omitted. Finally, the two fields are combined in an additive way. In general, the additive fieldsplit type solves the  $J_{11}$  and  $J_{22}$  block  $\begin{pmatrix} J_{11}^{-1} & 0 \\ 0 & J_{22}^{-1} \end{pmatrix}$ . Here, the inverse indicates the solution of a (preconditioned) linear system.

Next, the fieldsplit solution is combined with a block Jacobi ILU method in a composite preconditioner of multiplicative type. Let  $P_1$  and  $P_2$  be the two preconditioners. Then, the effect of the combined preconditioner  $P$  on a vector  $x$ ,  $y = Px$ , can be obtained by calculating:

$$y = P_1 x \quad (23)$$

$$w_1 = x - A y \quad (24)$$

$$y = y + P_2 w_1. \quad (25)$$

## 4 Numerical simulations

For comparison, all methods are implemented together with PETSc (the Portable Extensible Toolkit for Scientific Computation) [2–4]. This allows a change of the linear solver via a modification of only a few command line options and ensures maximum comparability and easy reproducibility of the results. We use PETSc 3.9.0 compiled with GCC 4.8.5 and OpenMPI 1.10.4. Test cases 1 to 3 run on Intel Xeon E5-2680 processors with 2.70 GHz. Each node consists of two processors with 8 cores. Test case 1 uses one node, i.e., 16 cores, while test cases 2 and 3 use four nodes, i.e., 64 cores in total. JUQUEEN, a BlueGene/Q supercomputer, is used for the weak scaling test comprising one node (16 cores) up to 64 nodes (1024 cores). On JUQUEEN, we use PETSc development GIT revision v3.8.3-1672-gb907f15 compiled with GCC 4.8.1 and MPICH2. In this GIT revision, the non-scalable parts of PETSc’s MatCreateSubMatrix found in [12] are fixed.

SHEMAT-Suite [14, 38], a forward and inverse modeling code for the simulation of reactive flow, heat, and species transport in porous media is used as a platform for the implementation of the multiphase flow equations.

All test examples use constant fluid and gas properties. Permeability tensor is isotropic for the two-dimensional test cases and anisotropic for the Sleipner and SPE10B test case. Anisotropy factors reach values of 35 for the Sleipner case and 10,000 for the SPE10B case. CO<sub>2</sub> and brine phase are immiscible. Discontinuous capillary pressure is not considered. The simulations use rectangular grids with variable cell sizes.

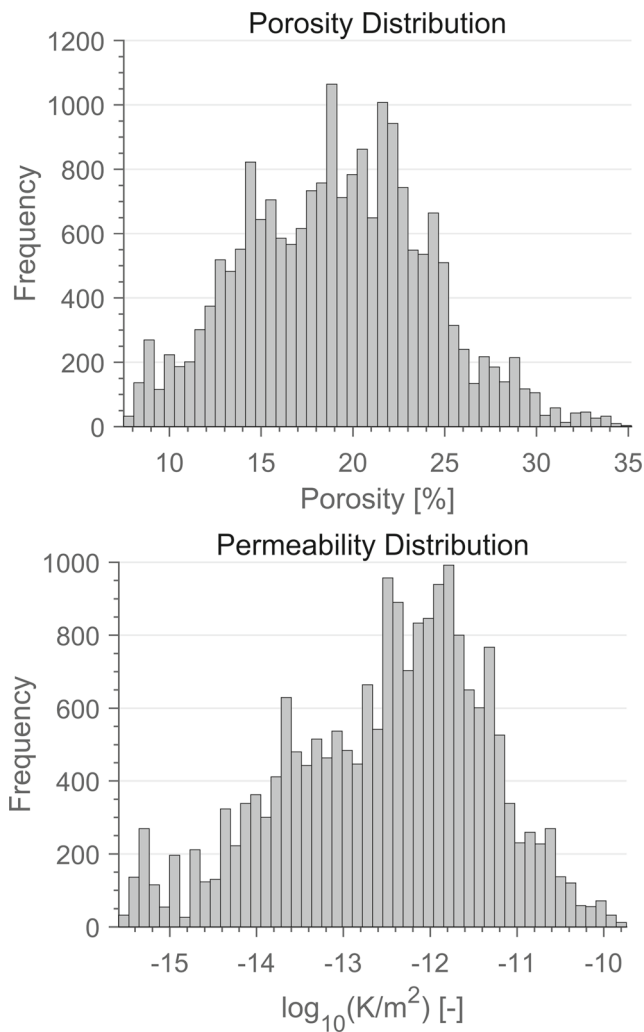


Fig. 5 Porosity and corresponding permeability distribution after [36]

### 4.1 Test case 1: Two-dimensional CO<sub>2</sub> injection

Two-dimensional CO<sub>2</sub> injection into a heterogeneous porous medium is examined. The domain of interest extends 600 m in *x*-direction and 100 m in *z*-direction with a thickness of 1 m in *y*-direction. The discretization consists of 2.5 m × 2.5 m blocks, yielding 240 cells in *x*-direction and 40 cells in *z*-direction, comprising 9600 cells in total. Thus, this problem has 19,200 degrees of freedom.

**Initial and boundary conditions** Initially, there is no CO<sub>2</sub>, i.e.,  $S_n = 0$ . Water pressure  $p_w$  is hydrostatic with the top of the domain situated at a depth of 800 m. This results in pressures at which CO<sub>2</sub> is in a supercritical state with liquid-like densities. Top and bottom boundaries have no-flow conditions, assuming the presence of an impermeable caprock above and below the reservoir. The left boundary is also impermeable and CO<sub>2</sub> is injected through the lower 5 cells, i.e., 12.5 m, with a total injection rate of 0.2 kg s<sup>-1</sup>. The right boundary is open and water pressure and gas saturation are held constant at their initial values. Table 3 shows rock properties, as well as the constant fluid properties for the incompressible simulations. The advection-dominated case uses an entry pressure of  $p_d = 0$  Pa and the diffusion-dominated case of  $p_d = 10^6$  Pa.

**Heterogeneous porosity and permeability distribution** To model heterogeneity, we sample the porosity field from the porosity distribution shown in Fig. 5. The corresponding permeability distribution is calculated after [36] with a fractal model valid for a Rotliegend sandstone typical of the northeastern German basin:

$$K = 155 \phi + 37\,315 \phi^2 + 630 (10 \phi)^{10}. \tag{26}$$

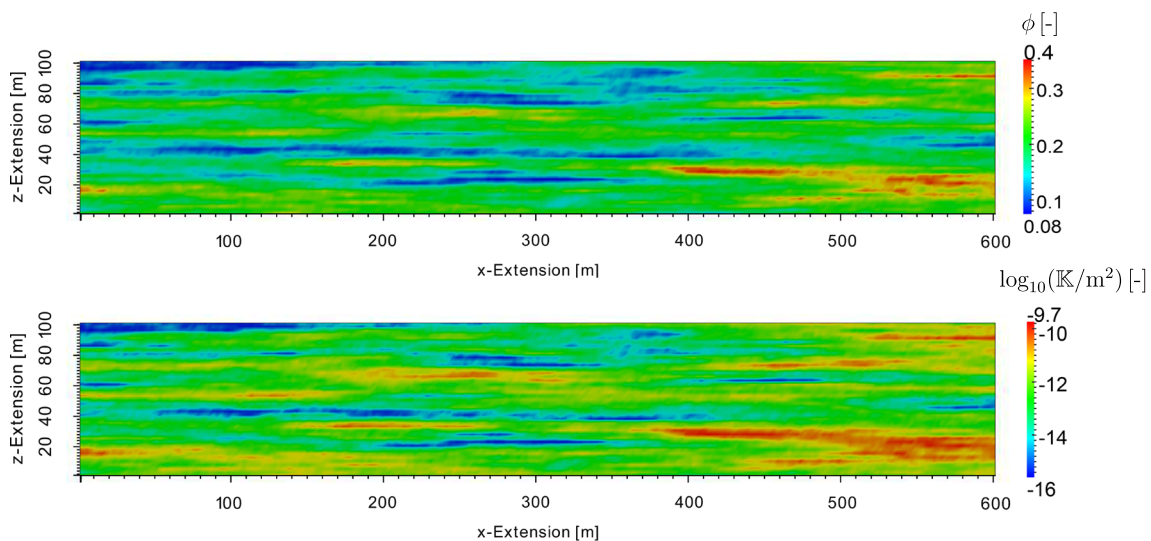


Fig. 6 Porosity field (top) and corresponding permeability field (bottom) after Eq. 26

**Fig. 7** Saturation distribution  $S_n$  after 67 days of injection with  $0.2 \text{ kg s}^{-1}$

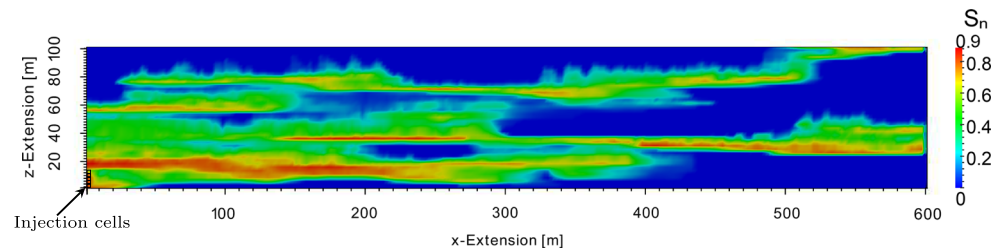


Figure 6 shows the porosity field and the corresponding permeability field calculated after Eq. 26. Figure 7 shows injection into the heterogeneous porous medium. The permeability distribution clearly influences the shape of the  $\text{CO}_2$  plume.

Tables 1 and 2 show the number of time steps (TS), number of Newton iterations (NI), and linear iterations (LI), as well as the total time to solution for both test cases, namely the advection- and diffusion-dominated ones. Both CPR-AMG and SCR-AMG profit from the exact Jacobian with fewer Newton iterations and fewer linear iterations. The time to solution is also better using AD compared with FD. For the advection-dominated case, the speedup is 1.17 for SCR-AMG and 1.22 for CPR-AMG when comparing AD and FD. Similarly, for the diffusion-dominated case, the corresponding speedups are 1.56 for SCR-AMG and 1.51 for CPR-AMG.

Comparing the advection-dominated case from Table 1 with the diffusion-dominated case from Table 2, CPR-AMG is faster in the advection case (speedup of 2.35 (AD) and 2.26 (FD)) whereas SCR-AMG is faster in the diffusion case (speedup of 1.51 (AD) and 1.45 (FD)).

## 4.2 Test case 2: SPE10B problem

The 10<sup>th</sup> SPE comparative solution project [44] comprises a three-dimensional domain of dimensions  $365.76 \text{ m} \times 670.56 \text{ m} \times 51.816 \text{ m}$ . This domain is discretized into  $60 \times 220 \times 85$  cells, with one cell having a size of  $6.096 \text{ m} \times 3.048 \text{ m} \times 0.6096 \text{ m}$ . This gives 1,122,000 cells yielding 2,244,000 unknowns. Figure 8 shows the permeability distribution. We inject  $\text{CO}_2$  at a rate of  $6.62 \text{ kg s}^{-1}$  at the

center of the domain along the entire  $z$ -range, i.e., into 85 cells. The initial  $\text{CO}_2$  saturation is zero and water pressure is hydrostatic with the top of the domain lying at a depth of 3.6 km. The upper and lower boundaries are closed while the other boundaries are open. The simulation time is 2000 days. Figure 9 shows  $\text{CO}_2$  saturation distribution after 1060 days of injection.

This test case is particularly challenging since it covers a permeability range of more than ten orders of magnitude. Thus, only methods using AD are able to finish the simulation in the allowed compute time of 72 h. Within this period, the methods using the FD Jacobian only manage to simulate 25.3% (SCR-AMG) and 44.9% (CPR-AMG) of the total simulation time. CPR-AMG performs better than SCR-AMG due to zero capillary pressure used in this test example. The speedup for AD is 5.81 comparing CPR-AMG and SCR-AMG. All results are summarized in Table 4.

## 4.3 Test case 3: $\text{CO}_2$ injection into the Sleipner reservoir

This test case simulates injection of  $\text{CO}_2$  into the Sleipner gas field [43] operated by Statoil and situated in the Norwegian part of the North Sea. The extracted gas contains high amounts of  $\text{CO}_2$ . This  $\text{CO}_2$  is not vented into the atmosphere, but rather compressed and reinjected into a permeable sandstone of the Utsira formation, approximately 800 m below the seabed. We use a discretization of  $65 \times 119 \times 50$  cells, totaling 386,750 cells with cell sizes of  $50 \text{ m} \times 50 \text{ m} \times 1 \text{ m}$ . This yields a domain size of  $3250 \text{ m} \times 5950 \text{ m} \times 50 \text{ m}$ . Assuming an annual injection of approximately 0.9 Mt, we prescribe an injection rate

**Table 1** Performance of different solvers for advection-dominated case where  $TS$  is number of time steps,  $NI$  number of nonlinear iterations,  $LI$  number of linear iterations, and total simulation time in seconds

Solver	TS	NI	LI	LI/NI	NI/TS	Time (s)
SCR-AMG (FD)	27	539	3992	7.40	19.96	208.84
SCR-AMG (AD)	27	509	3817	7.49	18.85	177.82
CPR-AMG (FD)	27	539	7146	13.26	19.96	92.31
CPR-AMG (AD)	27	509	6775	13.31	18.85	75.67

**Table 2** Performance of different solvers for diffusion-dominated case where  $TS$  is number of time steps,  $NI$  number of nonlinear iterations,  $LI$  number of linear iterations, and total simulation time in seconds

Solver	TS	NI	LI	LI/NI	NI/TS	Time (s)
SCR-AMG (FD)	27	807	3217	3.99	29.88	119.14
SCR-AMG (AD)	26	731	2810	3.84	28.11	76.33
CPR-AMG (FD)	29	858	13265	15.46	29.58	173.03
CPR-AMG (AD)	26	733	11364	15.50	28.19	114.91



**Table 3** Model parameters for test case 1 as well as fluid and rock properties

Properties	Parameter	Value	Parameter	Value
Model	Mesh size	$240 \times 1 \times 40$	Cell size	2.5 m
	Total dimensions	$600 \text{ m} \times 1 \text{ m} \times 100 \text{ m}$	Injection rate (over 12.5 m)	$0.2 \text{ kg s}^{-1}$
	Simulation time	67 days	Time step	variable
Fluid/rock	CO <sub>2</sub> density $\rho_n$	$450.0 \text{ kg m}^{-3}$	CO <sub>2</sub> viscosity $\mu_n$	$3.0 \times 10^{-5} \text{ Pa s}$
	Water density $\rho_w$	$1000.0 \text{ kg m}^{-3}$	Water viscosity $\mu_w$	$1.0 \times 10^{-3} \text{ Pa s}$
	Porosity $\phi$	0.2	Permeability $\mathbb{K}$	Heterogeneous
	Pore size distribution parameter $\lambda$	2	Entry pressure $p_d$	$0 \text{ Pa}/10^6 \text{ Pa}$

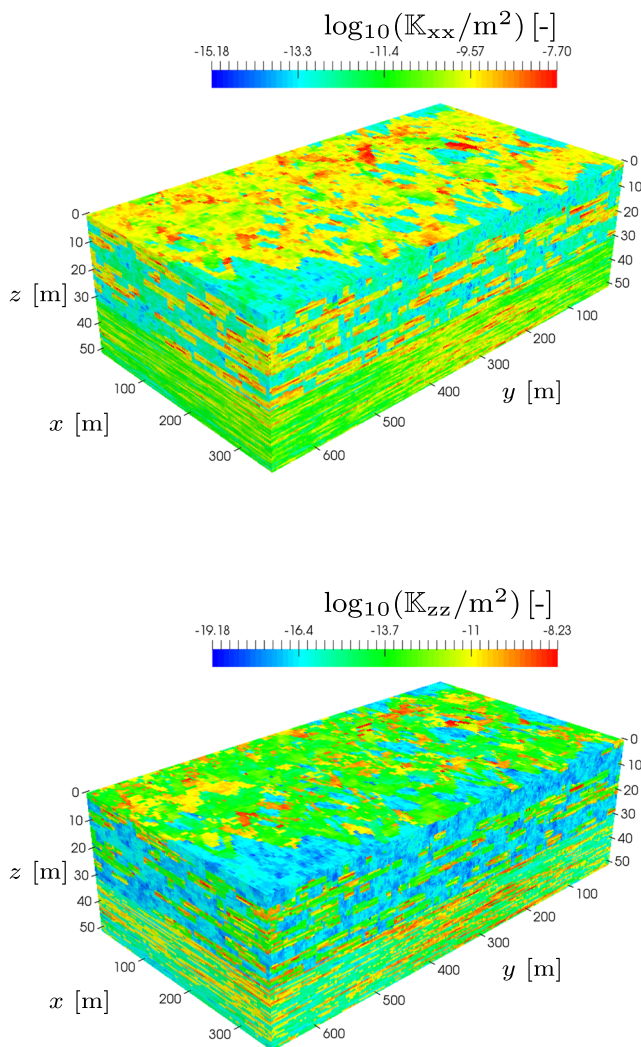
of  $28 \text{ kg s}^{-1}$  and simulate for a period of 30 years. Figure 10 shows the permeability distribution, as well as the topology of the sandstone layer. Figure 11 shows saturation distribution in layer 43 after injecting  $28 \text{ kg s}^{-1}$  of CO<sub>2</sub> for 17 years.

Table 5 shows the comparison of SCR-AMG and CPR-AMG using AD and FD for the Sleipner case. The speedups are 1.97 for SCR-AMG and 3.12 for CPR-AMG, comparing AD and FD. This test case favors SCR-AMG over CPR-AMG with a speedup of 6.58 (AD) and 10.4 (FD).

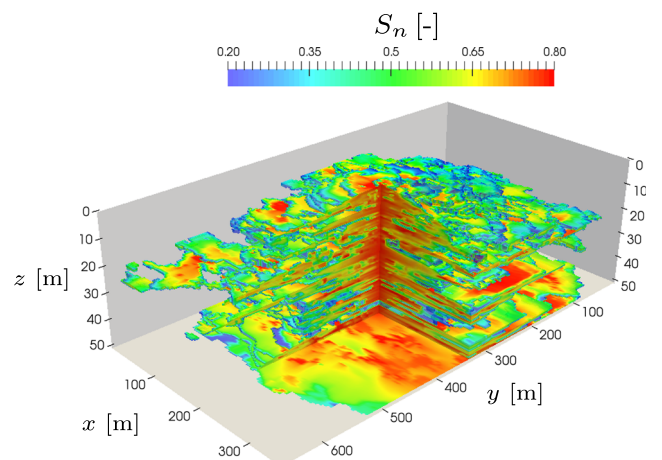
### 5 Weak scaling

We compare the Schur Complement Reduction method (SCR-AMG) and the Constrained Pressure Residual method (CPR-AMG) on JUQUEEN, a BlueGene/Q supercomputer from IBM, with 28,672 nodes located in Jülich, Germany. Each node consists of an IBM PowerPC A2 running at 1.6 GHz with 16 cores and 16 GB of memory. We start with 16 cores for 393,216 cells and finish with 1024 cores for 25,165,824 cells. We perform five timesteps each executing the same number of Newton iterations. We omit I/O completely for this test case, so neither reading the input file nor the output of computed quantities influences the overall computation time.

Figure 12 shows efficiency over the number of cells. Here, efficiency is defined as  $E = S/N$  with speedup  $S = T_1/T_N$ ,  $N$  the number of cores,  $T_1$  time to solution



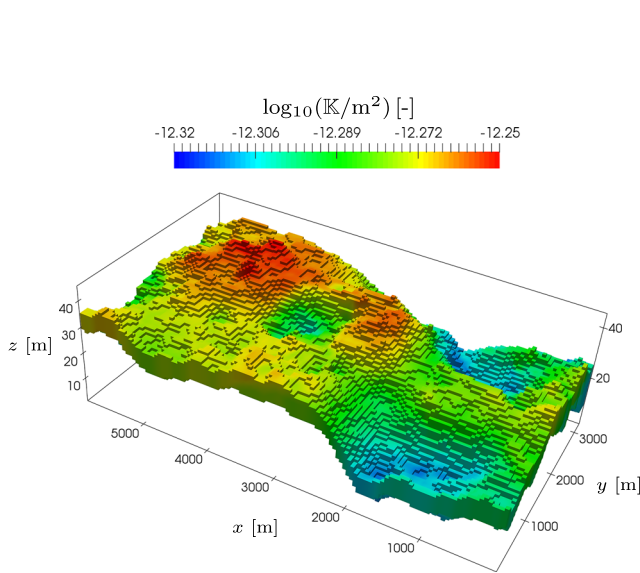
**Fig. 8** Horizontal (top) and vertical (bottom) permeability distribution of SPE10B (bottom view)



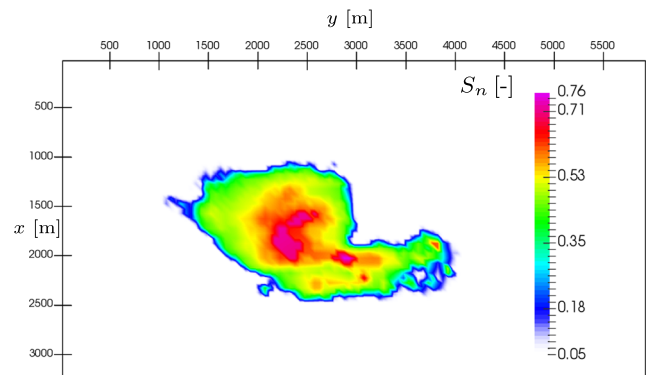
**Fig. 9** CO<sub>2</sub> saturation distribution after 1060 days of injection with  $6.62 \text{ kg s}^{-1}$

**Table 4** Performance of different solvers for SPE10B test case with *TS* number of time steps, *NI* number of nonlinear iterations, *LI* number of linear iterations, and total simulation time in hours

Solver	TS	NI	LI	LI/NI	NI/TS	Time (h)
SCR-AMG (FD)	987	42,488	199,269	4.69	43.04	> 72 (25.3% of $t_{end}$ )
SCR-AMG (AD)	392	2071	44,982	21.71	5.28	42.71
CPR-AMG (FD)	1279	51,252	561,667	10.95	40.07	> 72 (44.9% of $t_{end}$ )
CPR-AMG (AD)	452	1975	78,842	39.92	4.36	7.35



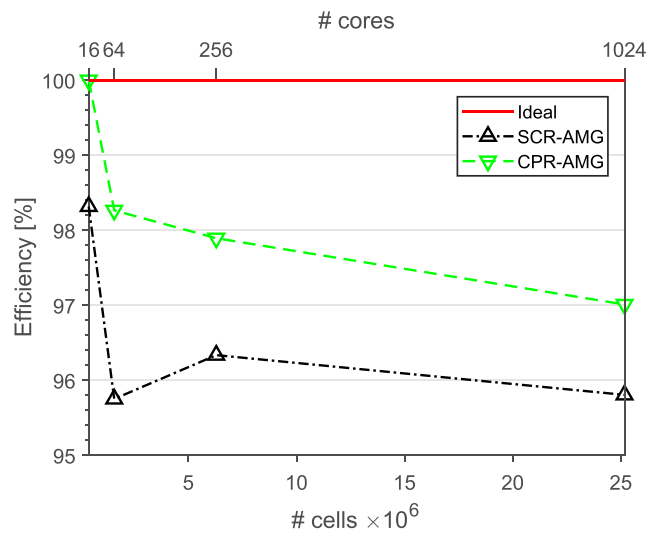
**Fig. 10** Permeability distribution of the Sleipner reservoir



**Fig. 11** Saturation distribution after injecting  $28 \text{ kg s}^{-1}$  in layer 43 for 17 years

**Table 5** Performance of different solvers for Sleipner test case with *TS* number of time steps, *NI* number of nonlinear iterations, *LI* number of linear iterations, and total simulation time in hours

Solver	TS	NI	LI	LI/NI	NI/TS	Time (h)
SCR-AMG (FD)	442	4075	29,954	7.35	9.21	5.33
SCR-AMG (AD)	442	2366	19,541	8.25	5.35	2.70
CPR-AMG (FD)	3760	24,305	1,256,671	51.70	6.46	55.41
CPR-AMG (AD)	3509	6492	491,901	75.77	1.85	17.76



**Fig. 12** Efficiency of SCR-AMG and CPR-AMG over the number of cells and number of cores compared with ideal efficiency. Note, efficiency starts at 95%

with one core, and  $T_N$  time to solution with  $N$  cores. SCR-AMG as well as CPR-AMG both have efficiencies above 95%, demonstrating nearly ideal scaling properties.

## 6 Summary and conclusion

The speedup when using AD instead of FD ranges from 1.17 (SCR-AMG) and 1.22 (CPR-AMG) in the advection-dominated two-dimensional case up to 1.51 (CPR-AMG) and 1.56 (SCR-AMG) in the diffusion-dominated two-dimensional case to speedups of 1.97 (SCR-AMG) and 3.12 (CPR-AMG) for the Sleipner reservoir. These differences in speedup are due to the different time percentages for the computation of the Jacobian and time spent in the linear solver.

The SPE10B problem turns out to be the most challenging problem. In this case, only the AD code finishes in the desired simulation time of 72 h. CPR-AMG is in particular suitable for advection-dominated cases (speedup of 5.81 in the SPE10B example) compared with SCR-AMG, whereas SCR-AMG is faster in the diffusion-dominated cases (speedup of 10.40 (FD) and 6.58 (AD) in the Sleipner example).

AD always reduces the overall runtime and reduces the number of Newton iterations and linear iterations. The speedup in the Sleipner case is more than threefold when using AD. Moreover, the SPE10B case could only be solved in time with AD. In addition, AD circumvents the definition of an FD step size and the associated approximation errors and avoids error-prone, hand-coded Jacobians. Consequently, we would always advise to use AD.

AD together with algebraic multigrid (AMG) makes these solution methods for the fully coupled, fully implicit two-phase flow equations highly competitive. Due to its physics-based splitting approach, the pressure and saturation field are addressed in an optimal way.

The two presented solvers are both efficient and scalable with efficiencies above 95% on JUQUEEN. While SCR-AMG is more suited for diffusion-dominated cases, CPR-AMG deals with advection-dominated cases better. Through PETSc, both solvers can be easily modified and enhanced. For example, the Schur complement solution step for SCR-AMG by Hypre/BoomerAMG could be replaced by a block ILU method, accounting for an advection-dominated  $J_{22}$  block. In contrast, for CPR-AMG, the additive fieldsplit could benefit from an additional solution step on the saturation field for a  $J_{22}$  block with a strong capillary pressure derivative.

**Funding** Open Access funding provided by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix 1: PETSc command line options for the different solvers

This appendix shows the different command line options used to invoke the linear solvers in PETSc.

The following options for the weak scaling test are shared by all the solvers.

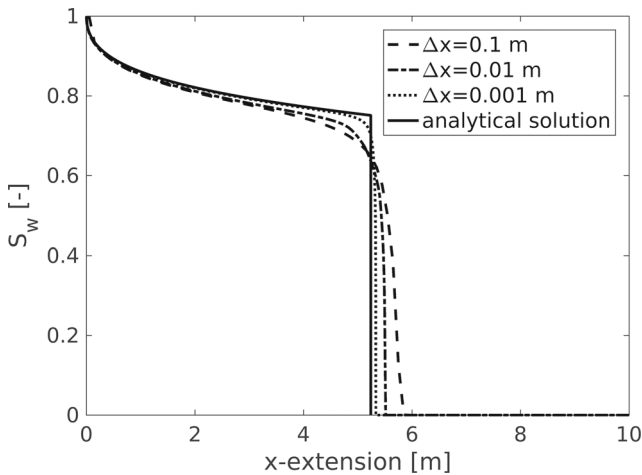
```
-ksp_atol 1e-50 -ksp_rtol 1e-6 -ksp_max_it 100
-ksp_type fgmres
```

### SCR-AMG

```
-pc_type fieldsplit
-pc_fieldsplit_type schur
-pc_fieldsplit_schur_precondition all
-fieldsplit_0_ksp_type preonly
-fieldsplit_0_pc_type hypre
-fieldsplit_0_pc_hypre_type boomeramg
-fieldsplit_1_ksp_type gmres
-fieldsplit_1_pc_type hypre
-fieldsplit_1_pc_hypre_type boomeramg
-fieldsplit_1_ksp_max_it 10
-fieldsplit_1_ksp_rtol 1e-2
```

### CPR-AMG

```
-pc_type composite
-pc_composite_type multiplicative
-pc_composite_pcs fieldsplit,bjacobi
-sub_0_ksp_type fgmres
-sub_0_pc_fieldsplit_type additive
-sub_0_fieldsplit_0_ksp_type gmres
-sub_0_fieldsplit_0_pc_type hypre
-sub_0_fieldsplit_0_pc_hypre_type boomeramg
-sub_0_fieldsplit_1_ksp_type preonly
-sub_0_fieldsplit_1_pc_type none
-sub_1_sub_pc_type ilu
```



**Fig. 13** Comparison of analytical Buckley-Leverett solution to numerical results showing grid convergence

### Appendix 2: Code verification using the Buckley-Leverett problem

The Buckley-Leverett problem describes the immiscible displacement of oil by water in a porous medium (cf. [9]). It is widely used for the verification of numerical models. Figure 13 shows the analytical solution for the problem and compares it with numerical solutions with different cell sizes. It is clear that the numerical solution approaches the analytical for increasingly smaller cell sizes.

### Appendix 3: Discussion of gravity numbers

The gravitational number is defined as:

$$Gr = \frac{(\rho_w - \rho_n) g \mathbb{K}}{\mu_n v_{cr}} = \frac{\text{gravitational forces}}{\text{viscous forces}} \quad (27)$$

and relates gravitational and viscous forces. Here,  $v_{cr} = \frac{\phi l_{cr}}{t_{cr}}$  is the characteristic velocity with a characteristic length  $l_{cr}$  and a characteristic time  $t_{cr}$ . Permeability is a scalar, which means that it is assumed to be isotropic.

For our test case 1, we have a reservoir thickness of 100 m, thus  $l_{cr} = 100$  m. Simulation time is 67 days and consequently  $t_{cr} = 5,788,800$  s. Densities are chosen as  $\rho_n = 450 \text{ kg m}^{-3}$  and  $\rho_w = 1000 \text{ kg m}^{-3}$ . Non-wetting viscosity is  $\mu_n = 3.0 \times 10^{-5} \text{ Pa s}$  and permeability is  $\mathbb{K} = 10^{-12} \text{ m}^2$ . All in all, we have:

$$Gr = \frac{(1000 - 450) \cdot 9.81 \cdot 10^{-12}}{3.0 \cdot 10^{-5} \cdot \left(\frac{0.2 \cdot 100}{67 \cdot 86400}\right)} \approx 52.06. \quad (28)$$

In addition, we consider two test cases: (i) injection into a hot aquifer with  $\rho_n = 250 \text{ kg m}^{-3}$  and  $\mu_n = 3.0 \times 10^{-5} \text{ Pa s}$  and (ii) injection into a deep aquifer with  $\rho_n = 650 \text{ kg m}^{-3}$

**Table 6** Performance of different solvers for advection-dominated cases and hot aquifer (high gravitational number) where *TS* is number of time steps, *NI* number of nonlinear iterations, and *LI* number of linear iterations

Solver	TS	NI	LI	LI/NI	NI/TS
SCR-AMG (FD)	26	566	4795	8.47	21.76
SCR-AMG (AD)	26	538	4615	8.57	20.69
CPR-AMG (FD)	26	566	8504	15.02	21.76
CPR-AMG (AD)	26	538	7773	14.44	20.69

and  $\mu_n = 8.0 \times 10^{-5} \text{ Pa s}$ . This leads to gravitational numbers of (i)  $Gr \approx 70.99$  and (ii)  $Gr \approx 12.42$ .

Comparing Table 1 with Table 6, we have a higher number of Newton and linear iterations for the hot aquifer. This is in line with the higher gravitational number of  $Gr \approx 70.99$  compared with  $Gr \approx 52.06$  for the original test case.

Table 7 shows results for the deep aquifer with a low gravitational number of  $Gr \approx 12.42$ . Here, we have a lower number of Newton and linear iterations. Again, this is in line with the lower gravitational number. Analogous results hold for the diffusion-dominated case.

Gravity numbers for the Sleipner and SPE10B case are  $Gr \approx 88.96$  and  $Gr \approx 555.20$ , respectively. To calculate a single gravitational number, we used mean permeabilities and porosities. The high gravitational number of the SPE10B case reflects its high degree of difficulty.

### Appendix 4: Discussion of CFL numbers

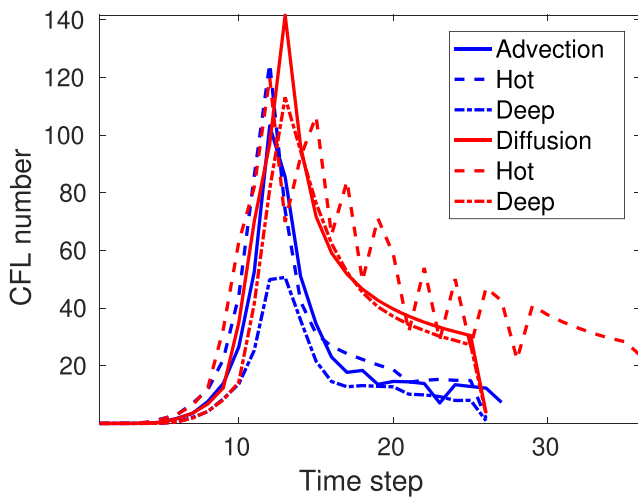
The Courant–Friedrichs–Lewy (CFL) condition can be derived using the saturation equation from the fractional flow formulation:

$$\frac{\partial S_w}{\partial t} + \frac{\mathbf{v}_{tot}}{\phi} \frac{df_w}{dS_w} \nabla S_w = q_w. \quad (29)$$

Here,  $\mathbf{v}_{tot} = \mathbf{v}_w + \mathbf{v}_n$  is the total velocity and  $f_w = \frac{\lambda_w}{\lambda_w + \lambda_n}$  is the fractional flow function. Then, the CFL number *C* is

**Table 7** Performance of different solvers for advection-dominated cases and deep aquifer (low gravitational number) where *TS* is number of time steps, *NI* number of nonlinear iterations, and *LI* number of linear iterations

Solver	TS	NI	LI	LI/NI	NI/TS
SCR-AMG (FD)	26	231	1845	7.98	8.88
SCR-AMG (AD)	26	231	1839	7.96	8.88
CPR-AMG (FD)	26	230	2525	10.97	8.84
CPR-AMG (AD)	26	230	2515	10.93	8.84



**Fig. 14** CFL numbers for advection- and diffusion-dominated two-dimensional test case for original, hot, and deep aquifer (finite difference case of SCR-AMG)

defined as:

$$C := \frac{\Delta t}{\phi} \left( \frac{|\mathbf{v}_{tot,x}|}{\Delta x} + \frac{|\mathbf{v}_{tot,y}|}{\Delta y} + \frac{|\mathbf{v}_{tot,z}|}{\Delta z} \right) \frac{df_w}{dS_w}. \quad (30)$$

Figure 14 shows the CFL number for every time step for the advection- and diffusion-dominated two-dimensional test case for the original, hot, and deep aquifer. The figure shows results from the finite difference approximation of the Jacobian and SCR-AMG.

For the advection-dominated case, we have CFL numbers of over 100 in the 12th time step. For the hot aquifer, CFL numbers reach values of approximately 125 also in the 12th time step and for the deep aquifer values of over 50 in the 13th time step. Similarly, for the diffusion-dominated case, we have CFL numbers of over 140 in the 13th time step. For the hot aquifer, CFL numbers reach a value of over 105 in the 15th time step and for the deep reservoir over 110 in the 13th time step. The high CFL numbers indicate that the fully implicit fully coupled approach is indeed well justified.

**Table 8** Performance of different solvers for diffusion-dominated cases and anisotropy factors of 1:100 where *TS* is number of time steps, *NI* number of nonlinear iterations, and *LI* number of linear iterations

Solver	TS	NI	LI	LI/NI	NI/TS
SCR (FD)	38	1490	6708	4.50	39.21
SCR (AD)	39	1568	7298	4.65	40.20
CPR (FD)	122	1092	33,192	30.39	8.95
CPR (AD)	122	1102	33,212	30.13	9.03

**Table 9** Performance of different solvers for diffusion-dominated cases and anisotropy factors of 1:10,000 where *TS* is number of time steps, *NI* number of nonlinear iterations, and *LI* number of linear iterations

Solver	TS	NI	LI	LI/NI	NI/TS
SCR (FD)	46	1593	9658	6.06	34.53
SCR (AD)	46	1715	8566	4.99	37.28
CPR (FD)	925	2612	163,633	62.64	2.82
CPR (AD)	889	2678	160,475	59.92	3.01

### Appendix 5: Influence of anisotropy

In this section, the influence of anisotropy on solver performance is examined. The diffusion-dominated two-dimensional test case from Section 4 is the basis for the comparison. Results for anisotropy factors of 1:100 and 1:10,000 are shown in Tables 8 and 9, respectively.

Comparing Table 8 with Table 2, performance of CPR-AMG decreases. An increase in the number of time steps (factor of 4.21 for FD and 4.69 for AD), Newton iterations, and linear iterations is observed. The performance of SCR-AMG is affected less by anisotropy with a slight increase in the number of time steps (factor of 1.41 for FD and 1.50 for AD), Newton iterations, and linear iterations.

Looking at Table 9, performance of CPR-AMG drastically decreases, both for the FD and the AD case. The number of time steps increases by a factor of 31.90 (FD) and 34.19 (AD) compared with the original values with no anisotropy. The performance of SCR-AMG is significantly more stable. The number of time steps increases only by a factor of 1.59 (FD) and 1.77 (AD) compared with the values with no anisotropy.

In summary, CPR-AMG is affected much more by anisotropy compared with SCR-AMG. Interestingly, this is also true for the advection-dominated case, where CPR-AMG originally outperformed SCR-AMG.

### References

1. Aziz, K., Settari, A.: Petroleum Reservoir Simulation. Applied Science Publishers, London (1979)
2. Balay, S., Gropp, W.D., McInnes, L.C., Smith, B.F.: Modern Software Tools in Scientific Computing, Birkhäuser Arge, E., Bruaset, A.M., Langtangen, H.P. (eds.) MA, Boston (1997)
3. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.D., Karpeyev, D., Kaushik, D., Knepley, M.G., May, D.A., McInnes, L.C., Mills, R.T., Munson, T., Rupp, K., Sanan, P., Smith, B.F., Zampini, S., Zhang, H., Zhang, H.: PETSc users manual Tech. Rep. ANL-95/11 - Revision 3.13, Argonne National Laboratory, Lemont, IL, USA, <https://www.mcs.anl.gov/petsc> (2020)
4. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.D.,

- Karpeyev, D., Kaushik, D., Knepley, M.G., May, D.A., McInnes, L.C., Mills, R.T., Munson, T., Rupp, K., Sanan, P., Smith, B.F., Zampini, S., Zhang, H., Zhang, H.: PETSc Web page <http://www.mcs.anl.gov/petsc> (2020)
5. Bastian, P.: Numerical Computation of Multiphase Flows in Porous Media. Habilitation thesis, Technische Fakultät, Christian-Albrechts-Universität Kiel [https://conan.iwr.uni-heidelberg.de/data/people/peter/pdf/Bastian\\_habilitationthesis.pdf](https://conan.iwr.uni-heidelberg.de/data/people/peter/pdf/Bastian_habilitationthesis.pdf) (1999)
  6. Bastian, P., Helmig, R.: Efficient fully-coupled solution techniques for two-phase flow in porous media: Parallel multigrid solution and large scale computations. *Adv. Water Resour.* **23**(3), 199–216 (1999). [https://doi.org/10.1016/S0309-1708\(99\)00014-7](https://doi.org/10.1016/S0309-1708(99)00014-7)
  7. Bear, J.: *Hydraulics of Groundwater* McGraw-Hill New York (1979)
  8. Brooks, R.J., Corey, A.T.: *Hydraulic Properties of Porous Media*, vol 3 Colorado State University Hydrology Paper Fort Collins CO (1964)
  9. Buckley, S.E., Leverett, M.C.: Mechanism of fluid displacement in sands. *Transactions of the AIME* **146**(01), 107–116 (1942)
  10. Bui, Q.M., Elman, H.C., Moulton, J.D.: Algebraic Multigrid Preconditioners for Multiphase Flow in Porous Media. *SIAM Journal on Scientific Computing* **39**(5), S662–S680 (2017). <https://doi.org/10.1137/16M1082652>
  11. Burdine, N.T.: Relative Permeability Calculations from Pore-Size Distribution Data. *Petroleum Transactions AIME* **198**, 71–77 (1953). <https://doi.org/10.2118/225-G>
  12. Büsing, H.: Efficient Solution Techniques for Multi-phase Flow in Porous Media. In: Lirkov I., Margenov, S., Large-Scale Scientific Computing, Springer International Publishing, Cham, Switzerland, pp. 572–579 (2018). [https://doi.org/10.1007/978-3-319-73441-5\\_63](https://doi.org/10.1007/978-3-319-73441-5_63)
  13. Chavent, G., Jaffrè, J.: *Mathematical Models and Finite Elements for Reservoir Simulation North-Holland Amsterdam* (1986)
  14. Clauser, C.: *Numerical Simulation of Reactive Flow in Hot Aquifers: SHEMAT and processing SHEMAT* Springer Heidelberg-Berlin (2003)
  15. Darcy, H.: *Les Fontaines Publiques de la Ville de Dijon* Dalmont Paris (1856)
  16. Dawson, C.N., Klíe, H., Wheeler, M.F., Woodward, C.S.: A parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton–Krylov solver. *Computational Geosciences* **1**(3–4), 215–249 (1997). <https://doi.org/10.1023/A:1011521413158>
  17. Douglas, J. Jr., Peaceman, D., Rachford, H. Jr.: A Method for Calculating Multi-Dimensional Immiscible Displacement. *Transactions of the Metallurgical Society of AIME* **216**, 297–308 (1959)
  18. Falgout, R.D., Yang, U.M.: hypre: A Library of High Performance Preconditioners Sloop PMA Hoekstra AG Tan CJK Dongarra JJ eds *Computational Science — ICCS 2002* Springer Heidelberg-Berlin (2002)
  19. Falgout, R.D., Jones, J.E., Yang, U.M., Bruaset, A.M., Tveito, A. (eds.): *The Design and Implementation of hypre, a Library of Parallel High Performance Preconditioners*. Springer, Heidelberg-Berlin (2006). [https://doi.org/10.1007/3-540-31619-1\\_8](https://doi.org/10.1007/3-540-31619-1_8)
  20. van Genuchten, M.T.: A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Science Society of America Journal* **44**, 892–898 (1980). <https://doi.org/10.2136/sssaj1980.03615995004400050002x>
  21. Gries, S.: On the Convergence of System-AMG in Reservoir Simulation. *SPE J.* **23**(02), 589–597 (2018). <https://doi.org/10.2118/182630-PA>
  22. Gries, S., Stüben, K., Brown, G.L., Chen, D., Collins, D.A., et al.: Preconditioning for Efficiently Applying Algebraic Multigrid in Fully Implicit Reservoir Simulations. *SPE J.* **19**(04), 726–736 (2014). <https://doi.org/10.2118/163608-PA>
  23. Griewank, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation Society for Industrial and Applied Mathematics SIAM Philadelphia PA* (2000)
  24. Hackbusch, W.: *Multi-Grid Methods and Applications* Springer Heidelberg-Berlin (1985)
  25. Hascoet, L., Pascual, V.: The Tapenade Automatic Differentiation tool: principles, model, and specification. *ACM Transactions on Mathematical Software TOMS* **39**(3), 20 (2013). <https://doi.org/10.1145/2450153.2450158>
  26. Henson, V., Yang, U.: BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.* **41**(1), 155–177 (2002). [https://doi.org/10.1016/S0168-9274\(01\)00115-5](https://doi.org/10.1016/S0168-9274(01)00115-5)
  27. Jenny, P., Lee, S., Tchelepi, H.: Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media. *J. Comput. Phys.* **217**(2), 627–641 (2006). <https://doi.org/10.1016/j.jcp.2006.01.028>
  28. Kayum, S., Cancelliere, M., Rogowski, M., Al-Zawawi, A.: Application of Algebraic Multigrid in Fully Implicit Massive Reservoir Simulations. In: *SPE Europec featured at 81st EAGE Conference and Exhibition*, London, England, UK, Society of Petroleum Engineers Richardson, TX, USA (2019). <https://doi.org/10.2118/195472-MS>
  29. Klíe, H.M., Ramè, M., Wheeler, M.: Two-stage Preconditioners for inexact Newton Methods in multi-phase reservoir simulation. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.7551&rep=rep1&type=pdf> (1996)
  30. Klíe, H.M., Monteagudo, J., Hoteit, H., Rodriguez, A.A.: Towards a New Generation of Physics Driven Solvers for Black Oil and Compositional Flow Simulation. In: *SPE Reservoir Simulation Symposium*, The Woodlands, TX, USA, Society of Petroleum Engineers, Richardson, TX, USA (2009). <https://doi.org/10.2118/118752-MS>
  31. Lacroix, S., Vassilevski, Y.V., Wheeler, M.F.: Decoupling preconditioners in the implicit parallel accurate reservoir simulator IPARS. *Numerical Linear Algebra with Applications* **8**(8), 537–549 (2001). <https://doi.org/10.1002/nla.264>
  32. Lacroix, S., Vassilevski, Y., Wheeler, J., Wheeler, M.: Iterative Solution Methods for Modeling Multiphase Flow in Porous Media Fully Implicitly. *SIAM J. Sci. Comput.* **25**(3), 905–926 (2003). <https://doi.org/10.1137/S106482750240443X>
  33. Mary, T.: Block low-rank multifrontal solvers : complexity, performance, and scalability. PhD thesis, Université Paul Sabatier - Toulouse III <https://tel.archives-ouvertes.fr/tel-01929478> (2017)
  34. Mishev, I.D., Shaw, J.S., Lu, P.: Numerical Experiments with AMG Solver in Reservoir Simulation. In: *SPE Reservoir Simulation Symposium*, The Woodlands, TX, USA, Society of Petroleum Engineers, Richardson, TX, USA (2011). <https://doi.org/10.2118/141765-MS>
  35. Mualem, Y.: A New Model for Predicting the Hydraulic Conductivity of Unsaturated Porous Media. *Water Resour. Res.* **12**, 513–522 (1976). <https://doi.org/10.1029/WR012i003p00513>
  36. Pape, H., Clauser, C., Iffland, J.: Permeability prediction based on fractal pore-space geometry. *Geophysics* **64**(5), 1447–1460 (1999). <https://doi.org/10.1190/1.1444649>
  37. Rall, L.B.: *Automatic Differentiation: Techniques and Applications* Springer New York (1981)
  38. Rath, V., Wolf, A., Bucker, M.: Joint three-dimensional inversion of coupled groundwater flow and heat transfer based on automatic differentiation: sensitivity calculation, verification and synthetic examples. *Geophys. J. Int.* **167**, 453–466 (2006). <https://doi.org/10.1111/j.1365-246X.2006.03074.x>

39. Saad, Y.: A Flexible Inner-Outer Preconditioned GMRES Algorithm. *SIAM J. Sci. Comput.* **14**(2), 461–469 (1993). <https://doi.org/10.1137/0914028>
40. Saad, Y.: *Iterative Methods for Sparse Linear Systems* 2nd edn Society for Industrial and Applied Mathematics Philadelphia, PA (2003)
41. Saad, Y., Schultz, M.H.: GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing* **7**(3), 856–869 (1986). <https://doi.org/10.1137/0907058>
42. Scheichl, R., Masson, R., Wendebourg, J.: Decoupling and block preconditioning for sedimentary basin simulations. *Computational Geosciences* **7**(4), 295–318 (2003). <https://doi.org/10.1023/B:COMG.0000005244.61636.4e>
43. Singh, V.P., Cavanagh, A., Hansen, H., Nazarian, B., Iding, M., Ringrose, P.S., et al.: Reservoir Modeling of CO<sub>2</sub> Plume Behavior Calibrated Against Monitoring Data From Sleipner, Norway. In: *SPE Annual Technical Conference and Exhibition*, Florence, Italy, Society of Petroleum Engineers Richardson TX USA (2010). <https://doi.org/10.2118/134891-MS>
44. SPE: SPE 10th Comparative Solution Project Description of Model 2: SPE Project Website: <http://www.spe.org/web/csp/datasets/set02.htm> Richardson TX (2001)
45. Stüben, K., Clees, T., Klie, H., Lu, B., Wheeler, M.F.: Algebraic Multigrid Methods (AMG) for the Efficient Solution of Fully Implicit Formulations in Reservoir Simulation. In: *SPE Reservoir Simulation Symposium*, Houston, TX, USA, Society of Petroleum Engineers, Richardson, TX, USA (2007). <https://doi.org/10.2118/105832-MS>
46. Van der Vorst, H.A.: Bi-CGSTab: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* **13**(2), 631–644 (1992). <https://doi.org/10.1137/0913035>
47. Wallis, J. et al.: Incomplete Gaussian Elimination as a Preconditioning for Generalized Conjugate Gradient Acceleration. In: *SPE Reservoir Simulation Symposium*, San Francisco, CA, Society of Petroleum Engineers, Richardson, TX, USA (1983). <https://doi.org/10.2118/12265-MS>
48. Wallis, J.R., Kendall, R., Little, T., et al.: Constrained Residual Acceleration of Conjugate Residual Methods. <https://doi.org/10.2118/13536-MS> (1985)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.