Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 292 (2015) 3–15

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Evaluating linear and nonlinear solvers for density driven flow

Arne Nägel, Andreas Vogel, Gabriel Wittum*

Goethe-Center for Scientific Computing, Goethe University, Kettenhofweg 139, 60325 Frankfurt am Main, Germany

Available online 4 December 2014

Highlights

- We investigate different non-linear solvers for density driven flow.
- Iterative solvers of decoupling and fully coupled type are compared.
- For the fully coupled Newton iteration, we develop a transformation mimicking a local decoupling of the unknowns (pressure/salt mass fraction).
- Based on this transformation, we tailor an algebraic multigrid method for density driven flow.

Abstract

This study investigates properties of different solvers for density driven flow problems. The focus is on both non-linear and linear solvers. For the non-linear part, we compare fully coupled method using a Newton linearization and iteratively coupled versions of Jacobi and Gauss–Seidel type. Fully coupled methods require effective preconditioners for the Jacobian. To that end we present a transformation eliminating some couplings and present a strategy for employing algebraic multigrid to the transformed system as well. The work covers theoretical aspects, and provides numerical experiments. Although the primary focus is on density driven flow, we believe that the analysis may well be extended beyond to similar equations with coupled phenomena, such as geomechanics.

Keywords: Density driven flow; Solvers; Preconditioners; Algebraic multigrid

1. Introduction

In many problems from porous media flow effects induced by a variable density play an important role. If water is flowing through a rock matrix containing salt, the salt is dissolved, and turns fresh water turn into brine. Since this newly formed brine has a higher density than the water in its environment, it tends to sink down, inducing a so called *density-driven-flow*. Processes belonging to same category are thermal flows, flows of fluids containing gases, like CO_2, or, as a combination of effects, thermohaline flows.

* Corresponding author.
E-mail addresses: naegel@gcsc.uni-frankfurt.de (A. Nägel), andreas.vogel@gcsc.uni-frankfurt.de (A. Vogel), wittum@gcsc.uni-frankfurt.de (G. Wittum).

http://dx.doi.org/10.1016/j.cma.2014.11.041

The demand for fast solvers for this time-dependent, non-linear process is obvious: In each single time step, a non-linear equation must be solved. This is typically achieved by some fixed-point iteration. Since a fully coupled Newton iteration is often regarded being very demanding with respect to both discretization and solvers, often variants of Picard or Newton iterations are preferred.

Early works, e.g., [1], highlighted the benefits of a *partial Newton* method. These approximate the Jacobian and consider only the self couplings for each unknown component. This strategy is also employed, e.g., in [2,3]. These works describe a predictor–corrector with an explicit predictor and an implicit corrector. The scheme is also suitable for thermohaline flow and features a time stepping strategy and error estimates.

A related class of solvers are *iterative coupling* strategies. These provide a natural way to couple different modules and can be considered as variants of operator splitting technique. This class has widely been applied, e.g., to multiphase flow [4,5], or geomechanics [6–9]. Based on a Picard iteration a similar (partially explicit) strategy is pursued in [10,11].

However, *fully coupled Newton* iterations have also been applied successfully to both density driven [12,13] and thermohaline flow [14] based on the d^3f software [15,16].

One key component is (adaptive) multilevel preconditioners for a fully coupled linear systems. Multi-grid methods belong to the class of domain-decomposition solvers that effectively use smoothing properties of a simple relaxation process, such as Gauss–Seidel, on different levels of a (geometric) finite element hierarchy [17,18]. As a result, all frequencies of the error are reduced equally well, resulting in a method that provides optimal (linear) for many elliptic problems. When geometric information about the hierarchy is not accessible to the user, so called algebraic multigrid (AMG) methods provide an interesting alternative [19,20].

For scalar problems emerging in the partial Newton method, the SAMG algorithm has successfully been applied [3]. However, designing an AMG algorithm for a fully coupled Newton is a more involved task. We address this challenge based on the Filtering Algebraic Multigrid (FAMG) algorithm [21]. The key step will be a special transformation eliminating negative entries from the diagonal.

This work is organized as follows: In Section 2 we set the stage by presenting the governing equations. In Section 3, we define various solvers and preconditioners. These are then analyzed in the numerical experiments presented in Section 4. Conclusions are found in Section 5.

2. Preliminaries

2.1. Governing equations

In brine solutions density effects play an important role, which gives rise to the problem of density driven flow. The full equations state conservations of the fluid and the salt mass respectively, and are typically formulated in terms of *pressure* p and *salt mass fraction* ω, e.g., [22–24]:

$$\partial_t(\Phi\rho\omega) + \nabla \cdot [\rho\omega\mathbf{q} - \rho\mathbf{D}\nabla\omega] = \rho\omega q_V \tag{1a}$$

$$\partial_t(\Phi\rho) + \nabla \cdot [\rho\mathbf{q}] = \rho q_V. \tag{1b}$$

We assume that the fluid is moving with the Darcy velocity

$$\mathbf{q} = -\frac{K}{\mu}(\nabla p - \rho\mathbf{g}), \tag{2}$$

diffusion and dispersion is due to a Scheidegger-type tensor

$$\mathbf{D} = \mathbf{D}_{mol} + \mathbf{D}_{mec}(\mathbf{q}). \tag{3}$$

Here, the density $\rho = \rho(\omega)$ and viscosity $\mu = \mu(\omega)$ are given by non-linear material laws. The porosity Φ, permeability K, gravity $\mathbf{g}$, sources q_V are constant or depend on space only.

It is important to note that although (1) is written as a time dependent problem, it is essentially a differential algebraic equation of index 1: For any given salt mass fraction $\omega = \omega(t, x)$ (1b) serves as a constraint providing a solution ∇p that guarantees the conservation of fluid mass. Engaging some algebra, one observes that (1b) may be replaced by

$$\nabla \cdot \mathbf{q} = q_V + \frac{\partial}{\partial \omega} \left(\frac{1}{\rho} \right) \nabla \cdot [-\rho \mathbb{D} \nabla \omega].$$

Thus, only the salt mass fraction ω is transported in (1a).

2.2. *Boussinesq approximation*

A simplified and frequently used version of (1) is the Oberbeck–Boussinesq approximation [25,26]. In this case the dependence of $\rho = \rho(\omega)$ on ω is only considered for the velocity $\mathbf{q}$:

$$\nabla \cdot [\mathbf{q}] = q_V \tag{4a}$$

$$\partial_t (\Phi \omega) + \nabla \cdot [\omega \mathbf{q} - \mathbf{D} \nabla \omega] = \omega q_V. \tag{4b}$$

Although this formulation is known to be less accurate, e.g., [27], we consider it in Section 2.3 from a theoretical point of view.

2.3. *Model problem*

Our goal is to introduce solvers for problems (1) and (4) in the next section. These are based on fixed-point iterations. In an abstract setting, we seek a solution $u = (p, \omega)^T$ for

$$\mathcal{F}_p(p, \omega) = 0, \tag{5}$$

$$\mathcal{F}_\omega(p, \omega) = 0. \tag{6}$$

For the purpose of illustration and motivation, e.g., [28], consider $\mathcal{F}$ given by (4). Linearizing at $u_0 = (p_0, \omega_0)^T$ the Newton method determines a search direction $(\delta p, \delta \omega)^T$ as the solution of

$$\nabla \cdot \left[-\frac{K}{\mu} \nabla \delta p + \mathbf{q}_0' \delta \omega \right] = -\mathcal{F}_{p,0} \tag{7a}$$

$$\partial_t \left(\Phi \delta \omega \right) + \nabla \cdot \left[-\omega_0 \frac{K}{\mu_0} \nabla \delta p + \left(\omega_0 \mathbf{q}_0' + \mathbf{q}_0 \right) \delta \omega - \mathbf{D}_0 \nabla \delta \omega \right] = -\mathcal{F}_{\omega,0}. \tag{7b}$$

Here, all quantities with the subscript 0 are evaluated at the linearization point u_0. In particular

$$\mathbf{q}_0 := -\frac{K}{\mu} (\nabla p_0 - \rho_0 \mathbf{g}), \qquad \mathbf{q}_0' := \frac{K}{\mu} \rho_0' \mathbf{g}, \quad \rho_0' = \rho'(\omega_0)$$

are the Darcy velocity, its derivative w.r.t. ω, and the derivative of ρ in the linearization point respectively. For the sake of simplicity, derivatives of the dispersion tensor $\mathbf{D}_0$ and the viscosity μ_0 have been neglected.

Eq. (7) allows deducing the following facts for this system: First, the problem is elliptic w.r.t. p and parabolic w.r.t. ω. Second, if $\omega_0 =$ const, the variables decouple, since the dependence on δp in (7b) may be eliminated by means of (7a). In this case, one can first solve for $\delta \omega$ and then, in a next step for δp. Note that although this assumption is unrealistic, it may be fulfilled in parts of the computational domain. We will come back to this observation in Section 3.2.2.

2.4. *Discrete formulation*

The aforementioned system is discretized in space and time. For times t_n let $\mathbf{u}_h^{(n)} = (\mathbf{p}_h^{(n)}, \boldsymbol{\omega}_h^{(n)})^T$ denote the vector with coefficients w.r.t. the space discretization. Given $\mathbf{u}_h^{(n)}$ assume that we perform the step $t_n \to t_{n+1} := t_n + \tau$ using an implicit Euler method. This yields a non-linear equation for $\mathbf{u}_h^{(n+1)}$ at time t_{n+1}:

$$\mathbf{F}_h(\mathbf{u}_h^{(n+1)}) = \mathbf{L}_h(\mathbf{u}_h^{(n+1)}) + \mathbf{E}_h(\mathbf{u}_h^{(n)}) = 0. \tag{8}$$

The term $\mathbf{E}_h(\mathbf{u}_h^{(n)})$ summarizes all explicit dependences on the solution $\mathbf{u}_h^{(n)}$ at the old time, whereas $\mathbf{L}_h(\mathbf{u}_h^{(n+1)})$ summarizes the implicit, non-linear dependences on the solution $\mathbf{u}_h^{(n+1)}$. As a result of the time discretization, we may write $\mathbf{L}_h = \mathbf{M}_h + \tau \mathbf{A}_h$.

3. Methods

3.1. Nonlinear solvers

Rewriting (8) componentwise as in (5) and (6) yields

$$\mathbf{F}_{p,h}(\mathbf{p}_h^{(n+1)}, \boldsymbol{\omega}_h^{(n+1)}) = 0, \tag{9a}$$

$$\mathbf{F}_{\omega,h}(\mathbf{p}_h^{(n+1)}, \boldsymbol{\omega}_h^{(n+1)}) = 0. \tag{9b}$$

This must be solved by some fixed-point iteration. Various strategies exist; in this study, we focus on three iterative approaches. Since we are interested in the solution at a fixed time t_{n+1}, let us drop the time superscript index $n+1$. Instead we introduce the iteration index as a subscript k, i.e., $\mathbf{u}_{h,k} = (\mathbf{p}_{h,k}, \boldsymbol{\omega}_{h,k})^T$.

3.1.1. Full coupling: Newton method

The standard approach is to employ a Newton method to the fully coupled system. Defining

$$\mathbf{u}_{h,k+1} = \mathbf{u}_{h,k} + \delta \mathbf{u}_h$$

one aims to find a root of the linearized defect equation $\mathbf{F}_h(\mathbf{u}_{h,k+1}) \approx \mathbf{F}_h(\mathbf{u}_{h,k}) + J_h(\mathbf{u}_{h,k})\delta\mathbf{u}_{h,k} = 0$, i.e.,

$$J_h \delta \mathbf{u_{h,k}} = \begin{pmatrix} J_h^{pp} & J_h^{pw} \\ J_h^{wp} & J_h^{ww} \end{pmatrix} \begin{pmatrix} \delta \mathbf{p}_h \\ \delta \boldsymbol{\omega}_h \end{pmatrix} = -\mathbf{F}_h(\mathbf{u}_{h,k}). \tag{10}$$

Here, $\mathbf{F}_h(\mathbf{u}_{h,k})$ from (8) is the nonlinear defect of the current iterate, and $J_h = J_h(\mathbf{u}_{h,k})$ is the Jacobian. The vector $\delta\mathbf{u_h}$ is the resulting correction and search direction respectively. Typically, a line search strategy is employed for a globalization of the method.

3.1.2. Approximate coupling: partial Newton

Modifying (10) slightly one can approximate J_h by its diagonal [1,2]:

$$\tilde{J}_h \delta \mathbf{u_{h,k}} = \begin{pmatrix} J_h^{pp} & 0 \\ 0 & J_h^{ww} \end{pmatrix} \begin{pmatrix} \delta \mathbf{p}_h \\ \delta \boldsymbol{\omega}_h \end{pmatrix} = -\mathbf{F}_h(\mathbf{u}_{h,k}). \tag{11}$$

This strategy is also referred to as *partial Newton method* [1,3]. Note that solving (11) is much easier than solving fully coupled system (10): Since the matrices J_h^{pp} and J_h^{pp} correspond to discretizations of a Poisson-type problem and convection–diffusion equation respectively, good preconditioners are available. As a downside the method will, in general, only provide linear convergence.

One should stress that (11) can be viewed as a single Newton step applied to the system

$$\mathbf{F}_{p,h}(\mathbf{p}_{k+1}, \boldsymbol{\omega}_k) = 0, \tag{12a}$$

$$\mathbf{F}_{\omega,h}(\mathbf{p}_k, \boldsymbol{\omega}_{k+1}) = 0. \tag{12b}$$

This corresponds to an inexact nonlinear Jacobi iteration, where a potential line-search strategy provides a suitable damping factor.

3.1.3. Iterative coupling: nonlinear Gauss–Seidel

As a last alternative, we study a strategy also referred to as *iterative coupling* in the context of different equations [4–6,8]. Starting from (12), we state it as a non-linear Gauss–Seidel type iteration [29] here:

$$\mathbf{F}_{\omega,h}(\mathbf{p}_k, \boldsymbol{\omega}_{k+1}) = 0, \tag{13a}$$

$$\mathbf{F}_{p,h}(\mathbf{p}_{k+1}, \boldsymbol{\omega}_{k+1}) = 0. \tag{13b}$$

Again, both equations treated and solved independently. In contrast to (12), each substep employs the latest update that is available. Following [30], we solve for ω first and then for p. In this (ω, p)-ordering, a new distribution of salt is computed as a results of an unmodified pressure distribution first. Then, in the next step, a suitable pressure is determined. This can be viewed as a projection of the solution into the space where the conservation of fluid mass holds.

Like the partial Newton method from the previous subsection, this approach will at best provide linear convergence. The method is in particular attractive, when flow and transport equation are discretized and solved in different code modules. By adding an additional outer loop, the problem can be solved without any changes to the algorithmic design.

3.2. Preconditioners for the fully coupled Jacobian

The time for the solution process is often to a great extent governed by the computational time required for the solution of the linearized system (10). As multigrid methods feature an optimal computational complexity for a variety of problem, they provide an attractive solver and have frequently been used in this context. In some cases the classical geometric approaches are however not applicable: In some cases geometric methods are not robust (or may even fail) due to the choice of parameters and/or geometric anisotropies. In other cases, the mesh provided initially as a coarse grid may have a substantial size already. This is the case, e.g., fractures or known singularities are resolved on the coarsest mesh.

In these situations the class of algebraic multigrid (AMG) methods provides an attractive alternative. This does not rely on a grid hierarchy, which is provided initially. The key idea is instead to extract all information from the linear system to solve, e.g., from the matrix graph and the matrix entries.

One of the most popular, not to say classic, version of this class has been described in [19,31]. Other approaches include smoothed aggregation, cf. [32–34], or AMG based on element interpolation, cf. [35]. We focus on the filtering algebraic multigrid method (FAMG), previously described in [36,37]. The essential results are however likely to be applicable to different AMG approaches as well. Since the linear systems of equations arising from the decoupling approaches (12) and (13) can be treated by scalar solvers, e.g. [38], we concentrate on solvers for the fully coupled system (10) in this subsection only.

3.2.1. Smoothing analysis and algebraically smooth error

Multigrid is based on the assumption that the relaxation method produces an error that is *smooth* locally. Analogously, the proper choice of AMG components depends on the character of the so-called *algebraically smooth error*, i.e., those error components which are not treated efficiently by the smoother.

We study this further for the modified Elder problem, cf. [39]. This problem induces density driven fingers and is thus, at least for long simulation times, rich in convection. At the same time, the problem features only a milder complexity, since the terms μ and $\mathbb{D}$ are constant and the only non-linearity is in $\rho = \rho(\omega)$. The density function here is selected to be linear:

$$\rho(\omega) := \rho_0 + (\rho_1 - \rho_0)\omega.$$

For this analysis we use the Boussinesq approximation (4) as implemented in the corresponding modules in the *ug4* software toolbox [40] with partial upwinding for the velocity.

Fig. 1 shows the solution $\mathbf{u}_{h,0} = (\mathbf{p}_{h,0}, \boldsymbol{\omega}_{h,0})^T$ at time $t = 5a$. Only the salt mass fraction ω and the velocity field $\mathbf{q}$ are shown. The distribution of the pressure p is less important and thus omitted. The solution $\mathbf{u}_{h,0}$ serves as the linearization point for the first Newton iteration. The time is selected arbitrarily, but it is important to note that several fingers have evolved.

Our goal is to identify error components being *smooth* after several relaxation sweeps. To that end, we relax on the test equation $J_h(\mathbf{u}_{h,0})\delta\mathbf{u}_h = 0$ with a random initial guess $\delta\mathbf{u}_h^{(0)} = (0, \delta\mathbf{r}_\omega)^T$. The coefficients of $\delta\mathbf{r}_\omega$ are uniformly distributed in $(-1, 1)$. Fig. 2 shows the error $\delta\mathbf{u}_h^{(3)} = (\delta\mathbf{p}_h^{(3)}, \delta\boldsymbol{\omega}_h^{(3)})^T$ after three steps of point-block symmetric Gauss–Seidel. The picture is for $\tau = 0.1a$; step lengths $\tau = 0.01, 0.001a$ yield similar results, with a smaller error in the ω component.

In addition to the error, we can also study the residuals

$$\mathbf{r}_h^{(k)} = J_h(\mathbf{u}_h^0)\delta\mathbf{u}_h^{(k)} = (\mathbf{r}_p^{(k)}, \mathbf{r}_\omega^{(k)})^T$$

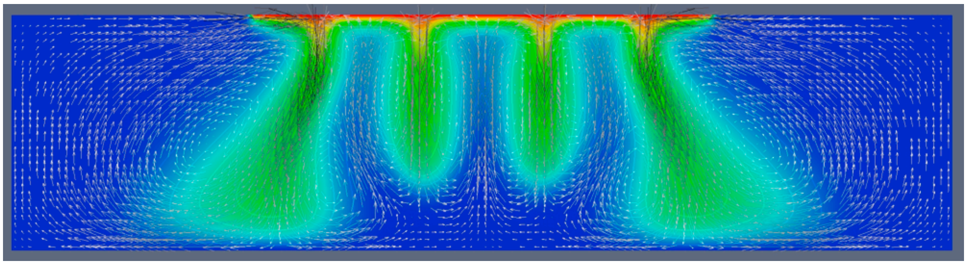

Fig. 1. Solution at time $t = 5a$. In this point, the problem is linearized and then solved with varying time steps ($\tau = 0.1, 0.01, 0.001a$).

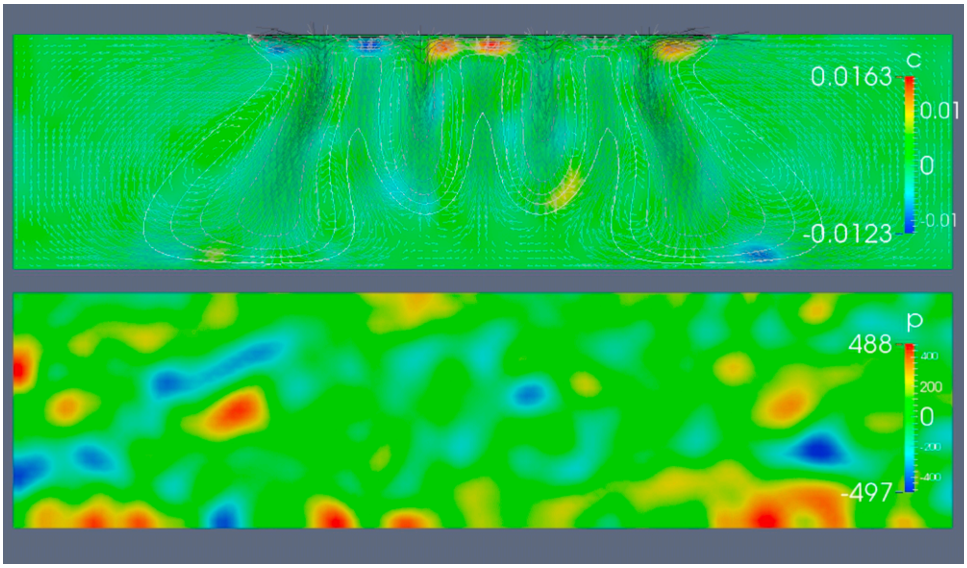

Fig. 2. Components $\delta\boldsymbol{\omega}_h^{(3)}$ (top) and $\delta\mathbf{p}_h^{(3)}$ (bottom) of a smooth error $\delta\mathbf{u}_h^{(3)}$ after 3 relaxation sweeps of a symmetric point-block Gauss–Seidel. The random initial guess was a vector $\delta\mathbf{u}_h^{(0)} = (0, \delta\mathbf{r}_\omega)^T$ with coefficients uniformly distributed in $(-1, 1)$.

for both components independently. Fig. 3 provides data for the reduction of the defects

$$d_{p,k} := \|\mathbf{r}_p^{(k)}\|, \qquad d_{\omega,k} := \|\mathbf{r}_\omega^{(k)}\|$$

for different time steps $\tau \in \{0.1a, 0.01a, 0.001a\}$. One observes that the smaller the time step τ, the smaller the defects. This is in agreement with the discretization where all time dependent terms are multiplied by τ. The large initial defects $d_{\omega,0}$ are rapidly reduced. For increasing numbers of iterations k, the ratio $d_{p,k}/d_{\omega,k}$ approaches a constant. Moreover, the asymptotic convergence rate is independent of τ, which reflects the elliptic contribution of the pressure equation.

3.2.2. *Left transformation*

One problem for FAMG preconditioner described in the next subsection, arises from potentially large negative diagonal entries in the $J_h^{\omega\omega}$ block of (10). This is due the derivative $\mathbf{q}_0'$ in the equations for the salt mass fraction (7b). As a remedy, the following strategy is used [41]:

Instead solving $J\mathbf{u} = \mathbf{f}$ one can alternatively consider a (left-)transformed system

$$LJ\mathbf{u} = L\mathbf{f}. \tag{14}$$

As motivated by analysis for the model problem (7) it is desirable to imitate a decoupling of p and ω locally. This can be achieved by a multiplication with a block-diagonal matrix $L = \mathrm{Diag}(L_{ii})$ which transforms the diagonal

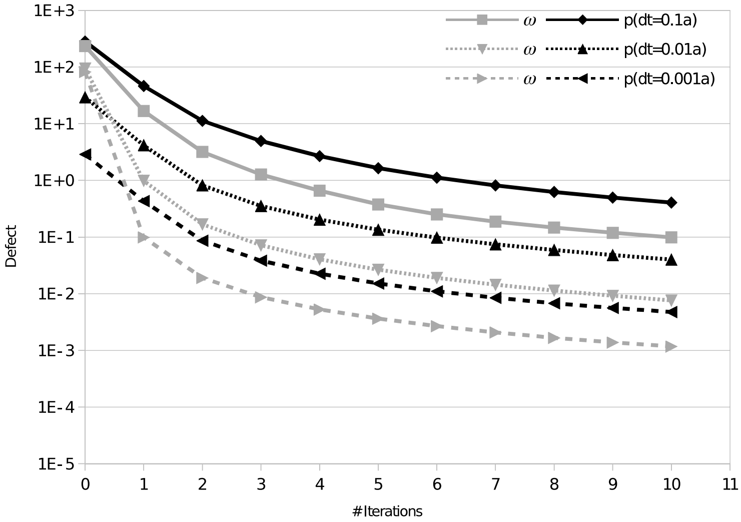

Fig. 3. Reduction of the defects $d_{p,k}$ and $d_{\omega,k}$ as a function of the iteration count k for various time steps ($\tau = 0.1, 0.01, 0.001a$).

entries

$$J_{ii} = \begin{pmatrix} j^{pp} & j^{p\omega} \\ j^{\omega p} & j^{(\omega\omega)} \end{pmatrix} \tag{15a}$$

into

$$(LJ)_{ii} = \begin{pmatrix} j^{pp} & j^{p\omega} \\ 0 & j^{\omega\omega} - \dfrac{j^{\omega p}}{j^{pp}} j^{p\omega} \end{pmatrix}. \tag{15b}$$

This transformation is inspired by previous works for two-phase flow where it is used for eliminating $j^{p\omega}$. These transformations are combined with both combinative (two-stage) preconditioners [42–46] as well as with fully coupled monolithic schemes [47,48]. Since (14) does not affect the convergence properties of standard point-block smoothers, it is attractive for a combination with FAMG for systems as proposed in the next section.

3.2.3. Algebraic multigrid

This section describes an algebraic multigrid (AMG) solver for the transformed system (14). We begin with an outline of the Filtering Algebraic Multigrid (FAMG) for scalar problems [36,37] and then turn to a specialized version for systems of PDEs and density driven flow respectively [21,41]. For the sake of brevity, some algorithmic details are omitted. An extended description is provided in the aforementioned references.

Outline of the setup phase. Most AMG solvers are preceded by a setup phase generating the coarse grids in the grid hierarchy automatically. For a matrix $A \in \mathbb{R}^{V \times V}$ with variables defined on a set of indices V, this is typically accomplished as follows:

1. Splitting the indices V into two disjoint sets $V = F \dot{\cup} C$. The indices C define the coarse variables on the next coarser level. The indices belonging to F correspond to fine variables that can be interpolated.
2. Computing an interpolation operator $P : \mathbb{R}^C \to \mathbb{R}^V$ for a grid transfer of a solution from the coarse to the fine grid.
3. Computing a coarse grid operator. This is typically achieved using the Galerkin product, e.g. $A_C = P^T A P \in \mathbb{R}^{C \times C}$. It is thus sufficient to restrict the following presentation to a two-grid method.

Proceeding with $V \equiv C$, this process can be iterated recursively until $|V|$ is small enough.

FAMG for scalar problems. We begin with a description for a matrix A being symmetric positive definite arising, e.g., from a scalar PDE of Poisson-type [36,37]. The key idea of FAMG is to minimize the norm of the coarse grid operator approximately, while an additional constraint guarantees that certain test vectors are interpolated exactly. Thus, we construct an interpolation operator P as the solution of the following global minimization problem:

$$\min_P \left|\left|\left| X^{1/2}(I - PR')S'X^{-1/2} \right|\right|\right|_2^2, \tag{16a}$$

$$\text{s.t. } (I - PR')\tilde{\mathbf{t}} = 0. \tag{16b}$$

The triple bar norm $|||\cdot|||_2$ refers to the Frobenius norm. $X = \mathrm{Diag}(x_{ii}) > 0$ is positive diagonal matrix providing a scaling, frequently chosen as $X = \mathrm{Diag}(A)$. Moreover S' is the error propagator of a smoother (or an approximation). The operator $P = (W_{FC}^T, I)^T$ is the interpolation and $R' = (0, I)$ is the injection to the coarse grid. The vector $\tilde{\mathbf{t}}$ is a representation of the near null space of the operator, e.g., the constant for a Poisson-type problem.

Let the vector $\mathbf{q}_i$ denote the ith row of $(I - PR')$, i.e.,

$$(\mathbf{q}_i)^T := \varepsilon_i^T (I - PR'),$$

where ε_i is the ith canonical unit vector for $i \in V$. Given a (possible) fine grid node i with a set of interpolatory parent nodes P_i, we have

$$\langle \mathbf{q}_i, \varepsilon_k \rangle = \begin{cases} 1, & i = k, \\ -w_{ik}, & i \in P_i, \\ 0, & k \notin P_i, \end{cases} \tag{17}$$

for $i \in F$, and $\mathbf{q}_i = 0$ for $i \in C$ respectively. With this definition, (16) is equivalent to the following local minimization problem:

$$\min_{\mathbf{q}_i} \; x_{ii} \; \|(S')^T \mathbf{q}_i\|_{X^{-1}}^2, \tag{18a}$$

$$\text{s.t. } \left\langle \mathbf{q}_i, \tilde{\mathbf{t}} \right\rangle = 0 \tag{18b}$$

for each $i \in V$. This formulation can be used to (i) compute an interpolation operator P, and (ii) to determine a suitable set of parent nodes P_i locally. Assigning all indices i with sufficiently small objective to F, and all interpolating parents P_i to C yields an aforementioned splitting of indices.

FAMG for density driven flow. Along the lines of the previous subsection, let us now specify FAMG for density driven flow. The matrix $A = LJ$ of the transformed operator will, in general, be non-symmetric. From Section 3.2.1, we can expect that a point-block-smoother is suitable for multigrid, as well as for defining S'. Moreover, Section 3.2.2 provided evidence that the scalar entries $a_{ii}^{\alpha\alpha}$ are also positive for $i \in V, \alpha \in \{\omega, p\}$. Thus, it is reasonable to define a diagonal scaling $X > 0$ by

$$X = \mathrm{Diag}(x_{ii}^{\alpha\alpha}), \quad x_{ii}^{\alpha\alpha} := |a_{ii}^{\alpha\alpha}|, \; \alpha \in \{p, \omega\}, \; i \in V. \tag{19}$$

For systems of equations, we select a block-diagonal interpolation, i.e. the interpolation weights w_{ik} from above are given by small 2 × 2 matrices

$$w_{ik} = \begin{pmatrix} w_{ik}^{pp} & 0 \\ 0 & w_{ik}^{\omega\omega} \end{pmatrix}$$

for fine grid nodes $i \in F$ and parent nodes $k \in C$. Interpolation for any $i \in V$ is then defined in terms of the block rows $Q_i := (\mathbf{q}_i^\alpha)_\alpha$, analogous to (18):

$$\min_{Q_i} \sum_\alpha x_{ii}^{\alpha\alpha} \|S'^T \mathbf{q}_i^\alpha\|_{X^{-1}}^2, \tag{20a}$$

$$\text{s.t. } \langle \mathbf{q}_i^\alpha, \tilde{\mathbf{t}}^\alpha \rangle = 0, \quad \forall \alpha. \tag{20b}$$

The test vectors $\tilde{\mathbf{t}}^\alpha$ are the constants for each component $\alpha \in \{p, \omega\}$. The smoother S' is a damped Jacobi, followed by a full Jacobi step that is only applied at node i. The objective (20a) provides a measure, how valuable $i \in V$ is as a fine grid node. Since the smoother may diverge, we only select those nodes with $-0.1 \leq \langle S'\epsilon_i^\alpha, \epsilon_i^\alpha \rangle \leq 0.5, i \in p, \omega$.

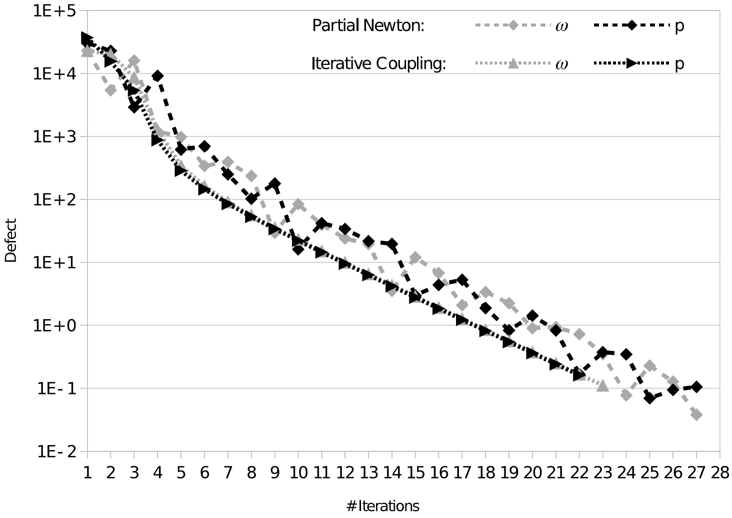

Fig. 4. Reduction of the defects $d^{nl}_{p,k}$ and $d^{nl}_{\omega,k}$ for partial Newton (diamond) and iterative coupling (triangle) for computing $t = \tau = 0.025a$ in the first time step.

4. Numerical experiments

4.1. Nonlinear solvers

Section 3.1 introduced three different solvers. These are compared and evaluate with respect to performance. As a benchmark we use the Elder problem. This features a highly dynamic velocity field in the beginning, which then stabilizes for larger times. Since effects of the linear solver should be avoided, we use a coarse spatial mesh with 4420 degrees of freedom (1024 elements). The tests are conducted for the full non-linear equations (1) using *ug4* [40].

The first test investigates the convergence of the two decoupling nonlinear solvers. In the first time step, the partial Newton and the iterative coupling achieve a reduction of the residual by 0.5×10^{-6} in 26 and 21 steps respectively. Fig. 4 visualizes details about the reduction of the nonlinear defects

$$d^{nl}_{p,k} := \|\mathbf{F}_{p,h,k}\|_2, \qquad d^{nl}_{\omega,k} = \|\mathbf{F}_{\omega,h,k}\|_2$$

of the k-th iterate. The vectors $\mathbf{F}_{\alpha,h,k}$ are defined by (9) for both components $\alpha \in \{p, \omega\}$. Both methods converge linearly with similar rates of convergence. However, one observes that $d^{nl}_{p,k}$ and $d^{nl}_{\omega,k}$ behave differently: While they resemble each other in the order of magnitude for the iterative coupling, they differ substantially for the partial Newton. In the latter case, an oscillating behavior can be observed.

Fig. 5 provides a history of the nonlinear iteration steps required for each single time step over a complete simulation run of 5 years. As expected, the full Newton method performs best. Large time steps are permitted ($\tau = 0.1a$), at the same time, only a constant number of 4 iterations per time step, i.e., a total of 80 iterations is required. Both decoupling iterations require a smaller time step $\tau = 0.025a$. In the comparison for a full simulation run, however, they behave differently: The partial Newton requires 25–30 iterations per time step, resulting in a total of 6384 iterations. Although the iterative coupling also starts with ~25 iterations in the first steps, the number of required iterations gradually decreases to 8–10 iterations per time step. This leads to a total of 1925 iterations.

According to this data, the iterative coupling should be preferred over a partial Newton. In particular, if the alterations in the velocity field are small, i.e., the corresponding initial guess for the nonlinear scheme is sufficiently good, the Gauss–Seidel-style arrangement of the iterative coupling seems to be more appropriate. However, as one can expect, the full Newton method outperforms both previously mentioned methods. The reason is that decoupling iterations produce iterates that oscillate around the fixed point: For the iterative coupling, for example, an erroneous velocity in the transport equation (13a) tends to overshoot the concentration, which is then corrected in the next step (13b) by the flow equation. Similar results have previously been reported, e.g., for fluid structure interactions [49,50].

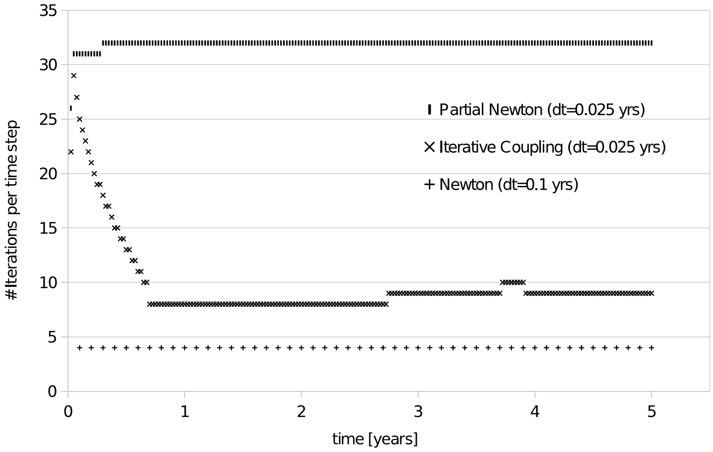

Fig. 5. Iterations per time step for a full simulation run: Newton (time step $\tau = 0.1a$), partial Newton ($\tau = 0.025a$), and iterative coupling ($\tau = 0.025a$).

4.2. *Algebraic multigrid preconditioners for the Jacobian*

After underlining the advantages of a fully coupled Newton in the previous subsection, we now evaluate the algebraic multigrid preconditioners developed for these kind of systems in Section 3.2.3.

We compare three different solvers: The first is a standard geometric multigrid solver (GMG). The second is a (ω, p)-Block–Gauss–Seidel scheme (BGS). Blocks are formed by unknowns according to (15b), and one solves for ω first, and then for p. This provides an exact solver for problems with $A^{\omega p} = 0$. Finally, we consider a monolithic FAMG solver. In this case both unknowns are solved simultaneously. All methods serve as a preconditioner in a Bi-CGSTAB method. We use a V(1, 1)-cycle with a point-block symmetric Gauss–Seidel smoother [16]. Note that the presented comparisons must be understood as a proof of concept: As the AMG method requires an additional setup, they are usually less efficient than their geometric counter parts. The performance of the method is studied for two different test cases. In both cases, the fully coupled non-linear system (1) is solved using the software package d^3f [15].

4.2.1. *Test 1: saltpool problem*

The first model problem is derived from the saltpool benchmark, e.g.,. [12]. The domain is a box$(0, \sqrt{2}a) \times (0, 1)$ which is resolved by a structured quadrilateral grid ($2 \times 66\,049$ dof). The parameter a characterizes the anisotropy in x-direction. Initially, the box is filled with a 10% salt mass fraction in the lower 30% of the box. In the course of the experiment, fresh water is injected in the upper left corner, and removed in the upper right corner. The walls are treated as impermeable boundaries. Since we only have few Dirichlet nodes, we use an iterative coarse grid solver for FAMG, which can deal with matrices that are close to singular.

Numerical results are provided in Table 1: As expected geometric multigrid deteriorates slightly with an increasing anisotropy a. The performance of BGS and FAMG is robust w.r.t. a. The convergence good convergence of BGS however is an indicator that we essentially observe a non-linear decoupling: Provided that we have large difference in the density, salt and fresh water phases essentially decouple. In the mixing region, the flow actually follows the boundary layer, and mixing is primarily due to diffusion.

4.2.2. *Test 2: layered aquifer*

The second problem serves as a test how well the method treat problems with variable and discontinuous permeabilities and variable velocity profile. Fig. 6 shows the coarse grid for a computational domain of an aquifer over a salt dome. The triangular, unstructured grid has $2 \times 24\,257$ degrees of freedom. The hydro-geological properties, in particular the permeabilities vary in space and follow a log-normal distribution. Two models are considered: The *Mixed Model* includes sand, fine sand and silt, whereas the *Sand model* has only one hydro-geological unit.

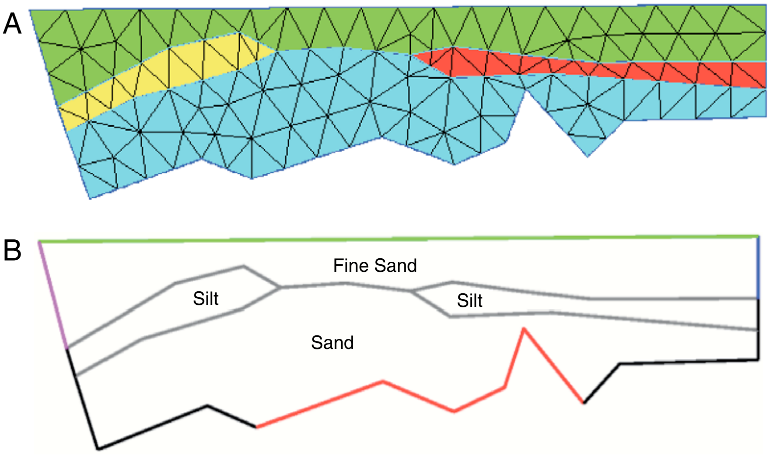

Fig. 6. (A) Coarse grid and (B) different hydro-geological areas and boundary conditions for the layered aquifer. On the red surface on the bottom the fluid is in contact with a salt dome. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Convergence results for the salt pool problem. Given are the number of time steps, the total number of Newton iterations (#nl), the total number of linear iterations (#lin) as well as the maximum and average number of linear iterations per Newton iteration(#lin/#nl).

Anisotropy *a*	Method	Steps	#nl	#lin	#lin/#nl (max)	#lin/#nl (ave)
	GMG	49	132	1183	53	8.96
1	BGS	44	117	2603	100	22.25
	FAMG	43	113	839	13	7.42
	GMG	46	113	1356	65	12.00
2	BGS	43	109	2295	63	21.06
	FAMG	43	109	826	14	7.58
	GMG	43	99	1529	43	12.72
4	BGS	43	101	2213	74	21.91
	FAMG	43	101	660	11	6.53
	GMG	43	113	2239	51	19.8
8	BGS	43	111	2231	70	20.01
	FAMG	43	111	705	13	6.35

Table 2
Convergence results for the aquifer problem, with column data as in Table 1. The BGS scheme does not converge.

Medium	Method	Steps	#nl	#lin	#lin/#nl (max)	#lin/#nl (ave)
Mixed	GMG	18	157	4 186	99	26.66
	FAMG	15	127	2 048	87	16.13
Sand	GMG	40	326	15 297	119	46.92
	FAMG	30	260	7 125	97	27.40

Numerical results for this model are provided in Table 2. We first note that the BGS method does not converge any more which indicates that problems are now completely coupled. The FAMG method tends to be a little bit more robust than geometric multigrid. In particular, it requires less time steps.

5. Conclusion

This study elucidated on two important aspects for solvers for density driven flow problems: First, we investigated different nonlinear solvers and compared their convergence properties in numerical experiments. Here, a fully coupled scheme proved being superior to decoupling schemes. Second, we tailored an multigrid solver for a fully coupled Newton method. The essential ingredient is a left transformation that eliminates possible negative entries arising from the self couplings for the salt mass fraction.

Acknowledgments

The authors would like to thank Mary F. Wheeler who pointed out the potential application of iterative coupling techniques, and Klaus Johannsen for all discussions on density driven flow, and, for emphasizing (7) in particular. Parts of this work were supported by the German Federal Ministry of Economics and Technology (BMWi) by grant nos. 02E10326 and 02E11062.

References

[1] M. Putti, C. Paniconi, Picard and Newton linearization for the coupled model for saltwater intrusion in aquifers, Adv. Water Resour. 18 (3) (1995) 159–170. http://dx.doi.org/10.1016/0309-1708(95)00006-5.

[2] H.-J.G. Diersch, O. Kolditz, Coupled groundwater flow and transport: 2. Thermohaline and 3D convection systems, Adv. Water Resour. 21 (5) (1998) 401–425. http://dx.doi.org/10.1016/S0309-1708(97)00003-1.

[3] H.-J.G. Diersch, O. Kolditz, Variable-density flow and transport in porous media: approaches and challenges, in: FEFLOW©White Paper Volume II, DHI-WASY GmbH, Berlin, 2009.

[4] S. Lacroix, Y.V. Vassilevski, M.F. Wheeler, Decoupling preconditioners in the implicit parallel accurate reservoir simulator (IPARS), Numer. Linear Algebra Appl. 8 (8) (2001) 537–549. http://dx.doi.org/10.1002/nla.264.

[5] B. Lu, M. Wheeler, Iterative coupling reservoir simulation on high performance computers, Pet. Sci. 6 (2009) 43–50. http://dx.doi.org/10.1007/s12182-009-0008-x.

[6] J. Kim, H. Tchelepi, R. Juanes, Stability, accuracy, and efficiency of sequential methods for coupled flow and geomechanics, SPE J. 16 (2) (2011) 249–262.

[7] J. Kim, H. Tchelepi, R. Juanes, Stability and convergence of sequential methods for coupled flow and geomechanics: fixed-stress and fixed-strain splits, Comput. Methods Appl. Mech. Engrg. 200 (13–16) (2011) 1591–1606. http://dx.doi.org/10.1016/j.cma.2010.12.022.

[8] A. Mikelić, M.F. Wheeler, Convergence of iterative coupling for coupled flow and geomechanics, Comput. Geosci. 17 (3) (2013) 455–461.

[9] A. Mikelić, B. Wang, M.F. Wheeler, Numerical convergence study of iterative coupling for coupled flow and geomechanics, Comput. Geosci. (2014) 1–17. http://dx.doi.org/10.1007/s10596-013-9393-8.

[10] C.D. Langevin, W. Guo, MODFLOW/MT3DMS-based simulation of variable-density ground water flow and transport, Ground Water 44 (3) (2006) 339–351. http://dx.doi.org/10.1111/j.1745-6584.2005.00156.x.

[11] C.D. Langevin, J. Daniel, T. Thorne, A.M. Dausman, M.C. Sukop, W. Guo, US Geological Survey Techniques and Methods Book 6, US Geological Survey, Reston, Virginia, 2008, Ch. A22: SEAWAT version 4: A Computer Program for Simulation of Multi-Species Solute and Heat Transport.

[12] K. Johannsen, W. Kinzelbach, S. Oswald, G. Wittum, The saltpool benchmark problem—numerical simulation of saltwater upconing in a porous medium, Adv. Water Resour. 25 (2002) 309–1708.

[13] S. Lang, G. Wittum, Large scale density driven flow simulations using parallel unstructured grid adaptation and local multigrid methods, Concurr. Comput. 17 (11) (2005) 1415–1440.

[14] A. Grillo, M. Lampe, G. Wittum, Three-dimensional simulation of the thermohaline-driven buoyancy of a brine parcel, Comput. Vis. Sci. 13 (2010) 287–297.

[15] E. Fein, A. Schneider, d^3f—Ein Programmpaket zur Modellierung von Dichteströmungen, Tech. Rep. GRS 139, Gesellschaft für Anlagenbau und Reaktorsicherheit (GRS) mbH, 1999, October.

[16] K. Johannsen, Numerische Aspekte dichtegetriebener Strömung in porösen Medien (Habilitationsschrift), 2004.

[17] W. Hackbusch, Multi-Grid Methods and Applications, Springer, Berlin, 1985.

[18] J. Xu, Iterative methods by space decomposition and subspace correction, SIAM Rev. 34 (4) (1992) 581–613.

[19] J.W. Ruge, K. Stüben, Algebraic multigrid (AMG), in: Multigrid Methods, in: Frontiers in Applied Mathematics, vol. 3, SIAM, Philadelphia, PA, 1987, pp. 73–130. (Chapter).

[20] K. Stüben, An Introduction to Algebraic Multigrid, Academic Press, 2001, pp. 413–532. Ch. Appendix A.

[21] A. Nägel, Schnelle Löser für große Gleichungssysteme mit Anwendungen in der Biophysik und den Lebenswissenschaften (Ph.D. thesis), Universität Heidelberg, 2010.

[22] J. Bear, Y. Bachmat, Introduction to Modeling of Transport Phenomena in Porous Media, in: Theory and Applications of Transport in Porous Media, Kluwer Academic, Dordrecht, 1991.

[23] A. Leijnse, Three-dimensional modeling of coupled flow and transport in porous media (Ph.D. thesis), University of Notre Dame, Indiana, 1992.

[24] E. Holzbecher, Modeling Density-Driven Flow in Porous Media, Springer, Berlin, Heidelberg, 1998.

[25] A. Oberbeck, Über die Wärmeleitung der Flüssigkeiten bei Berücksichtigung der Strömungen infolge von Temperaturdifferenzen, Ann. Phys. Chem. (1879) 271–292.

[26] J. Boussinesq, Theorie Analytique de la Chaleur, Vol. 2, Gauthier-Villlars, Paris, 1903.

[27] K. Johannsen, On the validity of the Boussinesq approximation for the Elder problem, Comput. Geosci. 7 (3) (2003) 169–182.

[28] K. Johannsen, Numerical aspects of density driven flow in porous media, in: XVI International Conference on Computational Methods in Water Resources, 2006. http://dx.doi.org/10.4122/1.1000000246.

[29] W.C. Rheinboldt, Methods for Solving Systems of Nonlinear Equations, second ed., SIAM, 1998.

[30] P. Ackerer, A. Younes, M. Mancip, A new coupling algorithm for density-driven flow in porous media, Geophys. Res. Lett. 31 (2004) L12506. http://dx.doi.org/10.1029/2004GL019496. 1–4.

[31] K. Stüben, A review of algebraic multigrid, J. Comput. Appl. Math. 128 (1–2) (2001) 281–309. http://dx.doi.org/10.1016/S0377-0427(00)00516-1.

[32] P. Vanek, J. Mandel, M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing 56 (1996) 179–196. http://dx.doi.org/10.1007/BF02238511.

[33] J. Mandel, M. Brezina, P. Vanek, Energy optimization of algebraic multigrid bases, Computing 62 (1999) 205–228. http://dx.doi.org/10.1007/s006070050022.

[34] P. Vanek, M. Brezina, J. Mandel, Convergence of algebraic multigrid based on smoothed aggregation, Numer. Math. 88 (3) (2001) 559–579.

[35] M. Brezina, A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. McCormick, J.W. Ruge, Algebraic multigrid based on element interpolation (AMGe), SIAM J. Sci. Comput. 22 (5) (2001) 1570–1592.

[36] C. Wagner, On the algebraic construction of multilevel transfer operators, Computing 65 (2000) 73–95.

[37] A. Naegel, R. Falgout, G. Wittum, Filtering algebraic multigrid and adaptive strategies, Comput. Vis. Sci. 11 (3) (2008) 159–167. http://dx.doi.org/10.1007/s00791-007-0066-9.

[38] H.-J.G. Diersch, Using and testing the algebraic multigrid solver samg in FEFLOW, in: FEFLOW©White Paper Volume III, DHI-WASY GmbH, Berlin, 2009.

[39] C. Voss, W. Souza, Variable density flow and solute transport simulation of regional aquifers containing a narrow freshwater–saltwater transition zone, Water Resour. Res. 26 (1987) 2097–2106.

[40] A. Vogel, S. Reiter, M. Rupp, A. Nägel, G. Wittum, UG 4: A novel flexible software system for simulating PDE based models on high performance computers, Comput. Vis. Sci. (2014) 1–15. http://dx.doi.org/10.1007/s00791-014-0232-9.

[41] A. Vogel, A. Nägel, S. Reiter, Numerical advances, in: A. Schneider (Ed.), Enhancement of the codes d^3f and r^3t (GRS-292), Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, 2012, pp. 156–196.

[42] A. Behie, P. Vinsome, Block iterative methods for fully implicit reservoir simulation, Soc. Pet. Eng. J. 22 (1982) 658–668. http://dx.doi.org/10.2118/9303-PA.

[43] J. Wallis, R. Kendall, T. Little, Constrained residual acceleration of conjugate residual methods, in: SPE 13536, 1985, pp. 415–426.

[44] S. Lacroix, Y. Vassilevski, J. Wheeler, M. Wheeler, Iterative solution methods for modeling multiphase flow in porous media fully implicitly, SIAM J. Sci. Comput. 25 (3) (2003) 905–926. http://dx.doi.org/10.1137/S106482750240443X.

[45] R. Scheichl, R. Masson, J. Wendebourg, Decoupling and block preconditioning for sedimentary basin simulations, Comput. Geosci. 7 (2003) 295–318. http://dx.doi.org/10.1023/B:COMG.0000005244.61636.4e.

[46] H. Klíe, M.F. Wheeler, Advanced solver methods for subsurface environmental problems, Tech. Rep., The University of Texas at Austin, Austin, TX, 78712, 2005.

[47] K. Stüben, T. Clees, H. Klie, B. Lu, M. Wheeler, Algebraic multigrid methods (AMG) for the efficient solution of fully implicit formulations in reservoir simulation, in: 2007, Paper SPE 105832 Presented at the 2007 SPE Reservoir Simulation Symposium, Houston, TX, February 28–30, 2007.

[48] T. Clees, L. Ganzer, An efficient algebraic multigrid solver strategy for adaptive implicit methods in oil reservoir simulation, in: Paper SPE 105789 Presented at the 2007 SPE Reservoir Simulation Symposium, Houston, TX, February 28–30, 2007.

[49] M. Heil, An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems, Comput. Methods Appl. Mech. Engrg. 193 (1–2) (2004) 1–23. http://dx.doi.org/10.1016/j.cma.2003.09.006.

[50] H. Matthies, R. Niekamp, J. Steindorf, Algorithms for strong coupling procedures, Comput. Methods Appl. Mech. Engrg. 195 (17–18) (2006) 2028–2049. http://dx.doi.org/10.1016/j.cma.2004.11.032.