DECOUPLING AND BLOCK PRECONDITIONING FOR SEDIMENTARY BASIN SIMULATIONS IN TEMIS3D*

ROBERT SCHEICHL[†], ROLAND MASSON[‡], AND JOHANNES WENDEBOURG[§]

Abstract. The simulation of sedimentary basins aims at reconstructing its historical evolution in order to provide quantitative predictions about phenomena leading to hydrocarbon accumulations. The kernel of this simulation is the numerical solution of a complex system of time dependent, three-dimensional partial differential equations (PDE) of mixed parabolic-hyperbolic type. A discretisation (Finite Volumes + Implicit Euler) and linearisation (Newton) of this system leads to very ill-conditioned, strongly non-symmetric and large systems of linear equations with three unknowns per mesh element, i.e. pressure, geostatic load, and hydrocarbon saturation.

The preconditioning which we will present for these systems consists in three stages. First of all the equations for pressure and saturation are locally decoupled on each element. This decoupling aims not only at reducing the coupling, but also at concentrating in the "pressure block" the elliptic part of the system which is then in the second stage preconditioned by efficient methods like AMG. The third step finally consists in "recoupling" the equations (e.g. Block Gauss-Seidel, combinative techniques,...).

In almost all our numerical tests on real test problems from case studies we observed a considerable reduction of the CPU-time for the linear solver, up to a factor 4.3 with respect to ILU(0) preconditioning (which is used at the moment in `TEMIS3D`). The performance of the preconditioner shows no degradation with respect to the number of elements, the size of the time step, high migration ratios, or strong heterogeneities and anisotropies in the porous media.

Key words. three-dimensional multiphase Darcy flow, compaction of porous media, numerical solution, block preconditioning, algebraic multigrid

AMS subject classifications. 65F10, 65M99, 65Y20, 74F10, 74S10, 76T99

1. Introduction.

In recent years the 3D modelling and simulation of sedimentary basins has become an integral part in the exploration of present and future reservoirs for almost all major oil companies. Ideally a basin simulator should span the entire process of source rock burial, hydrocarbon generation, expulsion, migration into a potential trap, and assessment of trap integrity throughout the evolution of a basin. The code `TEMIS3D` developed by the Institut Français du Pétrole (IFP) and marketed by its subsidiary Beicip-Franlab represents such a tool. It aims at reconstructing the historical evolution of a sedimentary basin in order to provide quantitative predictions about phenomena leading to hydrocarbon accumulations.

Given present-day maps of the distribution of lithology, bathymetry, temperature, thickness, heat flux, kerogen and total organic content (TOC) and "paleogeological" maps describing those distributions throughout the history of the basin we would ideally like to calculate the evolution of the basin and of its geometry, and the migration of the hydrocarbons backward in time. However, the solution of this inverse problem is way beyond our means. Therefore only an extremely simplified model is calculated backward in time to reconstruct the geometry of the basin (*back-stripping*). The full model comprising the process of hydrocarbon generation, heat transfer, compaction of the porous media, pressure generation, multi-phase Darcy flow and oil and gas migration is then calculated forward in time adding layer after layer using the geometric history calculated during the back-stripping process.

---

*This work was financed through the Marie-Curie Fellowship HPMI-CT-1999-00012.

[†]Department of Mathematical Sciences, University of Bath, BA2 7AY, UK (`masrs@bath.ac.uk`).

[‡]Division Informatique Scientifique et Mathématiques Appliquées, Institut Français du Pétrole, Rueil-Malmaison, France.

[§]Division Geologie–Geochimie, Institut Français du Pétrole, Rueil-Malmaison, France.

Thus, a 3D case study is composed of three major steps: 3D block building and mesh construction, back-stripping, and forward simulation (see Schneider et al. [9] for an example of such a case study). The 3D block building step, usually carried out by the geologist, consists in preparing the necessary data, i.e. in defining a coherent 3D block which represents the studied area and in assembling with this block a set of geological maps. It also consists in defining a 3D mesh on the block which will usually be Cartesian in the horizontal plane and such that the vertical columns coincide with the chronostratigraphic columns. In the back-stripping step the historical evolution of the geometry of the basin will then be calculated separate for each column (Multi-1D) backward in time by taking off sedimented material and adding eroded material. At each time step once the sedimentation and the erosion have been accounted for, the remaining sediments are decompacted or compacted, respectively, by using porosity/depth relationships for each of the lithologies. The final step is a forward simulation of the full 3D model for the sedimentary basin on the moving geometry (mesh) calculated during the back-stripping phase.

In this paper we will only be looking at the forward simulation which is the most computing intensive part and consists in solving numerically a complex system of time dependent, three-dimensional partial differential equations (PDEs) modelling heat transfer, compaction of the porous media and multi-phase Darcy flow. These equations are discretised using a cell-centred finite volume method in space and an implicit Euler scheme in time. However, to simplify the treatment of the resulting algebraic system we apply a splitting technique and thus decouple the temperature from the other unknowns. In the case of two-phase flow (water – oil) we will therefore have to solve at each time step a system of coupled non-linear equations in four unknowns per element (pressure, geostatic load, porosity, oil saturation), followed by a system of linear equations for the temperature. Here we focus on the solution of the coupled nonlinear system and thus from now on we will consider the temperature as given. A linearisation of the system of non-linear equations using a Newton method and an elimination of the porosity will finally lead to a system of linear equations of the block form

$$A\boldsymbol{x} := \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \end{pmatrix} =: \boldsymbol{b}\,. \tag{1.1}$$

System (1.1) is usually very large, strongly non-symmetric and very ill-conditioned. For an efficient and robust iterative solution of this system, it is therefore crucial to find a good preconditioner. It will be the subject of this paper to develop such a preconditioner and to evaluate its robustness and efficiency by applying it to a number of representative test cases from real case studies and by comparing with standard preconditioning techniques employed at the moment in `TEMIS3D`.

Since this is an entirely novel problem, there is no literature available, but we can nevertheless make strong use of the vast literature for the related problem of oil reservoir simulation. In particular, the idea for our preconditioning strategy is based on the methods developed by Behie & Vinsome [2], Edwards [3], Lacroix, Vassilevski & Wheeler [5], and Lacroix, Vassilevski, Wheeler & Wheeler [6]. It consists of three stages. First of all the equations for pressure and saturation are locally decoupled by taking linear combinations of the two equations on each element. This corresponds to multiplying (1.1) by a matrix G, i.e.

$$A\boldsymbol{x} = \boldsymbol{b} \qquad \longrightarrow \qquad (GA)\,\boldsymbol{x} = (G\boldsymbol{b}).$$

This decoupling aims not only at reducing the coupling, but also at concentrating in the (pressure) block $(GA)_{11}$ the elliptic part of the system which requires a good global preconditioning. The second stage consists then in preconditioning $(GA)_{11}$ by Algebraic Multigrid (AMG). The *recoupling* of the pressure with the other unknowns is then in a final stage achieved by applying a block Gauss-Seidel strategy or the combinative technique defined in [2] for the reservoir case.

In a series of numerical tests from real case studies, the performance of the preconditioner is compared to ILU(0) which is used at the moment in `TEMIS3D`. In almost all cases we observe a considerable reduction of the CPU-time for the linear solver, up to a factor 4.3 with respect to ILU(0). The performance of the preconditioner shows no degradation with respect to the number of elements, the size of the time step, high migration ratios, or strong heterogeneities and anisotropies in the porous media.

The paper is organised as follows. In Section 2 we will present the mathematical model for the forward problem, and describe briefly its discretisation and linearisation. In Section 3 we will give a detailed description of the preconditioning strategy, followed by a series of numerical tests in Section 4. We finish with some conclusions and an outlook on future investigations in Section 5.

2. Mathematical model.

2.1. The continuous problem. In this section we present the mathematical model for the forward problem (see e.g. Schneider et al. [9] and the references therein). A more rigorous analysis of multi-phase Darcy flow in porous media in conjunction with reservoir simulation can be found in Gagneux et al. [4]. We will only discuss the two-phase case (water – oil) in detail, but we note that the three-phase case (water – oil – gas) can be treated similarly. As noted above we will not include the heat equation in this model, since it will be decoupled from the rest of the problem in the solution process by a splitting method. Thus, the temperature distribution will in the following be considered as given at each point in time.

The model which is used in the forward simulation comprises the following equations:

Conservation of mass for water

$$\frac{\partial}{\partial t}\Big(\rho_w \phi S_w\Big) + \mathrm{div}\,\Big(\rho_w \phi S_w \vec{V}_w\Big) = \rho_w q_w \ , \tag{2.1}$$

Conservation of mass for oil

$$\frac{\partial}{\partial t}\Big(\rho_o \phi S_o\Big) + \mathrm{div}\,\Big(\rho_o \phi S_o \vec{V}_o\Big) = \rho_o q_o \ , \tag{2.2}$$

Conservation of (vertical) momentum[1]

$$\frac{\partial \sigma_z}{\partial z} = \Big(\phi \rho_f + (1-\phi)\rho_s\Big) g \ , \tag{2.3}$$

Porous medium rheology (elastoviscoplastic law)[2]

$$\frac{d_s \phi}{d_s t} = -\beta(\phi,\sigma)\,\frac{d_s \sigma}{d_s t} - \alpha(\phi,\sigma)\,\sigma \ , \tag{2.4}$$

[1] The coordinate system is chosen such that the z-axis is pointing vertically downwards.

[2] Here $\frac{d_s\,\cdot}{d_s t}$ represents the *total time derivative* $\frac{\partial\,\cdot}{\partial t} + \vec{V}_s \cdot \vec{\nabla}(.)$.

Generalised Darcy's law for each phase

$$\phi S_w \left(\vec{V}_w - \vec{V}_s\right) = -\eta_w(S_w) K(\phi) \left(\vec{\nabla} P_w - \rho_w g \vec{\nabla} z\right) \ , \tag{2.5}$$

$$\phi S_o \left(\vec{V}_o - \vec{V}_s\right) = -\eta_o(S_o) K(\phi) \left(\vec{\nabla} P_o - \rho_o g \vec{\nabla} z\right) \ . \tag{2.6}$$

This constitutes a system of 6 equations for the 9 unknown functions

S_α	saturation (or volumetric fraction in the fluid) of phase α (for $\alpha = w, o$),
P_α	pore pressure of the phase α (for $\alpha = w, o$),
σ_z	geostatic load (or vertical stress),
σ	mean effective stress,
ϕ	porosity of the porous medium,
$\vec{V}_\alpha$	mean velocity of the phase α (for $\alpha = w, o$).

The other terms appearing in equations (2.1–2.6) are parameters of the system, which depend in some cases on the unknown functions defined above or on the *temperature* T (assumed given at each point in time). They are

q_w	source term for water (sedimentation, erosion),
$q_o := q_o(T)$	source term for oil (hydrocarbon generation, i.e. cracking),
$\rho_\alpha := \rho_\alpha(T)$	density of the phase α as a function of T only,
$\rho_f := S_w \rho_w + S_o \rho_o$	average fluid density,
ρ_s	density of the solid "phase",
g	acceleration due to gravity,
$\alpha := \alpha(\phi, \sigma)$	rheology parameter,
$\beta := \beta(\phi, \sigma)$	rheology parameter,
$K := K(\phi)$	permeability tensor (*Koseny–Carman formula*),
$\eta_\alpha := \dfrac{kr_\alpha(S_\alpha)}{\mu_\alpha(T)}$	mobility of phase α, where kr_α denotes relative permeability and μ_α denotes viscosity (*Andrade formula*),
$\vec{V}_s$	velocity of the solid "phase" (moving geometry and mesh).

For an exact definition of those parameters and their dependency on the unknown functions see Schneider et al. [9, Appendix 1].

To close the system we need to impose further conditions. First of all, in a saturated medium the saturations satisfy the pore volume conservation equation

$$S_w \ = \ 1 - S_o \ , \tag{2.7}$$

and by introducing the *average pore pressure* P_f as a new unknown we can write

$$\sigma = \sigma_z - P_f \ , \tag{2.8}$$

$$P_w = P_f - S_o P_{cap}(S_w, S_o) \ , \tag{2.9}$$

$$P_o = P_f + S_w P_{cap}(S_w, S_o) \ , \tag{2.10}$$

where $P_{cap} := P_{cap}(S_w, S_o)$ denotes the *capillary pressure.* This closes the system and results in 10 equations for the 10 unknowns $S_w, S_o, P_w, P_o, P_f, \sigma_z, \sigma, \phi, \vec{V}_w, \vec{V}_o$.

We can now substitute equations (2.5–2.10) into equations (2.1–2.4) to obtain a system of two partial differential equations and two ordinary differential equations in the four unknowns S_o, P_f, σ_z, ϕ, which will be the *primary* unknowns of our model.

Given a bounded domain $\Omega := \Omega(t)$ with boundary $\Gamma := \Gamma(t)$ and outward unit normal $\vec{n} := \vec{n}(t)$, the forward simulation consists now in numerically solving this system on Ω subject to suitable boundary conditions and initial conditions. The particular form of Ω at each point t in time is the one calculated in the back-stripping phase. Here, we will only consider domains Ω with a constant rectangular vertical projection, a fixed lower boundary Γ_B, a strictly vertical lateral boundary $\Gamma_L := \Gamma_L(t)$ and a *free* upper boundary $\Gamma_T := \Gamma_T(t)$. In addition we will use the following initial conditions and boundary conditions.

2.1.1. Initial conditions. At $t = t^{(0)}$ we have

$$\begin{aligned}
\Gamma_T\big(t^{(0)}\big) &= \Gamma_B \ , \\
S_o\big(t^{(0)}\big) &= 0 \ , \\
P_f\big(t^{(0)}\big) &= P_{atm}\big(t^{(0)}\big) + \rho_w g h_w\big(t^{(0)}\big) \ , \\
\sigma_z\big(t^{(0)}\big) &= P_{atm}\big(t^{(0)}\big) + \rho_w g h_w\big(t^{(0)}\big) \ , \\
\phi\big(t^{(0)}\big) &= \phi_0\Big(\sigma_z\big(t^{(0)}\big)\Big) \ ,
\end{aligned}$$

where $P_{atm}(t)$ denotes the *atmospheric pressure* at time t, $h_w(t)$ denotes the height of the water column above $\Gamma_T(t)$ at time t, and $\phi_0(\sigma_z)$ denotes the porosity of the bottom layer as a function of the geostatic load σ_z.

2.1.2. Boundary conditions. The system is of mixed parabolic-hyperbolic type in pressure and saturation (coupled with two ordinary differential equations (2.3) and (2.4)). This is also reflected in the rather unusual boundary conditions for pressure and saturation:

On Γ_B: (1) $K\left(\vec{\nabla}P_w - \rho_w g\vec{\nabla}z\right)\cdot\vec{n} = 0$

(2) $K\left(\vec{\nabla}P_o - \rho_o g\vec{\nabla}z\right)\cdot\vec{n} = 0$.

On Γ_L: **either** (1) and (2) as for Γ_B **or**

(1′) $S_o = 0$ if $K\left(\vec{\nabla}P_w - \rho_w g\vec{\nabla}z\right)\cdot\vec{n} > 0$ or $K\left(\vec{\nabla}P_o - \rho_o g\vec{\nabla}z\right)\cdot\vec{n} > 0$

(2′) $\phi S_w\vec{V}_w\cdot\vec{n} = F_L(t)$ **or** $P_w = P_{hyd}(t)$ **or** $\frac{P_w - \rho_w g z}{\rho_w g} = \Psi_L(t)$.

On Γ_T: (1) $S_o = 0$ if $K\left(\vec{\nabla}P_w - \rho_w g\vec{\nabla}z\right)\cdot\vec{n} > 0$ or $K\left(\vec{\nabla}P_o - \rho_o g\vec{\nabla}z\right)\cdot\vec{n} > 0$

(2) $P_f = P_{atm}(t) + \rho_w g h_w(t)$.

where $F_L(t) < 0$, $P_{hyd}(t)$ and $\Psi_L(t)$ denote a *prescribed flux*, the *hydrostatic pressure* and a *prescribed head* for (parts of) the boundary $\Gamma_L(t)$ at time t, respectively.

Equation (2.3) is a first order ordinary differential equation in σ_z, so we need to impose only one boundary condition for σ_z on the top of the basin, i.e. $\sigma_z = P_{atm}(t) + \rho_w g h_w(t)$. Since there are no spatial derivatives involving the porosity ϕ in the system, it is not necessary to impose any boundary conditions for ϕ.

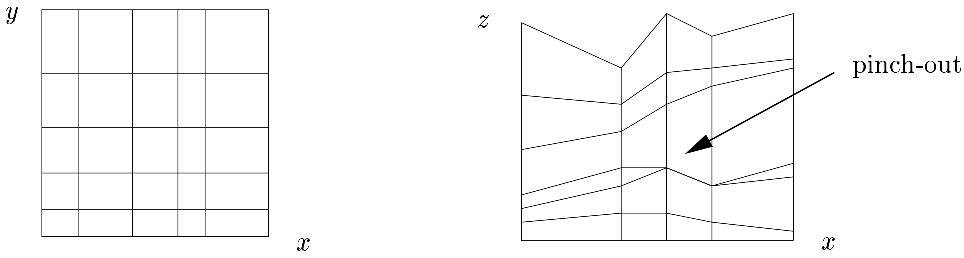

FIG. 1. *Typical horizontal (left) and vertical (right) cross sections of the mesh.*

2.2. Finite volume discretisation in space. In this section we will briefly describe the spatial discretisation of the conservation equations (2.1–2.3) using cell-centred finite volume schemes. In order to present this discretisation it is first of all necessary to specify the type of meshes we are going to use.

2.2.1. The 3D mesh (moving with velocity $\vec{V}_s(t)$). Given that the vertical projection of the considered domain $\Omega(t)$ is always rectangular and stays constant in time, we choose the mesh to be Cartesian in the horizontal plane (or xy-plane, see Figure 1, left). Vertically, the mesh is based on the lithology, so that each block (or cell) of our 3D mesh will correspond to a layer in the according chronostratigraphic column. In all our examples, the layers are considered to be continuous between two adjacent chronostratigraphic columns. Thus, the edges of the cells coincide on the interface between two columns, leading to conforming meshes. At each point in time, a cell is uniquely defined by the xy-coordinates of the centre of the corresponding chronostratigraphic column and by its four vertical edges.

It is however possible to model *"pinch-outs"*, i.e. layers that disappear in part of the domain (see Figure 1, right). In the meshes this is dealt with by including degenerate (or fictitious) cells of zero height which make it possible to work with a topologically Cartesian mesh in the xz-plane and the yz-plane as well. In other words, each cell has a unique neighbour in each direction, or – in the case of a pinch-out – no neighbour at all. Of course, during the discretisation process we will not include these degenerate elements.

We will denote the cells of the mesh by $\Omega_k := \Omega_k(t)$ with boundary $\partial\Omega_k$ and unit outward normal $\vec{n}_k := \vec{n}_k(t)$, for $k = 1, \ldots, N(t)$. The height of the cell is calculated as the arithmetic mean of the length of its four vertical edges and denoted by $h_k := h_k(t)$. Furthermore, for any physical quantity Y we denote by Y_k the value of Y at the centre of Ω_k.

REMARK 2.1. *Note that the number $N := N(t)$ of elements depends on the number of layers which have been sedimented at time t and will increase at the arrival of each new layer. However, it will also decrease, when cells degenerate through erosion.*

2.2.2. The mass conservation equations. To discretise equations (2.1) and (2.2) we let $\alpha \in \{w, o\}$ and integrate (2.1) and (2.2) respectively over each element Ω_k, $k = 1, \ldots, N$, of the mesh, i.e.

$$\int_{\Omega_k} \frac{\partial}{\partial t}\Big(\rho_\alpha \phi S_\alpha\Big)\, d\omega \; + \; \int_{\Omega_k} \mathrm{div}\, \Big(\rho_\alpha \phi S_\alpha \vec{V}_\alpha\Big)\, d\omega \; = \; \int_{\Omega_k} \rho_\alpha q_\alpha \, d\omega \; . \tag{2.11}$$

Now, using the given displacement velocity $\vec{V}_s$ at each point $\vec{x} \in \Omega_k$ and exchanging the order of integration and differentiation we can pass from the *partial* to the *full time derivative* in the first term of (2.11), i.e.

$$\int_{\Omega_k} \frac{\partial}{\partial t}(\rho_\alpha \phi S_\alpha) d\omega \;=\; \frac{d}{dt}\int_{\Omega_k} \rho_\alpha \phi S_\alpha \, d\omega \;-\; \int_{\Omega_k} \operatorname{div}(\rho_\alpha \phi S_\alpha \vec{V}_s) d\omega \;.$$

Substituting this into (2.11) and using Green's formula we obtain

$$\frac{d}{dt}\int_{\Omega_k} \rho_\alpha \phi S_\alpha \, d\omega \;+\; \int_{\partial\Omega_k} \rho_\alpha \phi S_\alpha (\vec{V}_\alpha - \vec{V}_s)\cdot \vec{n}_k d\gamma \;=\; \int_{\Omega_k} \rho_\alpha q_\alpha \, d\omega \;.$$

Finally using the generalised Darcy's law (2.5) (or (2.6) for the oil phase), we get

$$\frac{d}{dt}\int_{\Omega_k} \rho_\alpha \phi S_\alpha \, d\omega \;=\; \int_{\Omega_k} \rho_\alpha q \, d\omega \;+\; \sum_{\ell \in L_k} \int_{\Gamma_{k,\ell}} \rho_\alpha \eta K \left(\vec{\nabla} P_\alpha - \rho_\alpha g \vec{\nabla} z\right) \cdot \vec{n}_k d\gamma \;,$$

where L_k denotes the set of neighbouring cells Ω_ℓ of Ω_k and $\Gamma_{k,\ell}$ is the interface between Ω_ℓ and Ω_k.

To finish the spatial discretisation we need to apply quadrature rules to approximate the integrals leading to the following semidiscrete problem for each phase $\alpha = w, o$:

$$\frac{d}{dt} M_{\alpha,k} \;=\; Q_{\alpha,k} \;+\; \sum_{\ell \in L_k} F_{\alpha,\ell} \qquad \text{for} \;\; k = 1, \ldots, N \;. \tag{2.12}$$

Without going into the detail about the quadrature employed, note that the mass terms on the left hand side of (2.12) and the source terms on the right hand side have been approximated using the midpoint rule, i.e.

$$M_{\alpha,k} := |\Omega_k|\, \rho_{\alpha,k} \phi_k S_{\alpha,k} \qquad \text{and} \qquad Q_{\alpha,k} := |\Omega_k|\, \rho_{\alpha,k} q_{\alpha,k} \;, \tag{2.13}$$

containing only values of the parameters and of the unknown functions at the centre of the cell Ω_k (*cell-values*). Similarly, using (appropriate) averages of the cell-values on the faces $\Gamma_{k,\ell}$ of Ω_k and two-point finite difference approximations of the pressure gradients, the flux terms $F_{\alpha,\ell}$ in (2.12) have also been approximated using a midpoint rule on $\Gamma_{k,\ell}$. To ensure the stability of the discretisation we use an upstream approximation of the mobility, treating each phase separately.

2.2.3. The momentum equation. To discretise equation (2.3) on the other hand, let Ω_{k-1} be the cell situated above Ω_k in the same vertical column. We integrate (2.3) in z-direction between the cell centres of Ω_{k-1} and Ω_k, i.e.

$$\int_{z_{k-1}}^{z_k} \frac{\partial \sigma_z}{\partial z}\, dz = \int_{z_{k-1}}^{z_k} \Big(\phi \rho_f + (1-\phi)\rho_s\Big) g \, dz \tag{2.14}$$

The left hand side of (2.14) is equal to $\sigma_{z,k} - \sigma_{z,k-1}$. To approximate the right hand side of (2.14) on the other hand, we use the midpoint rule again to obtain the following discretisation of (2.3):

$$\sigma_{z,k} - \sigma_{z,k-1} = \tfrac{1}{2}\Big(W_k + W_{k-1}\Big) \qquad \text{for} \;\; k = 1, \ldots, N \;, \tag{2.15}$$

with

$$W_\ell := h_\ell \Big(\phi_\ell \rho_{f,\ell} + (1-\phi_\ell)\rho_s\Big) g, \qquad \text{for } \ell = k, k-1. \tag{2.16}$$

2.2.4. The rheology equation. Equation (2.4) does not contain any spatial derivatives, so to obtain a semidiscrete problem we can simply take (2.4) at the centre of each element Ω_k, i.e.

$$\frac{d\phi_k}{dt} = -\beta_k \frac{d\sigma_k}{dt} - \alpha_k \sigma_k \quad \text{with} \quad \sigma_k := \sigma_{z,k} - P_{f,k} \quad \text{for} \quad k = 1, \dots, N \; . \tag{2.17}$$

2.3. Time discretisation. In the previous section we have discretised equations (2.1–2.4) in space and by taking into account the boundary conditions and substituting $S_{w,k} = 1 - S_{o,k}$, we obtain a system of first order ordinary differential equations $(2.12)_w$, $(2.12)_o$, (2.17) coupled to a set of algebraic equations (2.15) for the unknown functions $\boldsymbol{s} := (S_{o,1}, \dots, S_{o,N})^T$, $\boldsymbol{p} := (P_{f,1}, \dots, P_{f,N})^T$, $\boldsymbol{\sigma} := (\sigma_{z,1}, \dots, \sigma_{z,N})^T$, and $\boldsymbol{\phi} := (\phi_1, \dots, \phi_N)^T$.

To fully discretise this system of equations we use implicit Euler in time. We define[3] a sequence of time steps $\Delta t^{(n)} > 0$ and times $t^{(n+1)} = t^{(n)} + \Delta t^{(n)}$, for $n = 0, 1, \dots$, and denote by $Y^{(n)}$ the value of Y at time $t^{(n)}$. Then we approximate the derivatives in (2.12) and (2.17) by simple finite differences between $t^{(n)}$ and $t^{(n+1)}$, and evaluate the remaining terms at $t^{(n+1)}$.

Starting with the known initial conditions at time $t^{(0)}$, we can thus iteratively calculate approximations for the unknown functions S_o, P_f, σ_z, and ϕ in (2.1–2.4) at time $t^{(n+1)}$, by solving the coupled system

$$\frac{M_{w,k}^{(n+1)} - M_{w,k}^{(n)}}{\Delta t^{(n)}} = Q_{w,k}^{(n+1)} \; + \; \sum_{\ell \in L_k} F_{w,\ell}^{(n+1)} \tag{2.18}$$

$$\frac{M_{o,k}^{(n+1)} - M_{o,k}^{(n)}}{\Delta t^{(n)}} = Q_{o,k}^{(n+1)} \; + \; \sum_{\ell \in L_k} F_{o,\ell}^{(n+1)} \tag{2.19}$$

$$\sigma_{z,k}^{(n+1)} - \sigma_{z,k-1}^{(n+1)} = \frac{1}{2} \left(W_k^{(n+1)} + W_{k-1}^{(n+1)} \right) \tag{2.20}$$

$$\frac{\phi_k^{(n+1)} - \phi_k^{(n)}}{\Delta t^{(n)}} = - \; \beta_k^{(n)} \frac{\sigma_k^{(n+1)} - \sigma_k^{(n)}}{\Delta t^{(n)}} \; - \; \alpha_k^{(n)} \sigma_k^{(n+1)} \tag{2.21}$$

of $4\, N^{(n+1)}$ non-linear equations for $\boldsymbol{s}^{(n+1)}$, $\boldsymbol{p}^{(n+1)}$, $\boldsymbol{\sigma}^{(n+1)}$, and $\boldsymbol{\phi}^{(n+1)}$.

To simplify the treatment of system (2.18–2.21) we have slightly relaxed the implicit treatment of σ_z and ϕ and evaluate the rheology parameters α_k and β_k in (2.21) at $t^{(n)}$ rather than at $t^{(n+1)}$. Furthermore, the assumption that the temperature T is known a priori at each point in time does not hold true in reality. On the contrary, it is calculated separately at each time step $t^{(n)}$. Therefore the parameters in (2.18–2.21) that depend on T are also evaluated at $T^{(n)}$ rather than at $T^{(n+1)}$.

REMARK 2.2. **(The IMPES scheme)** *In certain cases (e.g. low migration ratios) it is sufficient to discretise the saturations $S_{o,k}$ explicitly, thus leading to a much simpler system than (2.18–2.21) at each time step. However, the reduced stability of this discretisation usually results in a much larger number of necessary time steps. In the oil reservoir literature this method is called the IMPES scheme, i.e. implicit pressure explicit saturation.*

In this case, all the parameters in $F_{\alpha,\ell}$ and W_k involving $S_{o,k}$ are evaluated at $S_{o,k}^{(n)}$ instead of $S_{o,k}^{(n+1)}$. We can then eliminate the unknown saturations $S_{o,k}$ at time

[3]In reality the time steps $\Delta t^{(n)}$ will be chosen adaptively during the simulation in order to control the error.

$t^{(n+1)}$ from the system (2.18–2.21) altogether, by dividing each of the equations (2.18) and (2.19) by $\rho_{w,k}^{(n)}$ and $\rho_{o,k}^{(n)}$, respectively, and by adding up the resulting equations to obtain

$$\frac{|\Omega_k^{(n+1)}|\,\phi_k^{(n+1)} \;-\; |\Omega_k^{(n)}|\,\phi_k^{(n)}}{\Delta t^{(n)}} \;=\; \sum_{\alpha=w,o} \frac{1}{\rho_{\alpha,k}^{(n)}} \left(Q_{\alpha,k}^{(n)} \;+\; \sum_{\ell \in L_k} F_{\alpha,\ell}^{(n,n+1)} \right) \tag{2.22}$$

which does not depend on $\boldsymbol{s}^{(n+1)}$ anymore. Thus, the task at each time step reduces to solving a system (2.20–2.22) of $3\,N^{(n+1)}$ equations for $\boldsymbol{p}^{(n+1)}$, $\boldsymbol{\sigma}^{(n+1)}$, and $\boldsymbol{\phi}^{(n+1)}$.

In the following, we will not discuss this approach in detail, but we note that the preconditioners which are developed for the fully implicit case in Section 3 can be applied to this problem as well.

2.4. Newton linearisation and elimination of the porosity. We have seen that the numerical solution of (2.1–2.10) (with fully implicit time discretisation) reduces to solving at each time step $\Delta t^{(n)}$ a system of $4N \;:=\; 4\,N^{(n+1)}$ non-linear equations

$$\boldsymbol{\mathcal{F}}\left(\boldsymbol{y}^{(n+1)}\right) \;=\; 0 \tag{2.23}$$

for

$$\boldsymbol{y}^{(n+1)} := \left(P_{f,k}^{(n+1)}, \sigma_{z,k}^{(n+1)}, S_{o,k}^{(n+1)}, \phi_k^{(n+1)}\right)_{k=1,\dots,N}^T$$

where $\boldsymbol{\mathcal{F}} := \left(\boldsymbol{\mathcal{F}}_w^T, \boldsymbol{\mathcal{F}}_\sigma^T, \boldsymbol{\mathcal{F}}_o^T, \boldsymbol{\mathcal{F}}_\phi^T\right)^T$, $\boldsymbol{\mathcal{F}}_\alpha := \left(\mathcal{F}_{\alpha,k}\right)_{k=1,\dots,N}^T$ for $\alpha = w, \sigma, o, \phi$, and

$$\begin{aligned}
\mathcal{F}_{w,k} \;&:=\; \tfrac{1}{\rho_{w,k}^{(n)}}\left(M_{w,k}^{(n+1)} - M_{w,k}^{(n)}\right) \;-\; \tfrac{\Delta t^{(n+1)}}{\rho_{w,k}^{(n)}}\left(Q_{w,k}^{(n+1)} \;+\; \textstyle\sum_{\ell\in L_k} F_{w,\ell}^{(n+1)}\right),\\
\mathcal{F}_{\sigma,k} \;&:=\; \sigma_{z,k}^{(n+1)} - \sigma_{z,k-1}^{(n+1)} \;-\; \tfrac{1}{2}\left(W_k^{(n+1)} + W_{k-1}^{(n+1)}\right),\\
\mathcal{F}_{o,k} \;&:=\; \tfrac{1}{\rho_{o,k}^{(n)}}\left(M_{o,k}^{(n+1)} - M_{o,k}^{(n)}\right) \;-\; \tfrac{\Delta t^{(n+1)}}{\rho_{o,k}^{(n)}}\left(Q_{o,k}^{(n+1)} \;+\; \textstyle\sum_{\ell\in L_k} F_{o,\ell}^{(n+1)}\right),\\
\mathcal{F}_{\phi,k} \;&:=\; \phi_k^{(n+1)} - \phi_k^{(n)} \;+\; \beta_k^{(n)}\left(\sigma_k^{(n+1)} - \sigma_k^{(n)}\right) \;-\; \Delta t^{(n+1)}\alpha_k^{(n)}\sigma_k^{(n+1)}.
\end{aligned}$$

We solve system (2.23) using the classical Newton method, presented in Figure 2, with initial guess $\boldsymbol{y}^0 := \boldsymbol{y}^{(n)}$ and stopping criterion

$$\|\boldsymbol{\mathcal{F}}_w\|_\infty < \varepsilon_w\,, \quad \|\boldsymbol{\mathcal{F}}_\sigma\|_\infty < \varepsilon_\sigma\,, \quad \text{and} \quad \|\boldsymbol{\mathcal{F}}_o\|_\infty < \varepsilon_o\,, \tag{2.24}$$

leading in each Newton iteration to a system of linear equations

$$\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{y}}(\boldsymbol{y}^m)\;\boldsymbol{\delta y}^m = -\boldsymbol{\mathcal{F}}(\boldsymbol{y}^m). \tag{2.25}$$

Finally, we observe that $\partial \boldsymbol{\mathcal{F}}_\phi / \partial \boldsymbol{\phi} = I$ and that we can therefore eliminate the vector $\boldsymbol{\delta\phi}^{m+1}$, corresponding to the porosity from system (2.25). Thus, the computational core of the forward simulation is the solution of a $3N \times 3N$ system of linear equations of the form

$$A\,\boldsymbol{\delta y} \;:=\; \begin{pmatrix} A^{w,p} & A^{w,\sigma} & A^{w,s} \\ A^{\sigma,p} & A^{\sigma,\sigma} & 0 \\ A^{o,p} & A^{o,\sigma} & A^{o,s} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta p} \\ \boldsymbol{\delta\sigma} \\ \boldsymbol{\delta s} \end{pmatrix} = \begin{pmatrix} \mathbf{f}^w \\ \mathbf{f}^\sigma \\ \mathbf{f}^o \end{pmatrix} \;=:\; \mathbf{f} \tag{2.26}$$

```
Let y^0 := y^(n).
For m = 0, 1, ... until convergence
  Solve
        ∂F/∂y (y^m) δy^m = −F(y^m)
  Set y^{m+1} = y^m + δy^m
End For.
Set y^(n+1) := y^{m+1}.
```

Let $\boldsymbol{y}^0 := \boldsymbol{y}^{(n)}$.
For $m = 0, 1, \ldots$ until convergence

Solve
$$\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{y}}(\boldsymbol{y}^m)\, \boldsymbol{\delta y}^m = -\boldsymbol{\mathcal{F}}(\boldsymbol{y}^m)$$

Set $\boldsymbol{y}^{m+1} = \boldsymbol{y}^m + \boldsymbol{\delta y}^m$

End For.
Set $\boldsymbol{y}^{(n+1)} := \boldsymbol{y}^{m+1}$.

FIG. 2. *The classical Newton method*

in each Newton iteration and for each time step $\Delta t^{(n)}$, where

$$A^{\alpha,\beta} := \left(\frac{\partial \boldsymbol{\mathcal{F}}_\alpha}{\partial \beta} - \frac{\partial \boldsymbol{\mathcal{F}}_\alpha}{\partial \phi}\frac{\partial \boldsymbol{\mathcal{F}}_\phi}{\partial \beta}\right)(\boldsymbol{y}^m) \qquad \text{for} \quad \alpha = w, \sigma, o \quad \text{and} \quad \beta = p, \sigma, s \quad \text{and}$$
$$\mathbf{f}^\alpha := \left(-\boldsymbol{\mathcal{F}}_\alpha + \frac{\partial \boldsymbol{\mathcal{F}}_\alpha}{\partial \phi}\boldsymbol{\mathcal{F}}_\phi\right)(\boldsymbol{y}^m) \qquad \text{for} \quad \alpha = w, \sigma, o.$$

System (2.26) is very large, strongly non-symmetric and very ill-conditioned. For an efficient and robust iterative solution of this system, it is therefore crucial to find good preconditioners. The remainder of this paper will be devoted to the construction of such preconditioners and to the evaluation of their robustness and efficiency for representative test cases from real case studies.

3. Preconditioning strategy. The rate of convergence for almost all iterative methods depends largely on the condition number of the matrix A in (2.26). Preconditioning techniques are one of the main keys to a better performance. A preconditioning operator $\mathcal{P} = \mathcal{P}_\mathcal{L}\mathcal{P}_\mathcal{R}$ for a matrix A is chosen in such a way that the preconditioning systems $\mathcal{P}_\mathcal{L}\, \boldsymbol{w} = \boldsymbol{d}$ and $\mathcal{P}_\mathcal{R}\, \boldsymbol{w} = \boldsymbol{d}$ can be solved quickly and that the condition of the new (preconditioned) iteration matrix $\mathcal{P}_\mathcal{L}^{-1} A \mathcal{P}_\mathcal{R}^{-1}$ is significantly better than the condition number of A. The underlying system (2.26) is then transformed as follows:

$$A\, \boldsymbol{\delta y} = \mathbf{f} \quad \longmapsto \quad \tilde{A}\, \boldsymbol{\delta\tilde{y}} := (\mathcal{P}_\mathcal{L}^{-1} A \mathcal{P}_\mathcal{R}^{-1})\, (\mathcal{P}_\mathcal{R} \boldsymbol{\delta y}) = (\mathcal{P}_\mathcal{L}^{-1}\mathbf{f}) =: \tilde{\mathbf{f}}, \tag{3.1}$$

and the iterative method (e.g. a Krylov method like Bi-CGStab or GMRES) is applied to the transformed (or *preconditioned*) system

$$\tilde{A}\, \boldsymbol{\delta\tilde{y}} = \tilde{\mathbf{f}} \tag{3.2}$$

If $\mathcal{P}_\mathcal{R}$ or $\mathcal{P}_\mathcal{L}$ are chosen to be equal to the identity I, we speak of *left or right preconditioning*, respectively.

The idea for the preconditioning strategy which we are going to present in this section stems from the related field of oil reservoir simulation. In reservoir simulations, the geological time spans that are simulated are much shorter in comparison to the time spans considered here (i.e. years instead of million years), so that the porosity ϕ and the effective stress $\sigma = \sigma_z - P_f$ can be kept constant, thus leading to systems similar to (2.26), but in pressure and saturation only. In Behie & Vinsome [2], Edwards [3], and Lacroix, Vassilevski & Wheeler [5] a preconditioning strategy has been developed for these problems decoupling the treatment of pressure and saturation. We will use a similar strategy for (2.26).

3.1. Motivation. In order to have a proper foundation for the presented preconditioners, we will first discuss some important features of the underlying physical problems and deduce some assumptions for (2.26).

First of all we observe that the porous media which form a typical sedimentary basin can be considered to be very rigid, so that the rheology parameters $\beta_k^{(n)}$ and $\alpha_k^{(n)}$ are in general small. Therefore, the derivative

$$\left|\frac{\partial \mathcal{F}_{\phi,k}}{\partial \sigma_k}\right| = \left|\beta_k^{(n)} - \Delta t^{(n)} \alpha_k^{(n)}\right|$$

which plays the role of the compressibility of the rock is in general small. Applied to the linear system (2.26) this allows us to make

ASSUMPTION 3.1. *The contributions from the rheology equations to the first and the third block row in (2.26) are small compared to the contributions from the mass conservation equations, i.e.*

$$[A^{\alpha,p}]_{kk} \approx \frac{\partial \mathcal{F}_{\alpha,k}}{\partial P_{f,k}} \qquad \text{and} \qquad \left|[A^{\alpha,\sigma}]_{kk}\right| \ll \left|[A^{\alpha,p}]_{kk}\right| \quad \text{for} \quad \alpha = w, o.$$

A consequence of this assumption is that the blocks $A^{w,p}$ and $A^{o,p}$ are in principle discrete diffusion operators and thus close to elliptic. Furthermore, this assumption also allows us to neglect the blocks $A^{w,\sigma}$ and $A^{o,\sigma}$ in the preconditioner, thus decoupling the first and the third block row of (2.26) from the second.

Secondly, for fixed porosity ϕ and effective stress σ, the continuous problem (2.1–2.10) reduces to a system of two partial differential equations in pressure and saturation only, modelling two-phase Darcy flow. The analysis in Gagneux & Madaune-Taut [4, Section 1.3.2] shows that this system is essentially

(3.3) *elliptic* in P_f and *hyperbolic* or *transport dominated parabolic* in S_o

(see also the remark in Lacroix et al. [5, Section 2.2]). As a consequence P_f has to be treated implicitly while S_o *may* be treated explicitly (the IMPES scheme discussed in Remark 2.2). The successful application of the IMPES model in many situations implies that the influence of pressure on saturation is much more important than vice versa and allows us to make the following

ASSUMPTION 3.2. *The pressure block* $(A^{w,p} + A^{o,p})$ *obtained by taking the sum of the first and the third block row in (2.27) (as in Remark 2.2 for the IMPES scheme) is a good approximation for the Schur complement* $\left(A^{w,p} - A^{w,s}\,(A^{o,s})^{-1}\,A^{o,p}\right)$, *or in other words*

$$A^{o,s} \approx -A^{w,s}.$$

However neglecting the influence of saturation on pressure altogether would be too strong an assumption. Therefore using (3.3) again, we assume that this influence will mainly be local and make the following

ASSUMPTION 3.3. *A local decoupling of the first and the third block row in (2.26) will also result in a reasonable global decoupling.*

A further consequence of (3.3) is that simple pointwise relaxation schemes, like Jacobi or Gauss-Seidel, are in general sufficient to precondition the blocks $A^{w,s}$ and $A^{o,s}$.

Our preconditioning strategy for (2.26) will now consist of three stages:

(i) local decoupling of the two mass conservation equations on each cell Ω_k (or in other words, choice of a pressure equation on Ω_k);
(ii) preconditioning of the pressure unknowns;
(iii) updating of the remaining unknowns (Block Gauss-Seidel), or if this is not sufficient, feedback-coupling with the pressure equation (two-stage preconditioners).

We will discuss them in the following three sections.

3.2. Local decoupling – choice of the pressure equation. Let $k = 1, \dots, N$. The rows in (2.26) corresponding to the mass conservation equations (2.18) and (2.19) for water and oil on Ω_k can be written like

$$[A^{w,p}]_{kk}\,\delta p_k \;+\; [A^{w,s}]_{kk}\,\delta s_k \;+\; [A^{w,\sigma}]_{kk}\,\delta\sigma_k + \sum_{\ell\in L_k}\;\sum_{\beta=p,s,\sigma}[A^{w,\beta}]_{k\ell}\,\delta\beta_k \;=\; \mathrm{f}_k^w \tag{3.4}$$

$$[A^{o,p}]_{kk}\,\delta p_k \;+\; [A^{o,s}]_{kk}\,\delta s_k \;+\; [A^{o,\sigma}]_{kk}\,\delta\sigma_k + \sum_{\ell\in L_k}\;\sum_{\beta=p,s,\sigma}[A^{o,\beta}]_{k\ell}\,\delta\beta_k \;=\; \mathrm{f}_k^o \tag{3.5}$$

Similar to the approach in [5], we will now aim at taking linear combinations of (3.4) and (3.5) such that the first equation (in the following called the *pressure equation*) does not depend on δs_k anymore. As a consequence of Assumption 3.3, this should significantly reduce the dependency of the pressure equation on the saturations.

To formalise this *local decoupling* let us define

$$A_k := \begin{pmatrix} [A^{w,p}]_{kk} & [A^{w,s}]_{kk} \\ [A^{o,p}]_{kk} & [A^{o,s}]_{kk} \end{pmatrix} \qquad \text{and} \qquad G_k := \begin{pmatrix} \gamma_k^{p,w} & \gamma_k^{p,o} \\ \gamma_k^{s,w} & \gamma_k^{s,o} \end{pmatrix}.$$

We want to choose G_k such that

$$\gamma_k^{p,w}\,[A^{w,s}]_{kk} \;+\; \gamma_k^{p,o}\,[A^{o,s}]_{kk} \;=\; 0 \tag{3.6}$$

and therefore

$$G_k A_k \;=\; \begin{pmatrix} [\mathcal{A}^p]_{kk} & 0 \\ [\mathcal{A}^{s,p}]_{kk} & [\mathcal{A}^s]_{kk} \end{pmatrix}$$

so that we can locally decouple equations (3.4) and (3.5) by multiplying them with G_k.

Globally, this corresponds to multiplying system (2.26) by

$$G := \begin{pmatrix} \mathrm{Diag}(\gamma_k^{p,w}) & 0 & \mathrm{Diag}(\gamma_k^{p,o}) \\ 0 & I & 0 \\ \mathrm{Diag}(\gamma_k^{s,w}) & 0 & \mathrm{Diag}(\gamma_k^{s,o}) \end{pmatrix} \tag{3.7}$$

and leads to the transformed system

$$\mathcal{A}\,\boldsymbol{\delta y} \;:=\; \begin{pmatrix} \mathcal{A}^p & \mathcal{A}^{p,\sigma} & \mathcal{A}^{p,s} \\ \mathcal{A}^{\sigma,p} & \mathcal{A}^{\sigma} & 0 \\ \mathcal{A}^{s,p} & \mathcal{A}^{s,\sigma} & \mathcal{A}^s \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta p} \\ \boldsymbol{\delta\sigma} \\ \boldsymbol{\delta s} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}^p \\ \boldsymbol{f}^\sigma \\ \boldsymbol{f}^s \end{pmatrix} \;=:\; \boldsymbol{f} \tag{3.8}$$

with

$$\mathcal{A} \; := \; GA \qquad \text{and} \qquad \boldsymbol{f} \; := \; G\mathbf{f} \; .$$

Thus, the left preconditioner in (3.1) is chosen to be the *decoupling preconditioner*

$$\mathcal{P}_{\mathcal{L}}^{-1} \; := \; G.$$

Obviously, infinitely many different choices for G_k are possible which satisfy (3.6). However, we would like

(a) the action of G to be cheap,

(b) the block

$$\mathcal{A}^p \; := \; \mathrm{Diag}(\gamma_k^{p,w})\, A^{w,p} \; + \; \mathrm{Diag}(\gamma_k^{p,o})\, A^{o,p}$$

to preserve the assumed elliptic nature of $A^{w,p}$ and $A^{o,p}$,

(c) the norm of G to be $O(1)$.

This final criterion is particularly important, if we think of applying inexact Newton methods instead of classical Newton, since the stopping criterion for the linear solve in that case depends on the norm of the residual. An example for a choice of G_k which does in general not satisfy (c) is $G_k := A_k^{-1}$, and we will include some numerical results with this decoupling in Section 4.2.1 to show the consequences.

We also note that it is not easy to verify Criterion (b) explicitly, but the numerical tests in Section 4.2 show that all the choices for G_k presented below lead to elliptic pressure blocks $\mathcal{A}^p$.

To finish this section we present the three choices for G_k which we will use:

3.2.1. Householder reflection. Here, the matrix G_k is chosen to be the Householder reflector matrix

$$G_k^{\mathcal{H}} := I - 2\boldsymbol{w}_k \boldsymbol{w}_k^T \tag{3.9}$$

(see for example Saad [8, Section 1.7]) where $\boldsymbol{w}_k$ is selected such that

$$G_k^{\mathcal{H}} \boldsymbol{a}_k^s \; = \; \alpha\, \boldsymbol{e}_2 \quad \text{with} \quad \boldsymbol{a}_k^s := \Big([A^{w,s}]_{kk}, [A^{o,s}]_{kk} \Big)^T \quad \text{and} \quad \alpha := -\mathrm{sign}\Big([A^{o,s}]_{kk} \Big) \|\boldsymbol{a}_k^s\|_2$$

and thus (3.6) is satisfied.

This method conserves the $l2$-norm of the residual exactly, i.e. Criterion (c) is satisfied with $\|G\|_2 \; = \; 1$. However, it is rather expensive (compared to the other choices) and leads to an unnecessary fill-in in the saturation blocks.

3.2.2. Gauss elimination. The second choice uses Gauss elimination. We take

$$G_k^{\mathcal{G}} := \begin{pmatrix} 1 & -\frac{[A^{w,s}]_{kk}}{[A^{o,s}]_{kk}} \\ 0 & 1 \end{pmatrix} \;\text{ if }\; [A^{o,s}]_{kk} \neq 0, \qquad G_k^{\mathcal{G}} := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \;\text{ otherwise,} \tag{3.10}$$

which obviously satisfies (3.6) as well.

This method does not change the second equation and is in general much cheaper than Householder decoupling. Furthermore, as a consequence of Assumption 3.2,

$$-\frac{[A^{w,s}]_{kk}}{[A^{o,s}]_{kk}} \; \approx \; 1, \tag{3.11}$$

and thus Criterion (c) is also satisfied with $\|G\|_2 \approx \sqrt{\frac{3+\sqrt{5}}{2}} \approx 1.618$.

3.2.3. Quasi–IMPES decoupling. In this final choice we take

$$G_k^Q := \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{3.12}$$

which does not satisfy (3.6) exactly, but because of (3.11) we have at least that in general

$$\left| [\mathcal{A}^{p,s}]_{kk} \right| \;:=\; \left| [A^{w,s}]_{kk} + [A^{o,s}]_{kk} \right| \;\ll\; \max \left\{ \left| [A^{w,s}]_{kk} \right| , \left| [A^{o,s}]_{kk} \right| \right\} .$$

In fact, the linear combination used here to construct the pressure equation is exactly the same as the one used in Remark 2.2 to construct the IMPES equation (2.22) with the only difference that the flux terms $F_{\alpha,\ell}$ depend on the saturation at $t^{(n+1)}$. Therefore, this Quasi–IMPES decoupling eliminates the contributions to $[\mathcal{A}^{p,s}]_{kk}$ coming from the mass term $M_{\alpha,k}$, but not the ones from the flux terms $F_{\alpha,\ell}$. However, because of Assumption 3.2 this still leads to a significant reduction of the dependency of the pressure equation on the saturations.

With respect to cost, it is even cheaper than Gauss elimination while not changing the second equation either, and Criterion (c) is satisfied with $\|G\|_2 = \sqrt{\frac{3+\sqrt{5}}{2}}$.

3.3. Pressure preconditioner. As a consequence of Assumption 3.1 and of Criterion (b) in Section 3.2 we know that the *pressure block* $\mathcal{A}^p$ in the decoupled system (3.8) is elliptic, or at least close to elliptic, even though it is non-symmetric. In fact, the principle part of $\mathcal{A}^p$ corresponds to a discrete diffusion operator with a strongly heterogeneous and anisotropic diffusivity tensor, and thus the matrix requires a strong preconditioning. It is especially important to take care of the global error components which are present in the solution, and which can not be dealt with by simple relaxation schemes, like Jacobi or Gauss-Seidel, or by simple incomplete factorisation methods, like ILU(0). One way to overcome this problem is to employ multilevel preconditioners. However, simple geometric coarsening techniques are not robust with respect to anisotropies and heterogeneities and therefore we resort to *Algebraic Multigrid (AMG)* methods to precondition $\mathcal{A}^p$.

More precisely, we use the AMG code `AMG1R5` (Release 1.5, 1990) by Ruge and Stüben [7, 10]. The smoother on each "grid" is standard Gauss-Seidel, but the crucial feature of the method is the particular way of choosing the coarse grid cells and the grid transfer operators based on the matrix entries. For details we refer to [7] and [10]. As we will see in Section 4.2, it is sufficient to employ just one V-cycle of this multilevel method for the preconditioning of the pressure block $\mathcal{A}^p$ in (3.8), in the following referred to as $\mathcal{P}^p$

3.4. "Recoupling" – the global preconditioning method. Let us now construct a right preconditioner $\mathcal{P}_{\mathcal{R}}$ for the decoupled system (3.8) which will take care of the "recoupling" of the unknowns.

As a consequence of Assumption 3.1, the block $\mathcal{A}^{p,\sigma}$ is "small" compared to $\mathcal{A}^p$ and can thus in a first instance be neglected. Similarly, as a consequence of Assumption 3.2 and 3.3 and of condition (3.6), the block $\mathcal{A}^{p,s}$ is "small" compared to $\mathcal{A}^p$ and can also in a first instance be neglected. This leads to a block triangular system and to the idea for the following *Block Gauss-Seidel (BGS)* preconditioner:

$$\mathcal{P}_{BGS}^{-1} := \begin{pmatrix} \mathcal{P}^p & 0 & 0 \\ \mathcal{A}^{\sigma,p} & \mathcal{A}^\sigma & 0 \\ \mathcal{A}^{s,p} & \mathcal{A}^{s,\sigma} & \mathcal{P}^s \end{pmatrix}^{-1} \tag{3.13}$$

where $\mathcal{P}^p$ denotes the AMG preconditioner for $\mathcal{A}^p$ presented in Section 3.3 and $\mathcal{P}^s$ is a preconditioner for the saturation block $\mathcal{A}^s$. As mentioned above, it is sufficient to use a simple relaxation scheme, like Jacobi or Gauss-Seidel, for $\mathcal{P}^s$, because the dependency on saturation is hyperbolic or transport dominated parabolic. In Section 4 we will use one iteration of Gauss-Seidel.

Thus, to use the preconditioner practically, i.e. to apply $\mathcal{P}_{BGS}^{-1}$, it only remains to discuss how to invert $\mathcal{A}^\sigma := A^{\sigma,\sigma}$. It follows from the definition of $A^{\sigma,\sigma}$ that

$$[A^{\sigma,\sigma}]_{k\ell} \;=\; 0 \qquad \text{for all} \;\; \ell \notin \{k, k-1\}$$

and so the matrix is bidiagonal and can be inverted by simple forward substitutions.

REMARK 3.4. *Note that the particular choice of $\mathcal{P}^s$ (i.e. one iteration of Gauss-Seidel which is equivalent to a simple forward substitution) makes it possible to implement the block Gauss-Seidel preconditioner $\mathcal{P}_{BGS}^{-1}$ using a 2×2 block structure*

$$\mathcal{P}_{BGS}^{-1} := \begin{pmatrix} \mathcal{P}^p & 0 \\ \mathcal{L} & \mathcal{B} \end{pmatrix}^{-1} \qquad \textit{with} \quad \mathcal{B} := \begin{pmatrix} \mathcal{A}^\sigma & 0 \\ \mathcal{A}^{s,\sigma} & \mathcal{P}^s \end{pmatrix} \quad \textit{and} \quad \mathcal{L} := \begin{pmatrix} \mathcal{A}^{\sigma,p} \\ \mathcal{A}^{s,p} \end{pmatrix}$$

and applying one iteration of Gauss-Seidel to the $2N \times 2N$ block $\mathcal{B}$. This approach is algebraically equivalent, but leads to a better performance in terms of CPU-time.

If the assumptions in Section 3.1 are too strong, it might not be justified to simply neglect $\mathcal{A}^{p,\sigma}$ and $\mathcal{A}^{p,s}$. It is necessary in this case that the preconditioner also provides some feedback from the second and the third block row to the first. A way of providing such a feedback-coupling is to combine the Block Gauss-Seidel preconditioner $\mathcal{P}_{BGS}$ in a multiplicative way with a second preconditioner $\widetilde{\mathcal{P}}$ that provides (at least locally) such a coupling and that is cheap to invert. A good candidate for $\widetilde{\mathcal{P}}$ is *zero fill-in incomplete LU factorisation ILU(0)* of $\mathcal{A}$. It is cheap and provides a good local coupling of the different physical unknowns. The combined preconditioner can then be written as

$$\mathcal{P}_{C2S}^{-1} \;:=\; \Big(I \;-\; \widetilde{\mathcal{P}}^{-1}\,(\mathcal{A} \;-\; \mathcal{P}_{BGS})\Big)\,\mathcal{P}_{BGS}^{-1}\,. \tag{3.14}$$

Similar techniques in reservoir simulations, notably in [2] and [5], are usually referred to as *combinative two-stage preconditioners (C2S)* and thus we will also adopt this name.

The right preconditioner in (3.1) is now chosen to be either

$$\mathcal{P}_{\mathcal{R}}^{-1} \;:=\; \mathcal{P}_{BGS}^{-1} \qquad \text{or} \qquad \mathcal{P}_{\mathcal{R}}^{-1} \;:=\; \mathcal{P}_{C2S}^{-1}$$

which completes our discussion of the preconditioning strategy.

4. Numerical Results. In this section we will discuss the results of the numerical tests which we effectuated to evaluate the performance of the preconditioners in terms of efficiency and robustness on some test cases from real case studies. As the iterative solution method for the preconditioned systems (3.2) we use the Bi-CGStab method by Van der Vorst [11] which is particularly suited for nonsymmetric linear systems. To judge the efficiency and the robustness of the preconditioners we will regard the number of iterations of preconditioned Bi-CGStab that are necessary to reduce the residual by a factor of $\varepsilon = 10^{-6}$ and the CPU-time elapsed in the solver on an SGI Octane. We will compare the results obtained with our preconditioners with the results using ILU(0) preconditioning (which is used at the moment in `TEMIS3D`). The implementation of the preconditioners was carried out within the `PETSc` library of the Argonne National Laboratory, IL [1].

4.1. Description of the test cases. The four test cases are taken from real case studies of major oil companies. These studies are confidential and so we will refer to the test cases as Problems **A** – **D**.

They comprise models with strong heterogeneities and anisotropies in the porous media as well as models with high migration ratios. In our tests we will only regard the final period, when the last layer is deposited and thus when all z-layers are present. The only exception is Problem **B**, where an intermediate period is simulated. The number of time steps, the average step size, and the average number of Newton iterations per time step as well as the number of elements in the mesh will depend on the actual test case and are listed in Table 1.

TABLE 1
Technical details for Problems **A** – **D**

Problem	Time steps	Av. step size	Newton steps	Elements	Unknowns
A	12	0.11 Ma	2.0	5649	16947
B	33	0.01 Ma	6.6	7901	23703
C	11	0.18 Ma	2.5	23104	69312
D	27	0.39 Ma	2.0	32199	96597

Problems **A** and **D** are problems with strong heterogeneities and anisotropies in the permeability tensors of the porous media but with very low migration ratios, and thus represent almost single-phase flow. Problems **B** and **C** on the other hand are problems with high migration ratios, and thus much stronger changes in the saturations. In Table 1 they are ordered according to mesh size.

4.2. Testing and comparison of the different preconditioners. We will now test the preconditioning strategy on Problems **A** – **D** by employing the three different decoupling preconditioners $G^{\mathcal{Q}}$ (Quasi-IMPES), $G^{\mathcal{G}}$ (Gauss), and $G^{\mathcal{H}}$ (Householder) of Section 3.2 together with the two different recoupling preconditioners $\mathcal{P}_{BGS}^{-1}$ (Block Gauss-Seidel) and $\mathcal{P}_{C2S}^{-1}$ (Combinative Two-Stage preconditioner) of Section 3.4 and by comparing the results with ILU(0) (left-)preconditioning. The average numbers of iterations of preconditioned Bi-CGStab that are necessary to reduce the residual by a factor of $\varepsilon = 10^{-6}$ are listed in Table 2. The CPU-times elapsed on average in the solver are listed in Table 3.

In all cases the pressure preconditioner $\mathcal{P}^p$ in (3.13) has been chosen to be one V-cycle of AMG as presented in Section 3.3, and the preconditioner $\mathcal{P}^s$ for the saturation block was one iteration of Gauss-Seidel. The second preconditioner $\widetilde{\mathcal{P}}$ in the combinative two-stage technique (3.14) was ILU(0).

We note in Table 2 that the performance of the ILU(0) preconditioner (which is used at the moment in `TEMIS3D`) depends strongly on the heterogeneities and the anisotropies in the problem, and thus the number of iterations of Bi-CGStab is very high in Problems **A** and **D**. We observe also a dependency on the number of unknowns (47.8 in Problem **A** against 101.8 in Problem **D**). However, ILU(0) seems to be better suited to problems with high migration ratios and moderate heterogeneities (i.e. 20.1 iterations in Problem **B** and 21.0 in Problem **C**) where the local coupling of pressure and saturation is more important.

The block Gauss-Seidel preconditioner (Table 2, 3rd column), on the other hand, seems to be extremely well suited to heterogeneous and anisotropic problems, where a strong preconditioning of the pressure part is crucial and the coupling between pressure and saturation is less important (regardless of the decoupling method, see

TABLE 2
Comparison of different preconditioners in (3.1) – Bi-CGStab iterations.

Problem **A** ($N = 16947$)

$\mathcal{P}_{\mathcal{L}}^{-1} \backslash \mathcal{P}_{\mathcal{R}}^{-1}$	I	$\mathcal{P}_{BGS}^{-1}$	$\mathcal{P}_{C2S}^{-1}$
ILU(0)	47.8	-	-
$G^{\mathcal{Q}}$	-	4.2	3.5
$G^{\mathcal{G}}$	-	4.2	3.5
$G^{\mathcal{H}}$	-	4.1	3.5

Problem **B** ($N = 23703$)

$\mathcal{P}_{\mathcal{L}}^{-1} \backslash \mathcal{P}_{\mathcal{R}}^{-1}$	I	$\mathcal{P}_{BGS}^{-1}$	$\mathcal{P}_{C2S}^{-1}$
ILU(0)	20.1	-	-
$G^{\mathcal{Q}}$	-	10.2	3.8
$G^{\mathcal{G}}$	-	9.4	4.6
$G^{\mathcal{H}}$	-	9.6	4.6

Problem **C** ($N = 69312$)

$\mathcal{P}_{\mathcal{L}}^{-1} \backslash \mathcal{P}_{\mathcal{R}}^{-1}$	I	$\mathcal{P}_{BGS}^{-1}$	$\mathcal{P}_{C2S}^{-1}$
ILU(0)	21.0	-	-
$G^{\mathcal{Q}}$	-	13.6	6.0
$G^{\mathcal{G}}$	-	13.0	2.8
$G^{\mathcal{H}}$	-	5.3	2.9

Problem **D** ($N = 96597$)

$\mathcal{P}_{\mathcal{L}}^{-1} \backslash \mathcal{P}_{\mathcal{R}}^{-1}$	I	$\mathcal{P}_{BGS}^{-1}$	$\mathcal{P}_{C2S}^{-1}$
ILU(0)	101.8	-	-
$G^{\mathcal{Q}}$	-	3.9	3.1
$G^{\mathcal{G}}$	-	3.8	3.2
$G^{\mathcal{H}}$	-	3.3[a]	3.2

[a] 2.7 Newton steps (instead of 2.0)!

the results for Problems **A** and **D** in Table 2). However, we also note that for problems with strong migration ratios (i.e. Problems **B** and **C**), Assumption 3.2 is apparently too strong. The coupling between saturation and pressure in the Block Gauss-Seidel preconditioner (3.13) which simply neglects the block $\mathcal{A}^{p,s}$ is not sufficient (about 10 or more iterations of Bi-CGStab in Problems **B** and **C** against about 4 iterations in Problems **A** and **D**).

The combinative two-stage preconditioner (Table 2, 4th column) combines very well the advantages of the two other preconditioners and thus leads to a very robust preconditioning of the linear system in all situations and regardless of the decoupling method. It is robust with respect to heterogeneities and anisotropies, with respect to high migration ratios, with respect to mesh size and also with respect to the time step size. For example using Gauss decoupling (Table 2, 4th row, 4th column), the average number of iterations of Bi-CGStab in Problems **A**, **B**, **C** and **D**, is 3.5, 4.6, 2.8, and 3.2 respectively.

The differences in the numbers of iterations between the three decoupling techniques are very small (see Table 2). Only Problem **C** shows a dependency on the chosen technique (which is rather arbitrary though and hard to explain). However, we note that the application of the Householder decoupling $G^{\mathcal{H}}$ is about 1.5–2 times more expensive than the two other techniques which is also reflected in the CPU-times in Table 3. Measurements in our test runs showed that in general the decoupling makes up about 10% of the CPU-time elapsed in the solver for Quasi-IMPES $G^{\mathcal{Q}}$ and Gauss decoupling $G^{\mathcal{G}}$ and about 15–20% for Householder decoupling $G^{\mathcal{H}}$.

With respect to CPU-time (Table 3) the advantage of the new preconditioning strategy is less impressive because of its increased cost in the setup phase as well as in its application (the most expensive part being the setup for the AMG pressure preconditioner). Nevertheless, the CPU-times for the best combinations of decoupling and recoupling are better than the CPU-times with ILU(0) in all test cases. In Problem **D**, for example, Bi-CGStab is on average more than 4.3 times faster when

TABLE 3
Comparison of different preconditioners in (3.1) – CPU-time elapsed in solver.

Problem **A** ($N = 16947$)

$\mathcal{P}_{\mathcal{L}}^{-1} \backslash \mathcal{P}_{\mathcal{R}}^{-1}$	I	$\mathcal{P}_{BGS}^{-1}$	$\mathcal{P}_{C2S}^{-1}$
ILU(0)	1.41 s	-	-
$G^{\mathcal{Q}}$	-	0.47 s	0.57 s
$G^{\mathcal{G}}$	-	0.49 s	0.58 s
$G^{\mathcal{H}}$	-	0.65 s	0.76 s

Problem **B** ($N = 23703$)

$\mathcal{P}_{\mathcal{L}}^{-1} \backslash \mathcal{P}_{\mathcal{R}}^{-1}$	I	$\mathcal{P}_{BGS}^{-1}$	$\mathcal{P}_{C2S}^{-1}$
ILU(0)	1.18 s	-	-
$G^{\mathcal{Q}}$	-	1.41 s	1.08 s
$G^{\mathcal{G}}$	-	1.32 s	1.19 s
$G^{\mathcal{H}}$	-	1.67 s	1.45 s

Problem **C** ($N = 69312$)

$\mathcal{P}_{\mathcal{L}}^{-1} \backslash \mathcal{P}_{\mathcal{R}}^{-1}$	I	$\mathcal{P}_{BGS}^{-1}$	$\mathcal{P}_{C2S}^{-1}$
ILU(0)	3.9 s	-	-
$G^{\mathcal{Q}}$	-	5.4 s	4.2 s
$G^{\mathcal{G}}$	-	5.1 s	2.7 s
$G^{\mathcal{H}}$	-	3.5 s	3.5 s

Problem **D** ($N = 96597$)

$\mathcal{P}_{\mathcal{L}}^{-1} \backslash \mathcal{P}_{\mathcal{R}}^{-1}$	I	$\mathcal{P}_{BGS}^{-1}$	$\mathcal{P}_{C2S}^{-1}$
ILU(0)	22.9 s	-	-
$G^{\mathcal{Q}}$	-	5.4 s	5.9 s
$G^{\mathcal{G}}$	-	5.3 s	5.9 s
$G^{\mathcal{H}}$	-	5.8 s[a]	7.0 s

[a]2.7 Newton steps (instead of 2.0)!

applied to a system preconditioned with $\mathcal{P}_{\mathcal{L}}^{-1} := G^{\mathcal{G}}$ and $\mathcal{P}_{\mathcal{R}}^{-1} := \mathcal{P}_{BGS}^{-1}$ than applied to an ILU(0)-preconditioned system. Since the new preconditioning strategy does not depend on the mesh size it can be expected that this speedup with respect to ILU(0) will even be more impressive on larger problems particularly in the presence of heterogeneities and anisotropies in the porous media.

4.2.1. Decoupling using exact inversion of the local block. An example for a decoupling preconditioner which does not satisfy Criterion (c) in Section 3.2 is the *full decoupling*

$$G_k^{\mathcal{F}} := A_k^{-1}. \tag{4.1}$$

It changes the scaling between the equations in the different cells by many orders of magnitude (especially in the case of strong heterogeneities) and thus the norm of the residual changes as well. This will effect the stopping criterion for the iterative solver of the linear system and can lead to "oversolving" (i.e. unnecessary iterations of the iterative method, when the approximation is already sufficiently accurate). This is an important aspect in inexact Newton methods.

More importantly though, the decoupling (4.1) leads to strongly nonsymmetric pressure blocks $\mathcal{A}^p$, and floating point errors in the calculation of $\mathcal{A}^p$ can lead to rows which are no longer diagonally dominant. This has a strong effect on the performance of the AMG pressure preconditioner, as we can see in Table 4.

4.2.2. Influence of the approximate inversion of the pressure block $\mathcal{A}^p$. In this section we investigate the influence that the approximation of $\mathcal{A}^p$ by one V-cycle of AMG (as done in Section 4.2) has on the overall performance of the block Gauss-Seidel preconditioner $\mathcal{P}_{BGS}^{-1}$ and of the combinative two-stage preconditioner $\mathcal{P}_{C2S}^{-1}$. Thus, we replace the pressure preconditioner $\mathcal{P}^p$ by $\mathcal{A}^p$ and use an LU factorisation of $\mathcal{A}^p$ to apply the (exact) inverse. The results for both the block Gauss-Seidel as well as the combinative two-stage preconditioner are given in Table 5 for the Quasi-IMPES decoupling, i.e. $\mathcal{P}_{\mathcal{L}}^{-1} := G^{\mathcal{Q}}$. They show hardly any difference between the

TABLE 4
Comparison of Gauss decoupling $G^{\mathcal{G}}$ and full decoupling $G^{\mathcal{F}}$ – Iterations

	Block Gauss-Seidel		Combinative Two-Stage	
Problem	$\mathcal{P}_{\mathcal{L}}^{-1} := G^{\mathcal{G}}$	$\mathcal{P}_{\mathcal{L}}^{-1} := G^{\mathcal{F}}$	$\mathcal{P}_{\mathcal{L}}^{-1} := G^{\mathcal{G}}$	$\mathcal{P}_{\mathcal{L}}^{-1} := G^{\mathcal{F}}$
A	4.2	20.4	3.5	14.2
B	9.4	13.6	4.6	12.0
C	13.0	16.9	2.8	11.7
D	3.8	32.4	3.2	26.1

TABLE 5
Comparison of AMG and LU for $\mathcal{A}^p$ (Quasi-IMPES decoupling) – Iterations

	Block Gauss-Seidel		Combinative Two-Stage	
Problem	$\mathcal{P}^p := \mathcal{P}_{AMG}$	$\mathcal{P}^p := \mathcal{A}^p$	$\mathcal{P}^p := \mathcal{P}_{AMG}$	$\mathcal{P}^p := \mathcal{A}^p$
A	4.2	3.6	3.5	2.6
B	10.2	10.6	3.8	3.7
C	13.6	12.6	6.0	5.8
D	3.9	3.1	3.1	2.3

two choices and we can thus conclude that the effect of AMG preconditioning is almost as strong as a direct solution of the pressure block, which is astonishing.

4.2.3. Influence of the approximate inversion of the saturation block $\mathcal{A}^s$. In the same way, to test the influence of approximating the saturation block $\mathcal{A}^s$ by one iteration of Gauss-Seidel in $\mathcal{P}_{BGS}^{-1}$ and in $\mathcal{P}_{C2S}^{-1}$ we replaced $\mathcal{P}^s$ by $\mathcal{A}^s$ and used again an LU factorisation of $\mathcal{A}^s$ to apply the (exact) inverse. This had hardly any effect at all. In Problems **A**, **C** and **D** the number of Bi-CGStab iterations did not change at all and in Problem **B** it was reduced at most by one iteration per system. Thus we can conclude that one iteration of Gauss-Seidel is largely sufficient to precondition $\mathcal{A}^s$ as well.

5. Conclusions. We have presented and tested a new preconditioning strategy for the linear equation systems arising in the simulation process of sedimentary basins. This strategy is based on similar techniques developed in the related field of oil reservoir simulations and aims at decoupling the system (locally) on the basis of three assumptions which were made in view of the physical background of the linear systems. After the decoupling the crucial task is a robust and efficient preconditioning of the pressure block. We saw that AMG satisfies both those criteria more than sufficiently. Finally, we effectuated the global preconditioning of the linear equation system by recoupling the different blocks either in a Gauss-Seidel fashion or using a combinative two-stage technique.

In a series of numerical tests from real case studies, the performance of the preconditioner was compared to ILU(0) which is used at the moment in `TEMIS3D` and we observed in almost all cases a considerable reduction of the CPU-time for the linear solver, up to a factor 4.3 with respect to ILU(0). Even more important, using the combinative two-stage technique to recouple the different blocks the performance of the preconditioner showed no degradation with respect to the number of elements, the size of the time step, high migration ratios, or strong heterogeneities and anisotropies in the porous media.

Acknowledgments. The authors would like to thank Léo Agelas, Isabelle Faille, Sébastien Lacroix, Philippe Quandalle, Stéphane Réquena, Françoise Willien, and Sylvie Wolf (all Institut Français du Pétrole), as well as Yves Achdou (Paris VII) and Yuri Vassilevski (Institute of Numerical Mathematics, Moscow) for fruitful discussions and comments.

REFERENCES

[1] S. Balay, W. Gropp, L. Curfman McInnes, and B. Smith, *PETSc Users Manual*, Report ANL-95/11 (Revision 2.1.0), Argonne National Laboratory, IL (2001).

[2] A. Behie and P.K.W. Vinsome, *Block iterative methods for fully implicit reservoir simulation*, Soc. Petroleum Eng. J, 22 (1982), pp. 658–668.

[3] H.C. Edwards, *A parallel multilevel-preconditioned GMRES solver for multi-phase flow models in the Implicit Parallel Accurate Reservoir Simulator (IPARS)*, TICAM Report 98-04, University of Texas, Austin, 1998.

[4] G. Gagneux and M. Madaune-Tort, *Analyse mathématique de modèles non linéaires de l'ingénierie pétrolière*, Springer, Berlin, 1996.

[5] S. Lacroix, Y.V. Vassilevski, and M.F. Wheeler, *Decoupling preconditioners in the Implicit Parallel Accurate Reservoir Simulator (IPARS)*, to appear in Numer. Linear Algebra Appl., 2000.

[6] S. Lacroix, Y.V. Vassilevski, J. Wheeler, and M.F. Wheeler, *An iterative solution of linear systems in the Implicit Parallel Accurate Reservoir Simulator*, to be submitted, 2002.

[7] J.W. Ruge and K. Stüben, *Algebraic Multigrid (AMG)*, In Multigrid Methods, S.F. McCormick, ed., Frontiers in Applied Mathematics, Vol. 5, SIAM, Philadelphia, 1986.

[8] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.

[9] F. Schneider, S. Wolf, I. Faille, and D. Pot, *A 3D basin model for hydrocarbon potential evaluation: Application to Congo Offshore*, Oil and Gas Science and Technology - Rev. IFP, 55 (2000), pp. 3–13.

[10] K. Stüben, *Algebraic Multigrid (AMG): experiences and comparisons*, Applied Mathematics and Computation 13 (1983), pp. 419–452.

[11] H.A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Comput., 12 (1992), pp. 631–644.