Full Approximation Scheme for Reservoir Simulation

Raymond Toft[1], Olay Møyner[1], and Knut-Andreas Lie[1]

Department of Mathematical Sciences, NTNU, Trondheim, Norway

Abstract

Simulation of multiphase flow and transport in porous rock formations give rise to large systems of strongly coupled nonlinear equations. Solving these equations is computationally challenging because of orders of magnitude local variations in parameters, mixed hyperbolic-elliptic character, grids with high aspect ratios, and strong coupling between local and global flow effects.

The state-of-the-art solution approach is to use a Newton-type solver with an algebraic multigrid preconditioner for the elliptic part of the linearized system. Herein, we discuss the use and implementation of a full approximation scheme (FAS), in which algebraic multigrid is applied on a nonlinear level. By use of this method, global and semi-global nonlinearities can be resolved on the appropriate coarse scale.

Improved nonlinear convergence is demonstrated on standard benchmark cases from the petroleum literature. The method is implemented in the solver framework of the open-source Matlab Reservoir Simulation Toolbox (MRST).

Keywords: Nonlinear Multigrid, Full Approximation Scheme, Reservoir Simulation, Multiphase Flow

1 Introduction

Multiphase flow in porous media arise in various applications like CO_2 storage, groundwater, geothermal energy, and fuel cells, to name a few. Herein, our primary motivation is application in reservoir simulation to predict hydrocarbon recovery. The underlying system of PDEs describes conservation of fluid phases or fluid components [1] and is generally parabolic. However, the system has a mixed elliptic-hyperbolic character in the sense that the evolution of fluid pressure is almost elliptic (unless the flow is strongly compressible) whereas the transport of fluid phases is almost hyperbolic (unless capillary forces are dominant). Discretizing such systems of equations result in strongly nonlinear discrete problems, which are challenging to solve accurately and efficiently.

The standard approach to use a finite-volume spatial discretization combined with an implicit or semi-implicit temporal discretization to handle large variations in time constants. The discrete nonlinear system is then linearized using Newton-type methods that rely on global linearizations to converge to a solution. Natural porous media is often highly heterogeneous, and spans large scales. For this reason, the linear systems to be solved simultaneously for fluid pressure and transported phases/components are often poorly conditioned and require specialized linear solvers. To obtain satisfactory convergence rates, sophisticated preconditioners tailored to the governing equations are typically employed. The state-of-the art approach used in reservoir simulation is the constrained-pressure-residual (CPR) method([19], [20]). This is a two-step preconditioner that utilizes the mixed characteristic of the system to decouple it into an elliptic-like subequation that can be efficiently solved with an algebraic multigrid method (AMG) ([16], [18]), together with a second-stage local solver for the hyperbolic or parabolic saturation and composition variables. Multigrid methods often have excellent parallel properties and allows the usage of modern massively parallel computer architectures, generally referred to as GPUs, to accelerate the computations and thus speed up the simulation. An example of this is AGMG with GPU acceleration, which have proven to be a robust, efficient and scalable solver ([15], [8]). Ideally, a multigrid solver should maintain the same convergence rate independently of the grid size [17]. In practise, one typically observes that the solver scales like $\mathcal{O}(N^{\alpha})$ for N unknowns and $\alpha \gtrsim 1.2$.

Herein, we discuss an alternative approach suggested recently for two-phase flow problems by [4, 3], in which the whole system is solved simultaneoulsy by a *nonlinear* multigrid method, called the full approximation scheme (FAS). Implementing and testing such a method in a realistic three-phase setting usually requires considerable effort and time. To accelerate the testing, we have utilized functionality from the Matlab Reservoir Simulation Toolbox (MRST), which is an open-source community framework for rapid prototyping of new computational methods for the subsurface sciences ([12], [10], [10, 11]).

2 Governing Equations

We start by denoting the reservoir domain by $\Omega \in \mathbb{R}^3$, and the boundary as $\partial\Omega$. Conservation of mass for fluid phase α can then be written in differential form as

$$\frac{\partial\big(S_\alpha\rho_\alpha\phi\big)}{\partial t} + \nabla\cdot\big(\mathbf{v}_\alpha\rho_\alpha\big) = q_\alpha. \tag{1}$$

Here, ρ_α is the density, $\mathbf{v}_\alpha$ is the phase velocity, and S_α is the fraction of the void volume occupied by phase α. The void volume is in turn a fraction ϕ of the total bulk volume, and the phases are assumed to completely fill this void space such that $\sum_\alpha S_\alpha = 1$. With this constrain we remove one variable by expressing the oils saturation as $S_0 = 1 - S_w - S_g$. The term q_α represents fluid sources or sinks, which in our case will be injection and production wells, respectively.

As fluid is flowing through the tortous flow channels of the porous medium, there will be some resistance to the flow. This resistance to the flow in a porous medium is described by Darcy's law ([6], [21]). The fluid velocities can then be expressed in the form

$$\mathbf{v}_\alpha = -\frac{\mathbf{K}k_{r\alpha}}{\mu_\alpha}\big(\nabla p_\alpha - \rho g_\alpha \nabla z\big). \tag{2}$$

Here, the permeability tensor $\mathbf{K}$ is the porous medium's ability to let a single fluid pass through, μ is the viscosity, p pressure, and g the gravitational acceleration. The relative permeability $k_{r\alpha}$ depends on fluid saturation S_α and measures the resistance caused by the presence of other fluid phases and how they relate to the pores of the rock ([14]). For shorthand, we introduce the mobility $\lambda = k_r/\mu$.

To get a full model, we also need to describe closure relationships for the phase densitites ρ_α and phase pressures p_α. For simplicity, the latter are assumed to be identical (i.e., capillary forces are ignored). If the fluids are assumed to have constant compressibility, the densities are then function of a single pressure p,

$$c_\alpha = -\frac{1}{V_\alpha}\frac{\partial V_\alpha}{\partial p} = \frac{1}{\rho_\alpha}\frac{\partial \rho_\alpha}{\partial p}. \tag{3}$$

In this study, we consider a three-phase model (oil, water and gas) with no dissolution between the three fluid phases. This gives us three primary variables, p, S_g and S_w.

Real reservoirs have complex geometries and are typically produced from multiple injection and production wells. In the following, we only consider conceptual cases consisting of a single set of injection and production wells place inside rectangular reservoirs described on uniform Cartesian grids and subject to no-flow boundary conditions. Flow in and out of the wells is described by a semi-analytical subscale model that must be solved for either the pressure inside the wellbore or the corresponding flow rate.

3 Discretization

The reservoir rock is represented as a volumetric grid consisting of a collection of distinct cells i, for $i \in C = \{1, \dots, N\}$. The topology of the grid is represented by the sets $\mathcal{N}(i)$ that contain all cells sharing an interface with cell i. We discretize the governing equations in two steps. First, we perform the spatial discretization, where we utilize the finite-volume method with cell centering. The spatial discretization then becomes

$$\frac{\partial\left(V_p S_{\alpha,i}\right)}{\partial t} = q_{\alpha,i} - \sum_{j\in\mathcal{N}(i)} T_{ij}\lambda_{\alpha,ij}\left(\Delta p - \rho_\alpha g \Delta z\right)_{ij}, \quad i \in C, j \in \mathcal{N}_{(i)}. \tag{4}$$

Here, $\Delta p_{ij} = p_i - p_j$ is the pressure difference between cells i and j, $\Delta z_{ij} = z_i - z_j$ is the vertical distance between the corresponding cell centroids, and $q_{\alpha,i}$ the source term in cell i. Instead of explicitly working with the porosity, the equation has been multiplied with the pore volume, $V_p = \phi V$. The flow of fluids across an interface between cells i and j is further represented by the mobility $\lambda_{\alpha,ij}$ and the transmissibility T_{ij}. The latter measures the fluid's efficiency in flowing and is a combination of geometric quantities and the permeabilities on opposite sides of the cell interface. For a uniform Cartesian grid, we express the transmissibility as:

$$T_{ij} = \frac{A_{ij}k_{ij}}{\Delta h_{ij}}, \quad i \in C, j \in \mathcal{N}_{(i)}, \tag{5}$$

where A_{ij} is the area of the interface between cells i and j, Δh_{ij} denotes the distance between the centroids, and the effective permeability between the cells is approximated by the harmonic mean

$$K_{ij} = \frac{2K_iK_j}{K_i + K_j}. \tag{6}$$

To ensure stability, the flux over the interface between cells are approximated with a mobility weighting. In reservoir simulation, single-point upstream weighting is a commonly used scheme [1]. In this scheme, the flow direction determines from which side the approximation is made:

$$\lambda_{ij} = \begin{cases} \lambda_j, & \text{if } \left(\Delta p - \rho_\alpha g \Delta z\right)_{ij} < 0 \\ \lambda_i, & \text{otherwise.} \end{cases} \tag{7}$$

For the temporal discretization we apply the backward Euler method and Equation (4) becomes

$$V_p S_{\alpha,i}^{n+1} = V_p S_{\alpha,i}^{n} + \Delta t\left(q_{\alpha,i}^{n+1} + f_{\alpha,i}^{n+1}\right), \tag{8}$$

with the residuals

$$r_{\alpha,i}(S_\alpha) = \Delta t\left(s_{\alpha,i}^{n+1} + f_{\alpha,i}^{n+1}\right) + V_p S_{\alpha,i}^{n} - V_p S_{\alpha,i}^{n+1}, \tag{9}$$

$$r_{v,i} = \left(\sum_\alpha S_\alpha^{n+1}\right) - 1. \tag{10}$$

This is the fully discretized system of the mass-conservation equations. With the boundary conditions we discussed above, we have a well-posed problem that admits a unique solution.

4 The Full Approximation Scheme

Now that we have established the mathematical model of the three-phase flow and discretized it, we turn our attention towards examining the method we use to solve our system of nonlinear equations.

4.1 The Fundamentals of Multigrid Methods

The multigrid method can be described as an recursive two-layered grid (h, H) process ([18]). The initial step consists of a iterative relaxation to remove highly oscillatory error modes: This step is often referred to as smoothing or the application of a smoother. Many classic iterative methods that employ approximate factorizations of linear systems remove local error modes quickly: Jacobi's method, variants of Gauss-Seidel relaxation, and incomplete LU factorization are all iterative solvers for which convergence rates are highly problem dependent ([17]). The residual error in a system can be classified as high- and low-frequency errors. High-frequency errors are local in nature and are relatively simple to remove with a few iterations of any of the aforementioned smoothers. Low-frequency modes are smoother and more global in nature. Smooth error components is one of the primary requirements allowing for an approximation of a system on a coarser grid without any essential loss of information. With a few iterations with e.g., Jacobi or Gauss-Seidel, we can assure that the error of our approximation is smooth enough to be approximated on a coarser grid. What were smooth errors on a fine grid, become less smooth on a coarser grid as the connections between the remaining degrees of freedom are shorter.

The fact that the different frequencies are more visible to smoothers on their respective grids is called aliasing of the frequencies. If high-frequency errors are not smoothed away first, thay may lead to insufficient convergence rates and construction of artifacts when the coarse system is solved and interpolated back to the fine grid. The residual is then corrected by restricting the domain into a coarser grid, on which the relaxation is continued. On the coarsest grid, the residual equation is solved to acquire a correction term. The grid solution is then interpolated back to the finer grid, where the approximate solution is corrected. This basic recursive method works as a result of the linearity of the residual equation.

4.2 Nonlinear Multigrid Method: FAS

For nonlinear problems, a slightly different approach may be needed than for the linear case. We will look into a method that applies the multigrid procedure directly to the nonlinear problem. This method is known as the full approximation scheme (FAS). The nonlinear FAS multigrid method computes a coarse-grid correction term by use of the residual from the finer grids ([2], [13], [9]). Even though the mechanics are very similar to many other multigrid methods, the FAS is actually computing a full approximate solution on the coarsest grid, rather than only acting on the residual as in linear multigrid. The FAS method is performed in cycles and uses multiple layers of grids that are determined in advance. The number of levels is noted as l and the method starts at the finest level $k = 1$. Here, we note the sequence of grids as Ω_k.

In the case of a nonlinear problem, we can write the system of equations on Ω_h as

$$N(\mathbf{u}) = \mathbf{f}, \tag{11}$$

where N is the nonlinear operator on $\mathbf{u}$. Frequently, the nonlinearity of a system is written as $A(\mathbf{u})$, but in this thesis we choose to use N for simplicity and clarity. Here we define the error $\mathbf{e} = \mathbf{u} - \mathbf{v}$ and the residual $\mathbf{r} = \mathbf{f} - N(\mathbf{v})$ in the same manner as in the linear case. Since N is nonlinear, $N(\mathbf{e}) \neq \mathbf{r}$ in general. This means we can not determine the error by solving a simple linear equation on the coarse grid as with linear multigrid. The residual equation must thereby be written as

$$N(\mathbf{u}) - N(\mathbf{v}) = \mathbf{r}. \tag{12}$$

By using this error relation we can rewrite (12) to

$$N(\mathbf{v} + \mathbf{e}) - N(\mathbf{v}) = \mathbf{r}. \tag{13}$$

Suppose we have an appropriate discretization N_h and found an approximation, $\mathbf{v_h}$, then (13) on Ω_h becomes

$$N_h(\mathbf{v_h}+\mathbf{e_h})-N_h(\mathbf{v_h})=\mathbf{r_h}. \tag{14}$$

The coarse grid correction on Ω_H is then

$$N_H(\mathbf{v_H}+\mathbf{e_H})-N_H(\mathbf{v_H})=\mathbf{r_H} \tag{15}$$

where the coarse grid residual is the restriction of the residual on the fine grid

$$\mathbf{r_H}=I_h^H\mathbf{r_h}=I_h^H(\mathbf{f_h}-N_h\mathbf{v_h}). \tag{16}$$

Here, I_h^H is the restriction operator. In the same way, the fine-grid approximation is restricted to the coarse grid with $\mathbf{v_H}=I_h^H\mathbf{v_h}$. This restriction of the approximation is what makes this method different from linear multigrid, where only the residual is restricted. By substituting the restriction into the coarse-grid residual equation (15), we can write it as

$$N_H(\underbrace{I_h^H\mathbf{v_h}+\mathbf{e_H}}_{\mathbf{u_H}})=\underbrace{N_H(I_h^H\mathbf{v_h})+I_h^H(\mathbf{f_h}-N_h(\mathbf{v_h}))}_{\mathbf{f_H}}. \tag{17}$$

Here, the right-hand side of the equation is known and on the same form as (11). If we assume that we find a solution $\mathbf{u_H}$ to the system, we can then compute the coarse grid correction term

$$\mathbf{e_H}=\mathbf{u_H}-I_h^H\mathbf{v_h}. \tag{18}$$

This can then be interpolated back to the fine grid and used to correct the fine-grid approximation $\mathbf{v}_h$:

$$\mathbf{v_h}\leftarrow\mathbf{v_h}+I_H^h\mathbf{e_H}. \tag{19}$$

If N_h and N_H are linear operators, it is easy to see that the FAS two-grid method is equivalent to an linear multigrid correction scheme. FAS is by this regarded as a generalization of coarse-grid correction for nonlinear problems. A variety of the FAS is written to view the method as an enhancement of the coarse-grid equations. The coarse grid equations (17) then takes the form

$$N_H(\mathbf{u_H})=\mathbf{f}_H+\tau_{\mathbf{h}}^{\mathbf{H}}, \tag{20}$$

where

$$\tau_{\mathbf{h}}^{\mathbf{H}}=N_H(I_h^H\mathbf{v_h})-I_h^H(N_h(\mathbf{v_h})) \tag{21}$$

defines the *tau correction* $\tau_{\mathbf{h}}^{\mathbf{H}}$. In the literature, $\tau_{\mathbf{h}}^{\mathbf{H}}$ is also called the (h,H)-*relative truncation error* since it is closely related to the role of the *truncation error* $\tau_{\mathbf{h}}$, which is the local discretization error between the continuous solution on Ω and the discretized approximation on Ω_h. By this analogy we see that since $\tau_{\mathbf{h}}^{\mathbf{H}}\neq 0$ in general, the solution $\mathbf{u_H}$ of the coarse grid,is not the same as the solution of the original equation. The solution $\mathbf{u_H}$ is actually converging towards an accuracy that matches the solution of the fine grid, but with the resolution of the the coarse grid.

As described earlier, the method received its name since the coarse-grid problem is solved for the full approximation rather than only the error $\mathbf{e}_H$. A complete summary of the FAS multigrid cycle is described in pseudocode in Algorithm .1.

Algorithm .1 : Pseudocode of the FAS multigrid cycle

```
procedure FASCYCLE($\mathbf{u_k^m}, N_k, \mathbf{f_k}, \nu_1, \nu_2, k, l$)
    $\mathbf{\bar{u}_k^m}$ = SMOOTHRESIDUALS($\mathbf{u_k^m}, N_k, \mathbf{f_k}, \nu_1$)
    $\mathbf{\bar{r}_{k+1}^m} = \hat{I}_k^{k+1}\left(\mathbf{f_k} - N_k\mathbf{\bar{u}_k^m}\right)$
    $\mathbf{\bar{u}_{k+1}^m} = I_k^{k+1}\mathbf{\bar{u}_k^m}$
    $\mathbf{f_{k+1}} = \mathbf{\bar{r}_{k+1}^m} + N_{k+1}\left(\mathbf{\bar{u}_{k+1}^m}\right)$
    if k < l then
        $\mathbf{\hat{v}_{k+1}^m}$ = FASCYCLE($\mathbf{\bar{u}_{k+1}^m}, N_{k+1}, \mathbf{f_{k+1}}, \nu_1, \nu_2, k+1, l$))
    else
        $\mathbf{\hat{v}_{k+1}^m}$ = SOLVE($N_{k+1}, \mathbf{v_{k+1}^m}, \mathbf{f_{k+1}}$)
    end if
    $\mathbf{u_k^m} = \mathbf{\bar{u}_k^m} + I_{k+1}^k\left(\mathbf{\bar{u}_{k+1}^m} - \mathbf{\hat{v}_{k+1}^m}\right)$
    $\mathbf{u_k^{m+1}}$ = SMOOTHRESIDUALS($\mathbf{u_k^m}, N_k, \mathbf{f_k}, \nu_2$)
    return $\mathbf{u_k^{m+1}}$
end procedure
```

4.3 Multigrid Components

For the restriction of the system, the simplest and most commonly used choice is the standard semi-coarsening in which the mesh size h is doubled in each direction except from one dimension. This is especially useful in reservoir simulation where the domain is very thin in the vertical direction compared to the lateral directions. We will thus use $H = 2h$ for each coarsening step in the x- and y-direction. For the restriction operator $I_h^{H=2h}$ there is a large number of possible functions to choose from in the literature. We utilize a simple full-weight summing function. In the two-dimensional case, this can be written in stencil notation as

$$I_h^H = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}_h^H . \tag{22}$$

The prolongation of the system is performed with an analogous full-weight distribution function, referred to as a full-weight injection function.

$$I_H^h = \left]\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}\right[_H^h . \tag{23}$$

The brackets are reversed to indicate that it is a distribution process. These grid operators is sufficient for the regular grids used in the numerical experiments presented later in this study. For irregular grids, there exist more sophisticated methods in the literature. We do not consider restriction of the equations and degrees-of-freedom corresponding to wells: We assume it is sufficient to include them on the finest scale, and coarsen the residual contributions to the reservoir equations.

For linear multigrid it is enough to apply a few iterations of Jacobi's method or Gauss-Seidel for pre- and post-smoothing. For nonlinear equations this is not necessarily sufficient since there may be *nonlinear* local error frequencies not present in a linearized system. We therefore require a relaxation method for nonlinear systems. We have for this application chosen to use a single Newton iteration, in which the Jacobian is inverted effectively with an iterative GMRES solver preconditioned by CPR-AMG. On the coarsest grid, the system is solved with the same Newton-CPR-AGMG method until convergence. The advantage of this approach is that it is both simple to implement, and it reduces the considerable efforts in implementing robust preconditioners for fully-implicit problems in MRST, while still separating the nonlinear problem into different levels. For all tests in this study, we performed a single Newton iteration for the pre- and postsmoothing. We use the standard automatic differentiation solvers from MRST in all our experiments.

5 Numerical Experiments

In this section the two different solvers are compared. The industry-standard solution method is referred as the standard Newton, SN, or Newton and the FAS method is conventionally referred as FAS(l), where l indicate the number of used grid levels.

For the numerical experiments, three different reservoir models are used. The first is a problem with homogeneous permeability and porosity, and a quarter-five-spot (QFS) well-pattern with a single injector and producer ([22]). For our second test, we take the top layers of the SPE 10, Model 2 [5], which is a standard benchmark model with significant variance in the permeability and porosity fields. We use the two top layer of this model, corresponding to the Tarbert formation. The significant heterogeneity of the permeability field makes the simulation problem more challenging than the QFS problem. Our final tests are performed with a realization of the Olympus reservoir model ([7]). This optimization benchmark model uses a synthetic 3D reservoir, inspired by an oil field in the North Sea. The model is an unstructured grid with a permeability field consisting of multiple layers, several faults, and two zones separated by an impermeable shale layer. This provides a challenging test case and is the largest model in our tests with a few hundred thousand cells.

For all simulations, the pressure is initialized to 250 bar in all grid cells. The length of the simulation time is based on the size of the models. The simulation time is 30 years for the validation tests, 5 years for the grid-scaling and timestep tests, and 150 days for the injection rate tests. The same stopping criteria is applied for the simulations with both Newton and FAS, resulting in the same solution at convergence. The absolute tolerance is set to 10^{-5} and the relative tolerance is 10^{-4}. The parameters of the simulations is listed in Table 4 in Appendix A.

5.1 Quarter-Five-Spot

The QFS test is a standard benchmark in reservoir simulation due to its symmetry and known reference solution. The model is here also used to show difference in scalability properties between Newton and FAS. An illustration of the model is shown in Figure 1. Figure 2 shows the validation of FAS with the oil production compared with the industry-standard Newton solver.

For the first test we compare the number of outer iterations for the two methods with different grid sizes. The results are presented in Table 1. As we can see the algorithmic efficiency for FAS is higher than for the standard Newton. Both FAS(2) and FAS(3) requires fewer outer iterations. This difference grows in favor for FAS as the grid size is increased. This implies that the scalability of FAS is better than with Newton. For FAS(2) the change in outer iterations is close to negligible, and with three grid levels a second cycle is never performed. This is as expected with regard to the fact that multigrid methods are theoretically less affected by changes in grid sizes on the coarser grids.

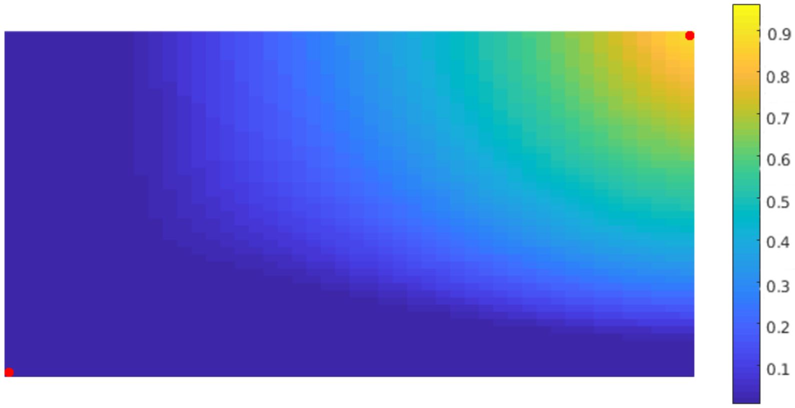

Figure 1: Quarter-five-spot model. A well in the top-right corner injects water and a production well is in the bottom-left corner. The water saturation is shown on the left side of the model. The domain size is 1000 × 500 × 20 ft.

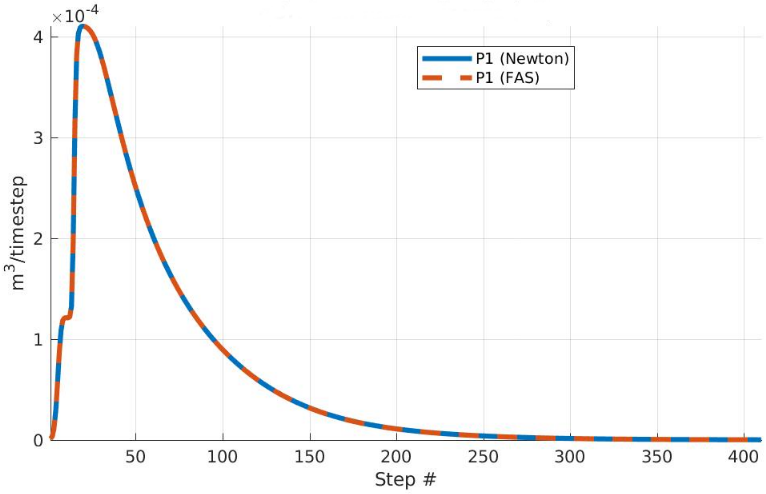

Figure 2: Validation of FAS against a industy standard Newton solver for the QFS model.

Table 1: Test of algorithmic efficiency with the QFS-model. Displays the average number of outer iterations for both Newton and FAS. Simulation duration were 5 years with 200 steps.

	Problem Size			
Method	16×16×10	32×32×10	64×64×10	128×128×10
Newton	2.05	2.07	2.09	2.16
FAS(2)	1.00	1.00	1.01	1.03
FAS(3)	1.00	1.00	1.00	1.00

In Figure 3 we see that the runtime of FAS is lower than for Newton for growing grid size. For the homogeneous permeability field the difference between applying two and three grids are close to negligible. FAS(3) is only slightly faster than FAS(2).

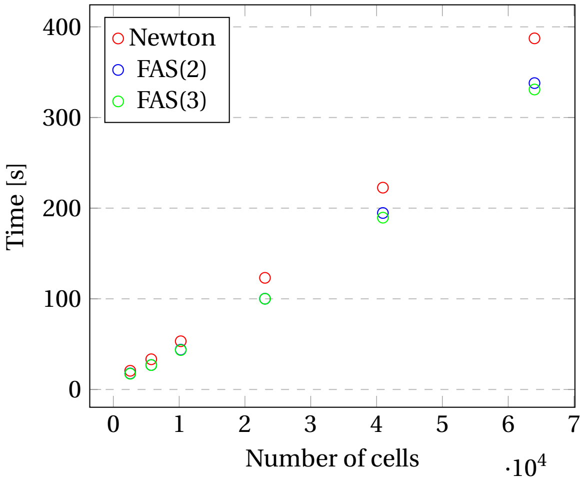

Figure 3: Runtime comparison between Newton and FAS for different grid sizes. The Permeability field was set to 100md with domain dimension of 1000 × 500 × 20m.

The case derived from Model 2 of SPE10 provides an appropriate benchmark test with its highly variable permeability field. Originally intended as an upscaling benchmark, the significant variation and freely available dataset has made it a de-facto benchmark for any novel solver. Figure 4 shows the permeability field of the top layer of SPE10, whereas Figure 5 reports the water saturation after injection with this permeability field. Here, the impact of the permeability field is clearly seen on the displacement front. As in the QFS example, FAS and standard Newton were tested for algorithmic efficiency.

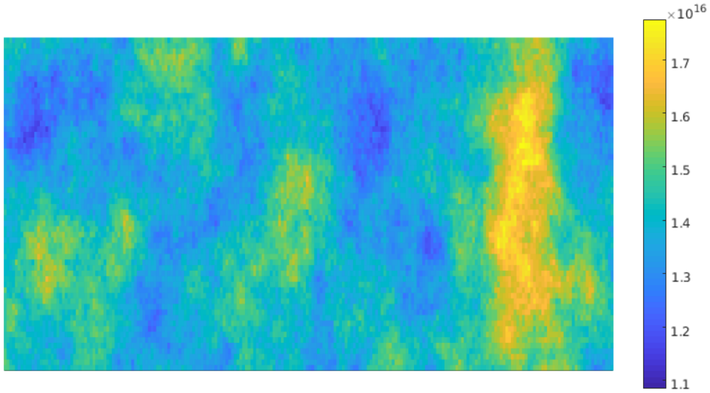

Figure 4: Logarithmic scaled permeability field of the top layer of the SPE10 model.

Compared to the previous case, we observe from Table 2 that FAS(2) requires more iterations than standard Newton for small timesteps. It is interesting to observe that only Newton needs additional iterations when longer time-steps are used. This is in contrast with FAS, which converges at approximately the same rate when the length of the time-steps increases, indicating a better treatment of the nonlinearity of the problem. It is worth noting that the linear systems do not generally

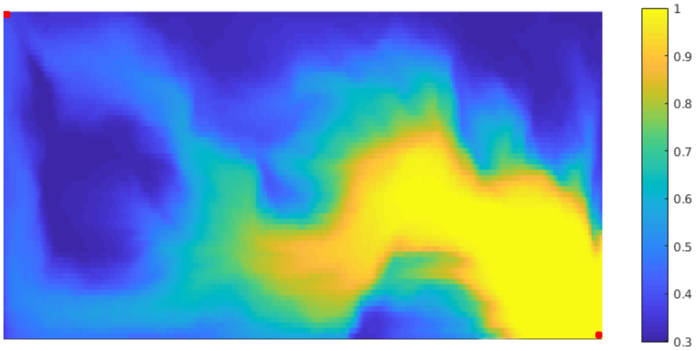

Figure 5: Illustration of the water saturation in the top SPE10 layer after 100 time-steps. An injection well is placed in the bottom right corner and a production well is in the top left corner. A highly heterogeneous permeability field significantly impacts the flow direction. The water saturation is shown on the left side of the model.

become more difficult to solve for longer time-steps. Longer time-steps are typically a source of reduced convergence, or even convergence failure in reservoir simulation. As before we see that FAS with three grids converges after only a single cycle. It is also worth noting that by investigation of runtime, FAS(3) uses approximately half the time FAS(2) requires to complete the simulation. This leads towards to the conclusion that FAS(3) is much more efficient than FAS(2) and Newton in terms of numerical efficiency.

Table 2: Test of algorithmic efficiency for the top SPE10 layer for different length of timesteps. The Displays the average number of outer iterations for both Newton and FAS. The duration of the simulation was 5 years.

	Step size in days					
Method	$\frac{1}{2}$	1	2	4	8	16
Newton	2.04	2.19	2.67	2.40	2.81	3.15
FAS(2)	2.99	2.99	2.98	2.97	2.95	2.93
FAS(3)	1.00	1.00	1.00	1.00	1.00	1.00

5.2 Olympus

The Olympus reservoir simulation model is an artificial model based on existing fields in the Nordic Sea. It consists of 16 layers and 6 faults. A fault is a displacement of the layers. This gives a discontinuity in the permeability fields and is frequently occurring in oil reservoirs. In reservoirs, one can often observe large differences in permeability between different rock types. The permeability field seen in Figure 6 is a typical example of high permeable channel deposits interbedded on a low-permeable background flood-plaine deposit. Here, the main flow will follow the high-permeable river channels, while at the same time the background flood-plain deposits have sufficient permeability to transmit fluids.

The injection rate indicates the speed and stress on the phases in the reservoir. In Table 3 we see the results from 150 days of simulation with 100 timesteps. As we observe, Newton have an increasing amount of outer timesteps for the corresponding increase of injection rate. The only exception

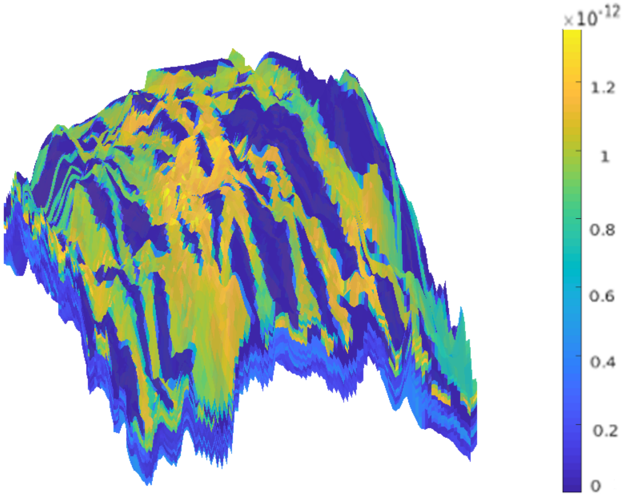

Figure 6: Illustration of the permeability field of the Olympus model.

seems to be with an injection rate of 80m^3/day, where it is a drop in the average number of iterations. For the same injection rate, we see that FAS(2) also have the lowest amount of iterations. The tendency we see with Newton is not present for FAS(2). There seems to be no particular pattern or correspondence between increasing injection rate and the algorithmic efficience of FAS(2). The results show that FAS(2) on average needs only slightly more than a single cycle to reach convergence for the different cases. As with most of the other results, we see that FAS have a significant advantage in algorithmic efficiency in reservoir simulation compared with the standard Newton.

Table 3: Test of algorithmic efficiency for an Olympus model with different injection rates. Displays the average number of outer iterations for both Newton and FAS. The simulation duration were 150 days with 100 timesteps.

	Injection rate (m^3/day)					
Method	40	60	80	100	120	140
Newton	2.32	2.34	2.31	2.35	2.37	2.38
FAS(2)	1.01	1.03	1.00	1.05	1.01	1.03

6 Conclusion

In this work, we have investigated the applicability of the nonlinear multigrid method FAS for compressible three-phase reservoir simulations. To investigate the method, we have studied two conceptual test cases with immiscible fluids and constant compressibility. With the standard Newton solvers for three-phase flow as a starting point, we have implemented the method in a generic manner that will later allow us to incorporate more advanced fluid behavior, reusing the already proven components of MRST. Our approach, favoring reuse and generic components, resulted in a semi-global linearization that was benchmarked against a standard Newton method with global linearization. For both solvers, we used a state-of-the-art CPR-AMG preconditioner to accelerate the solution process.

We have demonstrated that for the given model equations and problems considered, FAS outperforms standard Newton in terms of algorithmic efficiency. For the tests, both homogeneous and heterogeneous permeability fields have been applied. We have also demonstrated that the FAS method can easily be implemented with already existing solution methods in MRST as building blocks.

7 Future Work

Following this preliminary work for the application of FAS for reservoir simulation, there are several possible avenues for further research. One possible and natural direction is to extend the method to more advanced flow systems from MRST: Black-oil flow with dissolution and/or fully compositional problems containing strong nonlinearities, for which the FAS methodology may prove beneficial. Another consideration is systematic benchmarking of different choices for interpolation/restriction, smoothers, and other solver components. One could also consider implementing parts of the algorithm in a compiled language for a more accurate runtime assessment in a parallel environment.

8 Acknowledgments

The authors would like to thank Anne C. Elster at the HPC-lab at NTNU. The support and access to the HPC-lab have been of great help towards the completion of this work.

References

[1] K. Aziz and A. Settari. *Petroleum Reservoir Simulation*. Applied Sciences Publishers Ltd, Essex, UK, 1979.

[2] Achi Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods*, pages 220–312, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[3] M. l. C. Christensen. *Multilevel techniques for Reservoir Simulation*. PhD thesis, Technical University of Denmark, Kongens Lyngby, Denmark, 2016.

[4] M. l. C. Christensen, Klaus Langgren Eskildsen, Allan Peter Engsig-Karup, and Mark Wakefield. Nonlinear multigrid for reservoir simulation. *SPE Journal*, 21(3):888–898, 2016. doi:10.2118/178428-PA.

[5] M. A. Christie and M.J. Blunt. Tenth spe comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Evaluation & Engineering*, 4:308–217, 2010.

[6] H. Darcy. *Les fontaines publiques de la ville de Dijon*. Dalmont, Paris, France, 1856.

[7] R.M. Fonseca, C.R. Geel, and O. Leeuwenburgh. Description of olympus reservoir model for optimization challenge. 2017.

[8] R. Gandham, K. Esler, and Y. Zhang. A GPU accelerated aggregation algebraic multigrid method. *Computers & Mathematics with Applications*, 68 (10):1151–1160, 2014.

[9] V. E. Henson. Multigrid methods for nonlinear problems: An overview. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 5016, pages 36–48, 2003. doi: 10.1117/12.499473.

[10] S. Krogstad, K.-A. Lie, O. Møyner, H. M. Nilsen, X. Raynaud, and B. Skaflestad. MRST-AD - an open-source framework for rapid prototyping and evaluation of reservoir simulation problems. *SPE Reservoir Simulation Symposium*, 2015.

[11] K-A. Lie. *An Introduction to Reservoir Simulation Using MATLAB - User Guide for the Matlab Reservoir Simulation Toolbox (MRST)*. SINTEF ICT, Department of Applied Mathematics Oslo, Norway, 2016.

[12] K.-A Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad. Open-source MATLAB implementation of consistent discretisations on complex grids. *Computational Geosciences*, 16 (2):297–322, 2012.

[13] J. Molenaar. Multigrid methods for fully implicit oil reservoir simulation. In *Proceedings Copper Mountain Conference on Multigrid Methods*, volume Report 95-40, pages 581–590. Delft University of Technology, 1995.

[14] M. Muskat and R. D. Wyckoff. *The flow of homogeneous fluids through porous media.* McGraw-Hill Book Company, New York, USA, 1937.

[15] Y. Notay. An aggregation-based algebraic multigrid method. *Electronic Transactions On Numerical Analysis*, 37:123–146, 2010.

[16] J. Ruge and K. StÃijben. Algebraic multigrid. In *Multigrid Methods*, chapter 4, pages 73–130. SIAM, Pennsylvania, USA, 1987.

[17] Y. Saad. *Iterative Methods for Sparse Linear Systems.* SIAM, Philadelphia, USA, 2003.

[18] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid.* Academic Press, San Diego, California, USA, 2001.

[19] J.R. Wallis. Incomplete gaussian elimination as a preconditioning for generalized conjugate gradient acceleration. In *SPE Reservoir Simulation Symposium*, volume 27, pages 325 – 334, California, USA, 1983. SPE.

[20] J.R. Wallis, R.P Kendall, and T.E. Little. Constrained residual acceleration of conjugate residual methods. In *SPE Reservoir Simulation Symposium*, volume 37, pages 415–428, Texas, USA, 1985. SPE.

[21] S. Whitaker. Flow in porous media i: A theoretical derivation of darcy's law. *Transport in Porous Media*, 1(1):3–25, 1986.

[22] G. Willhite. *Waterflooding.* SPE, Texas, USA, 1986.

A Input values

For both models Table 4 contains the physical input values and reservoir-specific data used to generate the results presented in this study.

Medium	Property	Symbol	Value	unit
Water	Viscosity	μ_w	1	kg/ms
	Compressibility	c_w	10^{-8}	1/bar
	Density	ρ_w	1022	kg/m^3
	Initial Saturation	S_w	0.2	-
	Corey-exponent	-	2	-
Oil	Viscosity	μ_o	5	kg/sm
	Compressibility	c_o	10^{-5}	1/bar
	Density	ρ_o	800	kg/m^3
	Initial Saturation	S_o	0.7	-
	Corey-exponent	-	3	-
Gas	Viscosity	μ_o	0.1	kg/sm
	Compressibility	c_o	10^{-4}	1/bar
	Density	ρ_o	100	kg/m^3
	Initial Saturation	S_g	0.1	-
	Corey-exponent	-	5	-
Rock	Permeability	$\mathbf{K}$	100	md
	Porosity	ϕ	0.3	-

Table 4: Properties of the three phases and the rock formation.