SPE-173241-MS

Status of System-AMG for Reservoir Simulation Applications

Sebastian Gries and Hans-Joachim Plum, Fraunhofer Institute for Algorithms and Scientific Computing SCAI

This paper was prepared for presentation at the SPE Reservoir Simulation Symposium held in Houston, Texas, USA, 23–25 February 2015.

Abstract

Algebraic multigrid (AMG) methods for directly solving coupled systems of partial differential equations (PDEs) have been extensively used in various types of numerical simulations in engineering. A necessary condition for its efficient applicability is the simulation process being driven by elliptic components. In reservoir simulation the pressure, described by Darcy’s law, is known to drive the process and hence System-AMG should be applicable and outperform classical solvers.

In the context of adaptive and fully implicit reservoir simulations, the linearization of balance equations results in linear systems of equations that can be challenging. This makes it crucial to exploit all physical information from the full system to construct a robust AMG strategy and to extend it to more complex simulations than Black-Oil.

At the same time, the full set of information helps to get the best out of AMG. Just as with multigrid in general, System- AMG provides a framework for combining algorithmic modules rather than a fixed solution algorithm. The adaptation of the solution strategy to the concrete class of applications is the key to obtain the best performance.

Finally, System-AMG does not only allow for choosing more efficient algorithmic strategies, but also for exploiting parallelism in an optimized way, regardless of the simulation code being parallelized or not. Because the linear solver time is typically far dominating, even serial simulators can immediately and substantially benefit from MPI parallelism.

Introduction

The continuously growing complexity of fully and adaptive implicit reservoir simulations [1] makes scalable linear solvers essential. Algebraic multigrid (AMG) [2] [3] methods are well known to be a good choice, as long as certain ellipticity conditions on the system equations are fulfilled. In the oil reservoir simulation community, multigrid already is used, e.g., as a pressure preconditioner [4] [5] within the so-called Constrained Pressure Residual (CPR) [6] [7] method and can, compared to classical one-level techniques, provide massive accelerations.

The said complexity does not only refer to the sheer size of the problems to be solved, but also to the physical complexity of the modelling. Already in Black-Oil simulations, the linear systems involve several physical unknowns, namely pressure and saturations or concentrations. The more effects, e.g.

thermal or mechanical, are involved in the simulation, the more different physical unknowns have to be considered by the linear solver.

This makes the application of so-called *System-AMG* [8] [9] [2], a generalization of scalar AMG, a considerable option for these coupled systems. By regarding the *entirety* of the linear problem to be solved, System-AMG provides advantages that can be subsumed in the categories *extendibility, algorithmic versatility* and *efficiency:*

- *Extendibility*
  The System-AMG framework can in principal be applied to simulations with various further unknowns. Of course, it has to be analyzed whether System-AMG's requirements are met and how the framework should be set up concretely. However, the extension can be performed within the same framework, using the existing algorithmic modules.
- *Algorithmic versatility*
  With the full system matrix being available to the System-AMG approach, all information contained in the simulated physics can be exploited to create a better AMG strategy.
- *Efficiency*
  In addition to the "numerical efficiency", covered by the previous item, System-AMG also allows for an efficient parallelization and other performance measures. The full linear solution process is exposed to this "computational efficiency", rather than just a sub-part. A special variation of System-AMG enables all these variations of parallelism, e.g. distributed (i.e., normally, MPI based) one, even if the user code is not originally equipped with it. System-AMG can automatically distribute the AMG intrinsic data structures and features MPI parallelism. By exploiting modern computer architectures with MPI for the most time consuming part, the overall runtime of serial simulators can substantially benefit.

All said also implies that the *user interface* to System-AMG gets *simpler*: The user "just" passes the whole system to AMG and does not have to think about how to suitably embed AMG into the overall solution process. Nor do parallelism decisions have to be made – it is all opaque and encapsulated in System-AMG.

The requirement for an application of System-AMG is the coupled system being dominated by an elliptic component. This is the case for the pressure dominated problems in reservoir simulations, however, not necessarily in a straightforward way. The pressure related parts are not only influenced by the diffusive fluid flux, but also by source terms resulting from e.g. wells. These influences have to be carefully taken into account in order for AMG to construct a reasonable hierarchy [10].

This paper is organized as follows; First, a brief outline of System-AMG is given and afterwards, the linear problems to be solved in reservoir simulations and their general background are recalled. The fourth chapter describes an application strategy of System-AMG for these types of linear systems. Here it is also discussed, how System-AMG's requirements can be fulfilled – and at the same time a decoupling can be achieved for IMPES-cells from AIM simulations.

The following three chapters will describe the three basic advantages of System-AMG in more detail. Chapter five will briefly outline the extension to more complex simulations by exemplary test cases. In the sixth chapter, the adaptability of the approach will be demonstrated by deriving the robust applicability of aggressive coarsening which will result in up to 20% better solver runtimes (Figure 10) and less memory consumption. And finally, the last chapter will illustrate performance gains from exploiting the auto-parallelization of System-AMG for full simulations. This is presented for an exemplary SPE10 simulation, where the realized speedup of 2.2 is close to the theoretical infinite limit of 2.7 (Figure 12).

System-AMG

Algebraic Multigrid (AMG) methods have originally been developed to efficiently solve linear systems of equations with properties as they typically arise when discretizing scalar, elliptic partial differential equations (PDEs). By exploiting a hierarchy, with a cheap smoothing method at each level, multigrid methods provide a scalable solver strategy. In contrast to geometric multigrid, AMG approaches construct this hierarchy automatically. This allows AMG to rely only on the matrix that describes the system of equations. The matrices are required to – at least to some extent - feature certain properties that AMG aims to exploit (M matrix, definiteness). However, the automatic adaptation of the hierarchy to a concrete problem makes AMG generally suited for unstructured and heterogeneous problems, as they typically arise in reservoir simulations.

Algebraic multigrid provides the general strategy of constructing and exploiting a grid hierarchy. For the concrete realization, various options for the different steps of AMG are available. This allows to either rely on a robust default strategy or to adjust AMG to a concrete type of application. As long as the latter one can be realized robustly, performance gains can be obtained by choosing faster AMG components.

Detailed information on AMG methods and the different components can be found in the literature [2] [3] [11].

The situation is more involved if the linear problem to be solved is not resulting from a single, scalar PDE, but from a coupled system of PDEs. Here it is crucial to distinguish couplings between the different physical unknowns from those of the same unknown between different cells. The necessary information typically is available in the simulator anyway. Hence, it can be made available to AMG to extend the ideas of scalar AMG to a System-AMG [8] [2]. Basically two types of extension are possible that in principle could also be mixed.

In the so-called *unknown-wise* approach, AMG applies an individual coarsening and construction of transfer operators for the different physical unknowns. While independent hierarchies for the different unknowns are obtained, the cross couplings between them will not be neglected on the coarser levels. This interplay of individual coarsening and joint smoothing on the one hand requires unknown-wise smoothness of the error to imply overall smoothness. On the other hand, the method can easily account for different structures, e.g. different directions and strengths of anisotropies, for the different physical unknowns.

In contrast to this, the *point-wise* approach is closer to coarsening in a geometric grid by constructing one hierarchy that is used for all physical unknowns simultaneously. This is achieved by applying scalar coarsening to a primary matrix P, which properly represents the couplings between grid cells. Since the interpolation operators do not need to be separate per unknown, but can take point information into account, the point-wise approach can deal with much stronger cross couplings between the unknowns than the unknown-wise approach is able to.

Linear Systems in Reservoir Simulation

When simulating multiphase flow in porous media, the basic task of a reservoir simulation, the linear solver is faced with problems resulting from more than one single PDE. Each phase α (e.g. oil, gas or water) for each time step has to fulfil its individual mass conservation equation:

$$0 = \frac{\partial}{\partial t}(\varphi \rho_\alpha S_\alpha) - \nabla \left[\rho_\alpha \lambda_\alpha \vec{\vec{K}} (\nabla p_\alpha - g \nabla D) \right] + q_\alpha \tag{1}$$

Here, t is time, Φ the porosity and g and D are gravity and depth. ρ_α is the density of phase α and λ_α its mobility. The absolute permeability of the rock is given as K and the source term q_α accounts for external influences like wells. Finally, S_α and p_α are the phase's saturation and pressure – which are to be solved for in each time step.

There are various options in deriving slightly different mass conservation equations. However, all these options will not result in changes of the structure of the linear system to be solved in the end. For reasons of simplicity, these differences are not considered here. The reader is referred to the literature for a detailed discussion of modeling multiphase porous media flow [1].

The mass conservation equations (1) are highly nonlinear due to pressure and saturation dependencies in densities, mobilities and source terms. Solving the system of PDEs therefore requires a linearization which in practice usually is achieved by applying Newton's method. Here, additional, non-PDE relations like capillary pressures and saturation constraints are used to reduce the total number of unknowns to the number of PDEs. Then, with a Jacobian matrix that typically is ordered point-wise, the linear problem reads as:

$$J_{point\_wise}u =: \begin{pmatrix} [A]_{11} & \cdots & [A]_{1n_p} \\ \vdots & \ddots & \vdots \\ [A]_{n_p1} & \cdots & [A]_{n_pn_p} \end{pmatrix} \begin{pmatrix} [u]_1 \\ \vdots \\ [u]_{n_p} \end{pmatrix} = f \;\; \text{with } [u]_i = \begin{pmatrix} p_i \\ S_i \\ \vdots \end{pmatrix} \tag{2}$$

This is a coupled system, since in each cell different, interlinked unknowns have to be solved for. In the case of Black-Oil simulations, these are pressure and saturations. However, in more complex simulations, concentrations, temperature and mechanical unknowns will have to be considered in addition.

System-AMG for Black-Oil Simulations

Being able to solve Black-Oil problems efficiently is a prerequisite for extending the strategy to more complex simulations. Hence, System-AMG for this basic task is considered first.

System-AMG Strategy

In a Black-Oil simulation, the system (2) involves a pressure and one or two saturation unknowns. A natural System-AMG strategy for such a linear problem is derived from the following three considerations.

First of all, the pressure unknown is known to drive the fluid flow and dominate the problem (2). In addition, the pressure related parts of the Jacobian matrix are essentially based on the Darcy flow. It is therefore reasonable to define a point-wise AMG hierarchy by a primary matrix *P* based on these pressure parts.

Secondly, the full Jacobian, due to the cross couplings between the unknowns, will not be weakly diagonally dominant. Therefore, the Gauss-Seidel method, AMG's classical smoother, may diverge and ILU smoothing is preferable.

Finally, since the saturation related parts of (2) are based on hyperbolic and local effects, the smoother will typically not only smooth but to some extent solve for the respective unknowns. This waives the necessity of exploiting an AMG-hierarchy for the saturation unknowns.

For a plain Black-Oil simulation, this System-AMG approach yields a solution strategy close to the well-known Constrained Pressure Residual (CPR) method [6] [7]. However, the System-AMG approach allows exploiting the full set of AMG modules for solving the entire linear problem. This does not only provide the convenience of a single solver interface, but also allows exploiting the extendibility, versatility and efficiency of System-AMG, as claimed in the introduction. This will be outlined in the next chapters, after discussing the requirements for System-AMG to be applicable in the following.

Requirements for the pressure related parts of the Jacobian

Because the System-AMG hierarchy in Black-Oil simulations is constructed with a pressure based primary matrix *P*, *P* must fulfill certain properties that AMG will exploit for constructing its hierarchy. Reservoir simulations, however, may result in challenging linear problems for all types of hierarchical solvers. A preparation step is necessary with a matrix scaling that composes a *P* suited to construct a reasonable AMG hierarchy. That is what the earlier described Dynamic RowSum (DRS) preconditioner aims at [10]. It left-scales the point-wise Jacobian (2) with a block diagonal matrix:

$$C_{DRS} = \begin{pmatrix} [C_{DRS}]_1 & & \\ & \ddots & \\ & & [C_{DRS}]_{n_p} \end{pmatrix} \text{ where each } n_u \times n_u \text{ block is of the form } [C_{DRS}]_i = \begin{pmatrix} \delta_1^i & \delta_2^i & \cdots & \delta_{n_u}^i \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \tag{3}$$

For each cell, the different weights δ_j^i are computed according to the properties of the corresponding pressure related parts of the Jacobian. Essentially, P is a compromise between being close to the total pressure equation from IMPES simulation approaches and shielding AMG from problematic structures.

Remark: By experience, while the original linear problems often are challenging, System-AMG in practice usually is applicable after the DRS scaling. However, for the rare case of remaining problematic structures, within the system framework the scaling can advise the coarsening to exclude certain parts from the coarsening and apply some segregation process, e.g. Schwarz iterative methods, instead.

Saturation decoupling for IMPES cells in AIM simulations

While the DRS method aims at the applicability of AMG, it does not target at decoupling pressure and saturation. However, such a decoupling is of interest for those cells in AIM simulations, where the saturation part is treated explicitly. For such cells i, the pressure-saturation coupling is limited to the diagonal block. In particular, any block $[A]_{ji}$ reads:

$$[A]_{ii} = \begin{pmatrix} * & * & \cdots & * \\ * & * & & \\ \vdots & & \ddots & \\ * & & & * \end{pmatrix} \text{ and } [A]_{ji} = \begin{pmatrix} * & & & \\ * & & & \\ \vdots & & & \\ * & & & \end{pmatrix} \text{ for } j{=}i \text{ and } j{\neq}i\text{, respectively} \tag{4}$$

with * indicating a non-zero entry. Consequently, by decoupling pressure and saturations in the diagonal block $[A]_{ii}$, all couplings to the saturation unknowns of cell i could be removed. This would allow eliminating these unknowns from the linear system, to decrease its size, and compute them explicitly after the linear solver has converged.

Unfortunately, such a decoupling would heavily influence the respective pressure related entries. Clearly, it is not desired to achieve a size reduction of the linear problem at the cost of possibly waiving the applicability of AMG for an artificially scaled P. A compromise is to decouple all but one saturation unknowns by a combination of an additional right (SCE, [10]) and left scaling. However, one coupling would remain.

Instead, also by adjusting the ILU smoother at the finest level, it is possible to exploit the only local pressure-saturation coupling – and at the same time preserve the result from the DRS scaling for the hierarchical AMG process. Let C_{dcpl} be the nonsingular left scaling that would decouple all saturation-unknowns from IMPES cells in the point-wise Jacobian (2). And let $J_{dcpl}=C_{dcpl}\ J_{point\_wise}$ be the respective decoupled system matrix. Assume there is an incomplete factorization $J_{dcpl}=L_{dcpl}U_{dcpl}\text{-}R_{dcpl}$ given that results in a convergent iterative method for the linear system defined by J_{dcpl}. Then:

$$Id - U_{dcpl}^{-1}\, L_{dcpl}^{-1}\, C_{dcpl}\, J_{point\_wise} = Id - U_{dcpl}^{-1}\, L_{dcpl}^{-1}\, C_{dcpl}\, C_{dcpl}^{-1}\, J_{dcpl} = U_{dcpl}^{-1}\, L_{dcpl}^{-1}\, R_{dcpl} \tag{5}$$

From the calculation in (5) it can be concluded that the application of an incomplete factorization defined by L_{dcpl}, U_{dcpl} and C_{dcpl}, a decoupled smoothing, on J_{point_wise} results in the same operator as the incomplete factorization method for J_{dcpl}. Also for the factorization, J_{dcpl} does not have to be computed and stored explicitly, but the decoupling can be incorporated to the factorization algorithm applied to J_{point_wise}.

Figure 1 and Figure 2 for two representative test cases show the convergence behavior of System-AMG is not significantly influenced by this change in the smoother. The concrete runtime advantage depends

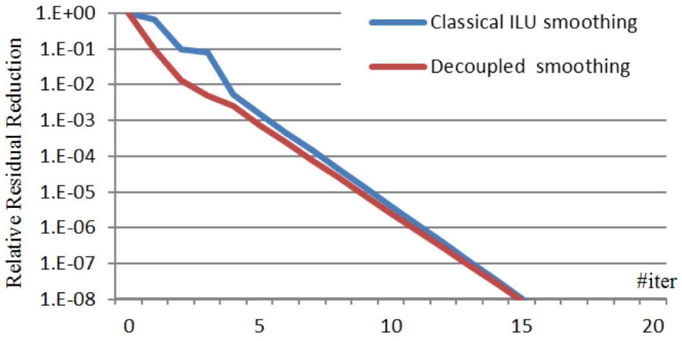

Figure 1—Convergence comparison of System-AMG with classical ILU(0) smoothing and the described decoupled smoothing. Full system matrix from a real AIM-simulation.

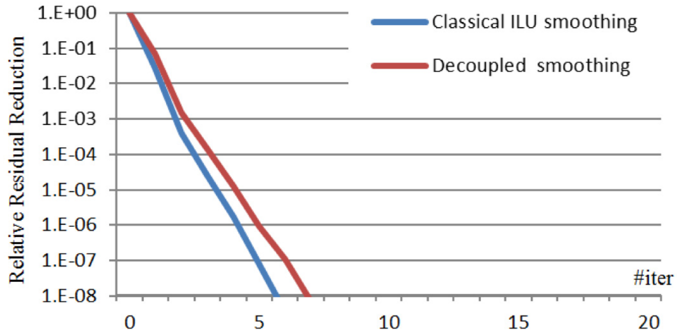

Figure 2—Convergence comparison of System-AMG with classical ILU(0) smoothing and the described decoupled smoothing. For a full system matrix from the SPE10 [12] case all off-diagonal saturation-related entries have been removed.

on the number of phases and components on the one and the amount of IMPES cells on the other hand. Both will determine the amount of work that can be saved in the factorization and its application.

Two remarks should be made:

- In the point-wise ordering it is important to compute the decoupled saturation unknowns only once all pressures, and further unknowns, have been updated. Otherwise, updated and old saturations will be mixed.
- In the extreme case of all saturation unknowns in all cells being treated explicitly, in the decoupled system Gauss-Seidel relaxation would be sufficient to achieve convergence. The decoupled smoother can account for this by choosing L_{dcpl} and U_{dcpl} such that the Gauss-Seidel method is formulated in the sense of an incomplete factorization – which does not require any computation. However, due to the weaker smoothing properties of Gauss-Seidel, the number of iterations may – slightly – increase compared to ILU smoothing, as seen in Figure 2.

Extendibility to more Complex Simulations

One advantage of the System-AMG approach is its extendibility to more complex simulations that typically will result in linear problems with additional physical unknowns involved. The potential of this extension by considering the additional matrix structures shall be illustrated with exemplary cases. A detailed discussion would be beyond the scope of this paper.

Compositional Simulations

In compositional simulations [1] [4], there’s not only a balance equation (1) per phase, but per component to be fulfilled. However, from a linear solver’s point of view, the resulting additional concentration

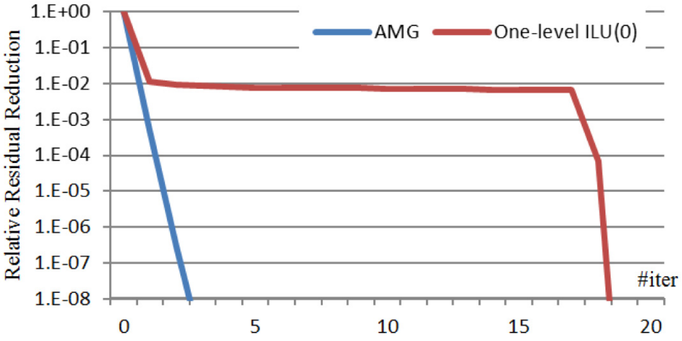

Figure 3—Convergence comparison of AMG and ILU(0) for a temperature sub-block from a steam injection simulation. Both methods are used as preconditioner for GMRes.

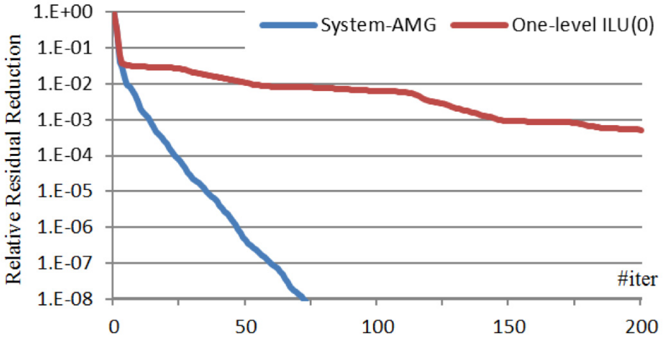

Figure 4—Convergence comparison of System-AMG and ILU(0) for a full system matrix from a steam injection simulation. Both methods are used as preconditioner for GMRes.

unknowns behave comparably to saturation unknowns and hence the basic structure of the Jacobian matrix does not change. The strategy derived for Black-Oil simulations therefore carries over.

Thermal Simulations

In the considered example here, the injection of steam instead of water – and the transport of the heat through the rock – is simulated [13]. In terms of modeling, this means the addition of an energy balance equation to the PDEs (1):

$$0 = \frac{\partial}{\partial t}\left(\varphi \rho_{fluid} U_{fluid} + (1-\varphi)\rho_{rock} U_{rock}\right) + \nabla \mathrm{H}\vec{u}_{total} - \nabla \vec{\vec{K}}_{\mathrm{thermal}} \nabla \mathrm{T} + q_{thermal} \tag{6}$$

Here U_{fluid} and U_{rock} are the internal, thermodynamic energy of the fluid and rock, respectively. u_{total} describes the fluid flux and H the enthalpy, the fluid's ability to transport energy. Finally, $q_{thermal}$ accounts for energy sinks and sources like additional in-situ heating facilities and $K_{thermal}$ is the thermal permeability, describing the rock's ability to transport energy in different directions.

The heat transport process is not a purely diffusive process, but also involves convective parts. Therefore one level solvers can typically deal with it better than with the highly diffusive fluid flow problems. However, the diffusive part for the heat transfer cannot be neglected and the special distribution of diffusive and convective influences is not clear a priori. AMG with its automatically constructed hierarchy therefore is a good candidate for this convective-diffusive process, as illustrated in Figure 3. For the full problem, System-AMG provides a significantly better convergence than one level approaches, as plotted in Figure 4.

Geomechanical Simulations

In geomechanical simulations [14] [15], the high pressure's influence on the rock – and hence the rock's transmissibility – is accounted for. This means, the momentum equilibrium needs to be fulfilled in addition to the mass balances (1). By inserting the Lamé equations, this results in the additional equation:

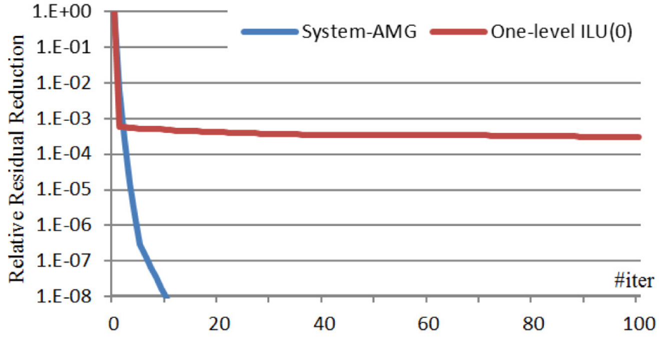

Figure 5—Convergence comparison of System-AMG and ILU(0) for a pure geomechanical problem without faults. (Emilia 923 from the University of Florida collection [16])

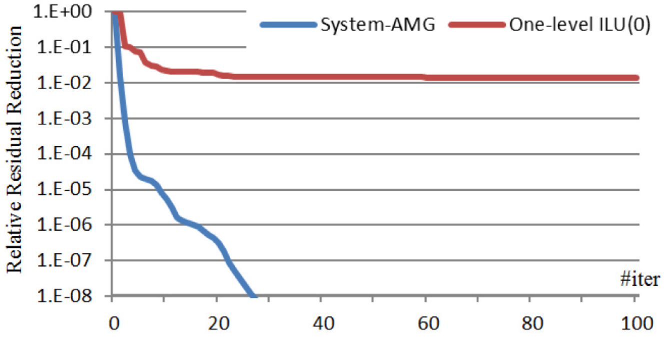

Figure 6—Convergence comparison of System-AMG and ILU(0) for a pure geomechanical problem with faults. (Fault 639 from the University of Florida collection [16])

$$0 = \nabla \lambda_L \nabla u + 2\mu_L \epsilon(u) - \alpha \nabla p_b + b \tag{7}$$

Here, u denotes the mechanical displacements. λ_L and μ_L are the first and second Lamé constants and α is the effective stress coefficient. p_b is the bulk pressure, $\varepsilon(u)$ is the displacement dependent strain and b a body force, typically gravity.

The fully coupled system of flow and mechanics has not yet been systematically studied with System-AMG. However, the additional mechanical part is a coupled system on its own, for which System-AMG is known to be in principle applicable. Figure 5 and Figure 6 illustrate the improvements that are possible compared to single level solvers.

Algorithmical Versatility: Aggressive Coarsening

The second advantage of the System-AMG approach is that all information of the full system and can be accessed and exploited in order to adapt the overall strategy, e.g. to improve performance. Effects of algorithmic changes apply to the full system, rather than to sub blocks only.

Adjusting the setup part is one option for tuning the System-AMG strategy in terms of performance. For instance, by constructing smaller coarse levels, the amount of work on these levels can be drastically reduced. That is what so-called aggressive coarsening [3] [2] aims at. Compared to classical coarsening, essentially, every other level of the hierarchy is omitted. This can be seen as 'two coarsening steps in one' and the resulting coarse level is much smaller than with non-aggressive coarsening, which yields the following advantages:

- a faster overall setup process
- each iteration will be cheaper

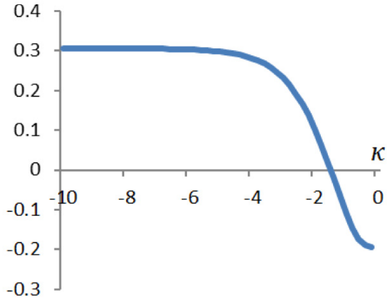

Figure 7—Plot of $\lambda\left(\frac{1}{h},1\right)-\lambda\left(\frac{1}{4h},1\right)$ for different values of $\varepsilon=10^k$. Positive values indicate identical smoothing factors for classical and aggressive coarsening.

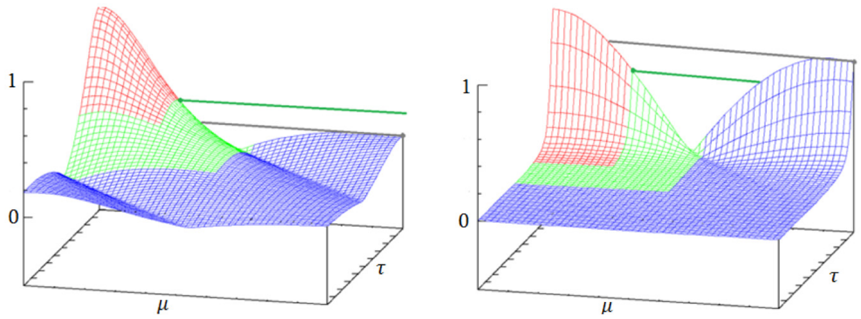

Figure 8—Plot of the reduction factors (μ,τ) for $\varepsilon=1$ (left) and $\varepsilon=10^{-2}$ (right). Those modes that have to be treated differently for classical and aggressive coarsening are marked in green. If these reduction factors are smaller than those colored in blue, identical convergence can be expected for classical and aggressive coarsening. A detailed description of the meaning all colors is given in the text.

- storing the hierarchy will require significantly less memory
- a smaller number of levels requires less data transfers in parallel applications

These advantages come at the price of a slower convergence. When dealing with a stand-alone, Laplacian based pressure correction problem, the cheaper setup and cycles usually outweigh the drawback of a slower convergence. However, in the application for reservoir simulations, a slower convergence is unacceptable. In contrast to the scalar problem, there are more unknowns to consider at the finest level, while the acceleration affects the hierarchical part only. This is, the choice of aggressive coarsening would have made sense in terms of the single pressure problem, while it appears questionable in the context of the entire linear problem.

However, ILU smoothing at the finest level has particular properties that can be exploited to nevertheless allow the exploitation of the aggressive coarsening strategy. A brief description of the corresponding AMG theory is given in the following, with a visualization in Figure 8.

Consider the anisotropic model problem:

$$\varepsilon\frac{\partial^2}{\partial x^2}u+\frac{\partial^2}{\partial y^2}=f \quad \text{with } x,y\in(0;1)^2 \text{ and } u=g \text{ for } x,y\in\partial(0;1)^2 \tag{8}$$

Let this model problem be discretized on a structured grid with cell width h by a five point stencil. Moreover, let $\varepsilon\leq 1$, the other case follows analogously. By local Fourier analysis (LFA) it was possible [17] to derive the absolute eigenvalues of the operator, representing the application of ILU(0) to the resulting matrix:

$$\lambda(\mu,\tau)=\frac{\varepsilon}{1+\varepsilon+\sqrt{2\varepsilon}}\,\frac{|\cos(\mu-\tau)\pi h|}{\left|1+\varepsilon-\varepsilon\cos(\mu\pi h)-\cos(\tau\pi h)+\frac{\varepsilon}{1+\varepsilon+\sqrt{2\varepsilon}}\cos((\mu-\tau)\pi h)\right|} \text{ with } \mu,\tau\in\left[1,2,\ldots,\frac{1}{h}\right)^2=\Omega\subset\mathbb{N} \tag{9}$$

This expression essentially determines how well each Fourier mode, defined by μ and τ, of the error can be reduced by each iteration of the ILU(0) method. Based on (9), the smoothing factor for ILU(0) can be derived. This quantity determines how efficiently ILU(0) reduces those high frequent error modes that cannot be transferred to the coarser level. Therefore, the smoothing factor has a direct impact on the AMG convergence. Which modes have to be considered high frequent, i.e. which fiber for μ and τ has to be taken into account, depends on the applied coarsening scheme. Maximizing (9) over the respective fiber results in the smoothing factor of ILU(0) for the particular coarsening:

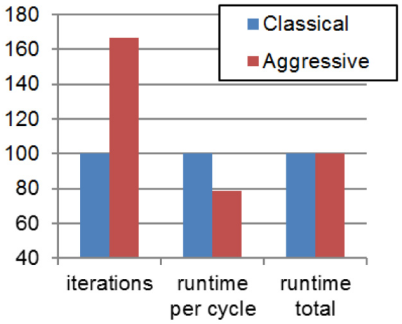

Figure 9—For comparison, the original pressure parts have been replaced by homogeneous graph Laplacians. Relative numbers (classical coarsening ≙ 100%).

- For classical coarsening the smoothing factor σ_{class} is derived as $\sigma_{class} = \lambda\left(\frac{1}{h} - 1{,}1\right)$
- For aggressive coarsening it can be shown σ_{aggr} is the maximum of $\lambda\left(\frac{1}{h} - 1{,}1\right)$ and $\lambda\left(\frac{1}{4h}, 1\right)$

Consequently, if σ_{aggr} was derived at $\lambda\left(\frac{1}{h} - 1{,}1\right)$, for the smoothing factor it would not make a difference whether classical or aggressive coarsening was used. A plot of the difference between both eigenvalues of importance, depending on ε, is given in Figure 7. For those ε that result in a positive difference, there will be $\sigma_{aggr} = \lambda\left(\frac{1}{h} - 1{,}1\right) = \sigma_{class}$. The break-even is at $\varepsilon = 10^{-\sqrt{2}}$. For smaller values, i.e. higher anisotropies, the smoothing factors of ILU(0) for classical and aggressive coarsening are identical. Hence, an identical convergence behavior can be expected.

In addition, in Figure 8, the plots of (μ,τ) over Ω for two different values of ε illustrate this situation. All modes colored in blue in any case have to be reduced by the smoother, whereas all modes colored in red never have to be considered for the smoother. Those modes colored in green make the difference between classical and aggressive coarsening. When classical coarsening is used, the smoother does not have to reduce these modes, while it has to do so for aggressive coarsening. If the maximal reduction factor for these modes, $\lambda\left(\frac{1}{4h}, 1\right)$, colored in dark green, is smaller than $\lambda\left(\frac{1}{h} - 1{,}1\right)$, colored in dark grey, the smoothing factor for classical and aggressive coarsening is identical.

While it is practically impossible to apply LFA theory to problems based on highly varying permeability fields, practical experience indicates the model problem's result carries over to problems from reservoir simulations. For high anisotropies, aggressive coarsening yields a convergence comparable to classical coarsening. Hence, the benefits of aggressive coarsening can be exploited.

First, for reasons of comparison, in Figure 9, the original pressure related parts of a full system matrix have artificially been replaced by graph Laplacians. This makes them homogeneous and consequently, also with ILU smoothing, the number of iterations drastically increases when using aggressive coarsening. This prevents the runtime advantage per iteration to carry over to the overall linear solution process.

In contrast to this experiment, Figure 10 illustrates the effect of aggressive coarsening applied to two representative, original matrices from industrial reservoir simulations, including all heterogeneities. Here, not only the memory consumption for storing AMG's hierarchy is reduced by 40-50%. Because the number of iterations does not increase, the acceleration per cycle also carries over to the full process and roughly 20% of the overall System-AMG runtime could be saved for these cases.

Efficiency: Parallelizing the Full Linear Solution Part of Sequential Simulation Codes by System-AMG

The third advantage of the presented approach is *efficiency*, or more precisely, *computational efficiency*. As the full linear system is in AMG's scope, all measures of optimization and parallelization can be

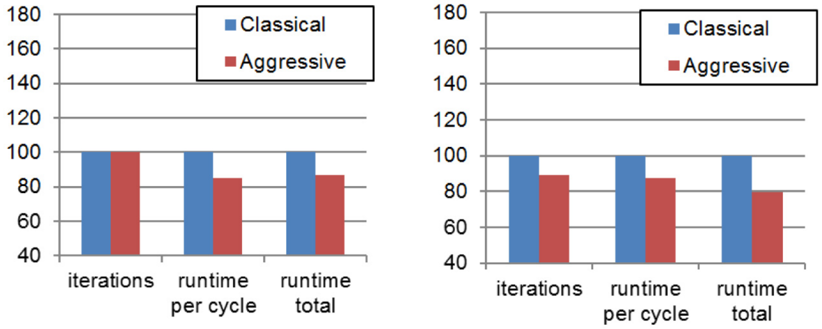

Figure 10—Comparison of classical and aggressive coarsening for two representative full system matrices from real reservoir simulation test cases. Relative numbers (classical coarsening ≙ 100%).

applied consistently to the entire system. The rule of thumb takes effect: the more data and operations are involved in the parallel sections, the better the efficiency. This is true for all paradigms that provide parallelism – loop based, multi-threaded, distributed parallelism and in principle even the application of special accelerator chips.

These measures can be encapsulated in AMG and the simulation code does not (necessarily) need to be adapted accordingly. This shall be demonstrated by a special variant of AMG using distributed parallelism for a full simulation of SPE10 [12] over 500 days simulated time. The AD-GPRS simulator [4] [18], developed at Stanford University has been used for these tests. Distributed, MPI based parallelism resulted in the best speedups for System-AMG.

At first glance, this may sound strange: Normally, distributed parallelism can only exist if the overall code runs distributed. However, the System-AMG variant works in a way that it receives non-distributed data as in a serial mode. It automatically distributes the AMG relevant data to the parallel architecture that could even involve different computing nodes. Fully parallel System-AMG is used to solve the problem and finally the solution is gathered back to the simulator.

There will naturally be some overhead for the auto-distribution, however, this overhead typically is small and the concept bears fruits. In particular, as in distributed parallelism in general, further powerful means are available, and have been used for the results below. For instance, the auto-distribution can use effective graph partitioners [19] in order to minimize parallel overheads. And not least, by allowing the usage of multiple compute nodes, as also done for the results below, the performance and memory limits are pushed without further work to be invested in the simulator.

It should be mentioned that such an approach is subject to “Amdahl’s Law”: This makes it crucial to apply the auto-parallelization to a dominating section of the simulator, i.e. the full linear solver part. If the section parallelized this way was not dominant and the rest is left serial, not much could be achieved. Consider for instance the application of this auto-parallel AMG for only a scalar pressure sub-problem. In the considered example, the AMG section then uses only 25% of the overall run time and even an “infinite” speedup in AMG would leave 75% of the run time. Hence, no more than 100/75=1.3 acceleration could be achieved.

System-AMG in turn covers roughly 60% of the work in SPE10, so that now at least factors better than 2 can be achieved. A speedup of 1.3 is already realized already by using two processes for System-AMG. Would, ideally, the AMG amount reach 90 % or more, the concept could provide accelerations in the 10x or better range.

Figure 11 shows the beneficial effect of graph partitioning. The speedup can be pushed to nearly eight, while it saturates earlier without using graph partitioning. Figure 12 shows the overall runtime of the simulation. The application of the auto-parallelized System-AMG gets the runtime close to the theoretical lower bound from Amdahl’s Law, indicated with a brown line. An overall speedup of 2.2 could be achieved.

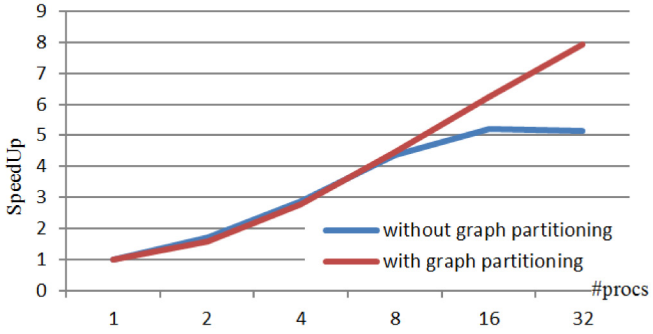

Figure 11—SpeedUp of the System-AMG part for an SPE10 simulation over 500 days. The auto-parallelization with and without graph partitioning is compared.

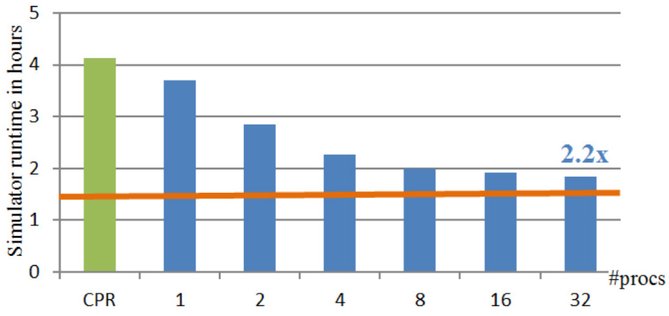

Figure 12—Overall runtime for an SPE10 simulation over 500 days with System-AMG. The brown line marks the lower bound from Amdahl's Law due to sequential non-solver parts.

Conclusions and Outlook

The application of the System-AMG, a generalization of AMG for coupled systems, for linear problems from reservoir simulations has been discussed in this paper. It has been described that the advantages of System-AMG, apart from the convenience of a single solver interface for the user, can be categorized as efficiency, algorithmic versatility and extendibility.

It has been demonstrated that System-AMG provides the possibility of efficiently parallelizing the entire linear solver part. Distributed memory parallelization can even be exploited by simulators that are not parallelized themselves. Because the linear solver typically is the dominating component w.r.t runtime, the overall runtime can substantially benefit from this mechanism. This has been illustrated with an SPE10 benchmark where the overall runtime has been accelerated by a factor of 2.2, where Amdahl's Law limited the ever achievable speedup to 2.7.

System-AMG does not only allow exploiting computational efficiency for the full system, but its algorithmical versatility allows the combination of the full set of available AMG components to the overall strategy. By exploiting all available physical information, as it can only be extracted from the full system, performance gains can be achieved by choosing a faster strategy. With the example of aggressive coarsening, in addition to significantly lower memory consumption, about 20% acceleration for the linear solver's runtime has been achieved.

In addition to the potential of accelerating the linear solution process, it has been demonstrated that the System-AMG in principle can be extended to simulations involving additional physical unknowns. This has been illustrated with exemplary test cases. Further research will target at defining robust and efficient System-AMG strategies for linear problems from the simulation of enhanced oil-recovery (EOR) processes.

Acknowledgements

This work has been done as part of an ongoing PhD-project at the University of Cologne.

The General Purpose Research Simulator (GPRS) developed by the Reservoir Simulation Research Group (SUPRI-B) at Stanford University was used in this work, to demonstrate the potential of parallelizing System-AMG. The authors would like to thank Timur Garipov for his advice in making the full system available to System-AMG.

References

1. K. Aziz und A. Settari, *Petroleum Reservoir Simulation,* Applied Sciences Publishers, 1979.
2. J. Ruge und K. Stüben, „Algebraic Multigrid (AMG)," in *Multigrid Methods,* Philadelphia, SIAM, Frontiers in Applied Mathematics, **5,** 1986.
3. K. Stüben, „*An Introduction to Algebraic Multigrid,*" in Multigrid, Academic Press, 2001, pp. 413–532.
4. H. Cao, *Development of Techniques for General Purpose Simulators,* Stanford University: PhD thesis, 2002.
5. S. Lacroix, Y. Vassilevski und J. Wheeler, „Iterative Solution Methods for Modeling Multiphase Flow in Porous Media Fully Implicitly," *SIAM Journal on Scientific Computing* **25,** pp. 905–926, 2003.
6. J. Wallis, „Incomplete Gaussian Elimination as a Preconditioning for Generalized Conjugate Gradient Acceleration," SPE Reservoir Simulation Symposium, 1983.
7. J. Wallis, R. Kendall, T. Little und J. Nolen, „Constrained Resdiaul Acceleration of Conjugate Gradient Acceleration," SPE Reservoir Simulation Symposium, 1985.
8. T. Clees, *AMG Strategies for PDE Systems with Applications in Industrial Semiconductor Simulation,* Shaker Verlag, 2005.
9. T. Clees und L. Ganzer, „An Efficient Algebraic Multigrid Solver Strategy for Adaptive Implicit Methods in Oil-Reservoir Simulation," *SPE Journal* **15**(3), pp. 670–681, 2010.
10. S. Gries, K. Stüben, G. L. Brown, D. Chen und D. C. Collins, „Preconditioning for Efficiently Applying Algebraic Multigrid in Fully Implicit Reservoir Simulations," *SPE Journal* **19**(4), pp. 726–736, 2014.
11. A. Cleary, R. Falgout, H. V., J. Jones, T. Manteuffel, G. R. J. Miranda und S. McCormick, „Robustness and Scalability of Algebraic Multigrid," *SIAM Journal on Scientific Computing,* Vol. **21,** pp. 1886–1908, 1998.
12. M. Christie und M. Blunt, „Tenth SPE Comparative Solution Project; a Comparison of Upscaling Techniques," *SPE Res Eval & Eng* **4**(4), pp. 308–317, 2001.
13. B. Rubin und W. L. Buchanan, „A General Purpose Thermal Model," *SPE Journal,* pp. 202–214, 1985.
14. J. Kim, H. Tchelepi und R. Juanes, „Stability, Accuracy and Efficiency of Sequential Methods for Coupled Flow and Geomechanics," *SPE Journal,* pp. 249–262, 2001.
15. L. Thomas, L. Chin, R. Person und J. Sylte, „Coupled Geomechanis and Reservoir Simulation," *SPE Journal,* pp. 350–358, 2003.
16. T. A. Davis und H. Yifan, „The University of Florida Sparse Matrix Collection," *ACM Transactions on Mathematical Software* **38**(1), pp. 1–28, 2011.
17. G. Wittum, „On the Robustness of ILU Smoothing," *SIAM Journal on Scientific Statistical Computing,* Vol. **10,** No. 4, pp. 699–717, 1989.
18. Y. Zhou, *Parallel General-Purpose Reservoir Simulator with Coupled Reservoir Models and Multisegment Wells,* Stanford University: PhD Thesis, 2012.

19. G. Karypis und V. Kumar, „A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *SIAM Journal on Scientific Computing,* Vol. **20,** No. 1, pp. 359–392, 1999.