NONLINEAR MULTIGRID SOLVER EXPLOITING AMGe COARSE SPACES WITH APPROXIMATION PROPERTIES

MAX LA COUR CHRISTENSEN[†‡], UMBERTO VILLA[§], ALLAN P. ENGSIG-KARUP[†], AND PANAYOT S. VASSILEVSKI[¶]

Abstract. The paper introduces a nonlinear multigrid solver for mixed finite element discretizations based on the Full Approximation Scheme (FAS) and element-based Algebraic Multigrid (AMGe). The main motivation to use FAS for unstructured problems is the guaranteed approximation property of the AMGe coarse spaces that were developed recently at Lawrence Livermore National Laboratory. These give the ability to derive stable and accurate coarse nonlinear discretization problems. The previous attempts (including ones with the original AMGe method, [5, 11]), were less successful due to lack of such good approximation properties of the coarse spaces. With coarse spaces with approximation properties, our FAS approach on unstructured meshes should be as powerful/successful as FAS on geometrically refined meshes. For comparison, Newton's method and Picard iterations with an inner state-of-the-art linear solver is compared to FAS on a nonlinear saddle point problem with applications to porous media flow. It is demonstrated that FAS is faster than Newton's method and Picard iterations for the experiments considered here. Due to the guaranteed approximation properties of our AMGe, the coarse spaces are very accurate, providing a solver with the potential for mesh-independent convergence on general unstructured meshes.

Key words. Nonlinear multigrid, Full Approximation Scheme (FAS), Element-based Algebraic Multigrid (AMGe), Nonlinear saddle point problem, Numerical upscaling, Multilevel upscaling, Mixed Finite Element Method (MFEM), Porous media flow, Subsurface flow.

1. Introduction.

The Full Approximation Scheme (FAS) is a multigrid method for nonlinear problems, [2, 7, 22, 8]. Its most widespread use is in geometric multigrid on structured grids due to difficulties associated with defining a coarse nonlinear operator on unstructured meshes. On unstructured grids, the most popular choice of nonlinear solver schemes is typically Newton-Krylov methods preconditioned by e.g. a black box method such as Algebraic Multigrid (AMG), [3, 20]. However, FAS offers potential benefits with respect to traditional methods, such as a larger basin of attraction, faster initial convergence, data locality and lower memory footprint. Several papers have addressed the application of FAS to unstructured grids. In [15, 16], FAS based on agglomeration multigrid is compared to Newton-Multigrid. In these papers, coarse grid control-volumes are formed by merging together finer grid control-volumes. Based on this agglomeration of control-volumes, the associated interpolators between grids are defined as simple injection/piecewise constants. In a multilevel context, piecewise constant interpolation between grids is insufficient and will result in loss of accuracy and therefore loss of performance in the overall multigrid scheme, [17]. An improvement was suggested in [17] to use an implicit prolongation operator, however, it may be too expensive to be worth the gain in convergence rate.

This paper can be seen as an extension of the work in [5, 11]. In these papers, FAS is combined with AMGe to obtain a nonlinear solver for lowest order nodal finite elements. Mesh-independent convergence is demonstrated (only) for an elliptic 2D model problem, [5]. The main difference between the work in our paper and [5, 11] is the underlying AMGe method providing the multigrid components, namely the restriction, prolongation and nonlinear coarse operators. In [5], the method is based on the AMGe introduced in [10, 23]. This results in coarse spaces, where only one degree of freedom can be used for each agglomerate. Consequently, it is difficult to maintain accuracy on very coarse agglomerate meshes, resulting in a degradation of the FAS solver performance. The version of AMGe used in this paper, [13, 14], allows the construction of operator-dependent coarse spaces for the whole de Rham complex (i.e. the sequence of H^1-conforming, $H(curl)$-conforming, $H(div)$-conforming and L^2-conforming spaces). This gives the foundation to cover a broad range of applications such as elliptic PDEs, Maxwell equations, Darcy flow equations, etc.

[†] Technical University of Denmark, Department of Applied Mathematics and Computer Science (max@maxlacour.com, apek@dtu.dk).

[‡] Lloyd's Register Consulting, Hellerup, Denmark.

[§] The Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin, (uvilla@ices.utexas.edu).

[¶] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, California, USA, (vassilevski1@llnl.gov).

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program.

In this paper, we restrict ourselves to the use the $H(curl)$, $H(div)$ and L^2 spaces. The recently developed AMGe technique with guaranteed approximation properties on coarse agglomerated meshes provides the coarse spaces used for the restriction, prolongation and nonlinear operators. These coarse spaces have the properties necessary (and are intended) to be used as an upscaling tool, but in this paper we demonstrate that the same coarse spaces can be reused for solvers. The coarse spaces have desirable properties analogous to the original finite element spaces: Nédélec, Raviart-Thomas and discontinuous piecewise. This is ensured by introducing additional degrees of freedom associated with non-planar interfaces/edges between coarse elements/faces (agglomerates of finer level elements/faces). In this way, the necessary number of degrees of freedom on coarse faces or coarse edges are automatically found via singular value decomposition.

The FAS-AMGe method implemented in this paper is tested on a nonlinear saddle point problem with applications in porous media flow. It is compared to exact and inexact Newton's method and Picard iterations. The comparison is done in a fair way by letting the FAS, Newton's method and Picard iterations utilize the same underlying components, namely the multilevel divergence free solver, recently developed at LLNL, for the solution of the mixed discretization of the Darcy problem.

The rest of the paper is structured as follows. In Section 2 we give a brief outline of the AMGe method we use. Section 3 summarizes the FAS we use in general terms. The model problem of our main interest is introduced in Section 4. A key ingredient of our solver, namely, a divergence-free preconditioner is briefly summarized in Section 5. The main part of this paper consisting of a large set of numerical tests, is given in Section 6. At the end, in Section 7 we provide some conclusions and perspectives.

2. Element-based Algebraic Multigrid (AMGe).

AMGe is a framework of multilevel methods for the solution of systems stemming from finite element discretizations. In contrast to AMG, where only system coefficients are used, AMGe also employs grid topology and finite element matrices. The specific version of AMGe used in this paper was introduced in [13, 14, 19]. The method facilitates the construction of operator-dependent coarse spaces which can be shown to guarantee approximation on coarse levels for general unstructured meshes. Thanks to the guaranteed approximation properties, this AMGe technique can be used as a discretization tool (upscaling) on coarse (agglomerated) meshes and allows for the generation of accurate coarse spaces for the FAS hierarchy.

In a setup phase, a hierarchy of agglomerated meshes is formed. Each agglomerate is formed by grouping together finer-grid elements (or agglomerates if already on a coarse level). For unstructured meshes, the agglomeration can be accomplished by the use of graph partitioners. In particular, this work uses METIS, [12], to form agglomerates. Once the hierarchy of agglomerated meshes is generated, coarse spaces are computed by restricting certain basis functions and by solving local saddle point problems for each agglomerate entity. A thorough description of the methods involved is out of scope for this paper. In addition, this version of AMGe allows to assemble the coarse grid residuals and Jacobians directly on coarse agglomerated meshes without visiting the fine grid. For details on the assembly procedure see [4], where the time-dependent two-phase porous media flow (reservoir simulation) is solved with optimal complexity on coarse (upscaled) levels. The software developed for this paper uses the Element-Agglomeration Algebraic Multigrid and Upscaling Library: ParElag developed at Lawrence Livermore National Laboratory. ParElag is based on the MFEM library, [1], for the finite element discretization and supports several solvers from the HYPRE library, [9].

3. Full Approximation Scheme (FAS).

The FAS, [2, 7, 22], can be considered as a generalization of multigrid methods to nonlinear problems. For a two-grid method, consider the nonlinear dicrete problem:

$$\mathcal{A}_h(u_h) = f_h, \tag{3.1}$$

where $\mathcal{A}_h$ is a nonlinear operator, and the subscript h indicates that all quantities are discretized on the fine grid. Introducing the approximate solution v_h, the residual equation is given by

$$\mathcal{A}_h(u_h) - \mathcal{A}_h(v_h) = r_h, \tag{3.2}$$

where $r_h = f_h - \mathcal{A}_h(v_h)$. Introducing the subscript H to refer to quantities defined on the coarse mesh, the coarse residual equation can be written as

$$\mathcal{A}_H(u_H) - \mathcal{A}_H(v_H) = r_H \;\Leftrightarrow\; \mathcal{A}_H(v_H + e_H) - \mathcal{A}_H(v_H) = r_H, \tag{3.3}$$

where e_H is the error $u_H - v_H$.

To restrict the fine quantities v_h and e_h, we use the projection operator $\Pi :\ h \to H$, while to restrict the residual r_h to the coarse grid we use the transpose of the prolongation operator $P :\ H \to h$. The operators P and Π are constructed by our AMGe algorithm such that $\Pi P = I_H$. More specifically, the coarse grid problem reads

$$\mathcal{A}_H\,\underbrace{(\Pi v_h + \Pi e_h)}_{u_H} = \underbrace{\mathcal{A}_H(\Pi v_h) + P^T r_h}_{f_H}, \tag{3.4}$$

The coarse grid correction is then given by $e_H = u_H - \Pi v_h$. This correction term is prolongated to the fine grid level by using the prologation operator P and the solution v_h is updated accordingly. Algorithm 1 contains a pseudo code for the multilevel implementation of a FAS V-cycle.

Algorithm 1 Pseudo code for FAS V-cycle implementation.
Inputs:
Approximate solution: $u_{l=0}$
Nonlinear operator: $\mathcal{A}_l$, $l = 0, \ldots,$ nLevels
Right hand side: $f_{l=0}$
Output:
Approximate solution $u_{l=0}$

```
function FAS_Vcycle(l)
    if l == nLevels-1 (coarsest grid) then
        Approximately solve $\mathcal{A}_l(u_l) = f_l$
    else
        Nonlinear smoothing of $\mathcal{A}_l(u_l) = f_l$
        Compute defect: $d_l = f_l - \mathcal{A}_l(u_l)$
        Restrict defect: $d_{l+1} = P^T d_l$
        Restrict solution: $u_{l+1} = \Pi u_l$
        Store approximate solution: $u_{\text{old}} = u_{l+1}$
        Compute right hand side for residual equation: $f_{l+1} = d_{l+1} + \mathcal{A}_{l+1}(u_{l+1})$
        Apply FAS_Vcycle(l+1) to compute updated $u_{l+1}$
        Compute correction: $v_{l+1} = u_{l+1} - u_{\text{old}}$
        Prolongate correction: $v_l = P v_{l+1}$
        Correct approximation: $u_l = u_l + v_l$
        Nonlinear smoothing of $A_l(u_l) = f_l$
    end if
end function
```

4. Model problem.

The saddle point problem of focus in this paper is chosen for the simplicity of the formulation, while still presenting itself as a numerically challenging nonlinear problem with applications in porous media flow. The problem stems from Darcy's law and reads

$$\left\{\begin{array}{l} k^{-1}(p)\mathbf{u} + \nabla p = f_u \\ \nabla \cdot \mathbf{u} = f_p, \end{array}\right. \tag{4.1}$$

where k is the conductivity field or permeability field as it is commonly called in petroleum engineering, p is the pressure and $\mathbf{u}$ is the velocity. The pressure dependency of the permeability is modeled as

$$k(p) = k_0 e^{-\alpha p}, \tag{4.2}$$

where k_0 is a user-given permeability at reference pressure 0. The problem can be used to model a steady-state single phase primary depletion of an oil reservoir, where the permeability decreases exponentially with the pressure.

For simplicity, we assume essential boundary conditions $\mathbf{u} \cdot \mathbf{n} = 0$, but non-homogeneous and natural boundary conditions can be handled in a similar way.

4.1. Notation. We now introduce some notation used throughout the paper. Let Ω be a bounded connected domain in $\mathbb{R}^d$ with a regular (Lipschitz continuous) boundary $\partial\Omega$, which has a well-defined unit outward normal vector $\mathbf{n} \in \mathbb{R}^d$. For the cases considered in this paper, $d = 3$. For the vectorial functions $\mathbf{u}, \mathbf{v} \in \mathbf{L}^2(\Omega) = [L^2(\Omega)]^d$ and scalar functions $p, w \in L^2(\Omega)$, we define the inner products $(\mathbf{u}, \mathbf{v}) = \int_\Omega \mathbf{u} \cdot \mathbf{v} \, d\Omega$ and $(p, w) = \int_\Omega p\, w \, d\Omega$. Finally, we introduce the functional space $H(\text{div}; \Omega)$ defined as

$$H(\text{div}; \Omega) := \{\mathbf{u} \in \mathbf{L}^2(\Omega) \mid \text{div}\, \mathbf{u} \in L^2(\Omega)\}.$$

Using the above notation, we define the functional spaces $\mathcal{R}$ and $\mathcal{W}$ as

$$\begin{aligned} &\mathcal{R} \equiv \{\mathbf{u} \in H(\text{div}; \Omega) \mid \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\}; \\ &\mathcal{W} \equiv L^2(\Omega). \end{aligned}$$

4.2. Weak formulation. To derive the weak formulation for the mixed system in equations (4.1), we multiply equations (4.1) with the test functions $\mathbf{v} \in \mathcal{R}$ and $w \in \mathcal{W}$ and integrate over the domain Ω. After integration-by-parts of the non-conforming terms and applying the no-flux boundary condition $\mathbf{u} \cdot \mathbf{n} = \mathbf{v} \cdot \mathbf{n} = 0$, we obtain the following variational problem

PROBLEM 1 *Find $(\mathbf{u}, p) \in \mathcal{R} \times \mathcal{W}$ such that*

$$\begin{cases} \Big(k(p)^{-1}\mathbf{u}, \mathbf{v}\Big) - \Big(p, \nabla \cdot \mathbf{v}\Big) = (f, \mathbf{v})\,, & \forall\ \mathbf{v} \in \mathcal{R} \\ \Big(\nabla \cdot \mathbf{u}, w\Big) = (q, w)\,, & \forall\ w \in \mathcal{W} \end{cases}$$

To solve the non-linear Problem 1 we consider both the Newton's and Quasi-Newton's (Picard) methods. In a compact notation, the Newton's/Picard's step reads

$$\begin{aligned} \text{Solve:}&\quad a(\delta\mathbf{u}, \delta p; \mathbf{v}, w) = -r(\mathbf{u}_{\text{old}}, p_{\text{old}}; \mathbf{v}, w), \qquad \forall(\mathbf{v}, \delta w) \in \mathcal{R} \times \mathcal{W}; \\ \text{Update:}&\quad \mathbf{u}_{\text{new}} = \mathbf{u}_{\text{old}} + \delta\mathbf{u}, \quad p_{\text{new}} = p_{\text{old}} + \delta p, \end{aligned} \tag{4.3}$$

where the residual variational form is

$$\begin{aligned} r(\mathbf{u}_{\text{old}}, p_{\text{old}}; \mathbf{v}, w) = \Big(k(p_{\text{old}})^{-1}\mathbf{u}_{\text{old}}, \mathbf{v}\Big) - \Big(p_{\text{old}}, \nabla \cdot \mathbf{v}\Big) - (f, \mathbf{v}) \\ + \Big(\nabla \cdot \mathbf{u}_{\text{old}}, w\Big) - (q, w), \quad \forall(\mathbf{v}, w) \in (\mathcal{R}, \mathcal{W}), \end{aligned} \tag{4.4}$$

and the bilinear form for the Jacobian (Approximate Jacobian) evaluated at $(\mathbf{u}_{\text{old}}, p_{\text{old}})$ is

$$\begin{aligned} a(\delta\mathbf{u}, \delta p; \mathbf{v}, w) = \Big(k(p_{\text{old}})^{-1}\delta\mathbf{u}, \mathbf{v}\Big) + \beta \cdot \Big(\frac{\partial k^{-1}}{\partial p}\Big|_{p=p_{\text{old}}} \mathbf{u}_{\text{old}}\delta p, \mathbf{v}\Big) \\ - \Big(\delta p, \nabla \cdot \mathbf{v}\Big) - (f, \mathbf{v}) + \Big(\nabla \cdot \delta\mathbf{u}, w\Big) - (q, w), \quad \forall(\mathbf{v}, w) \in (\mathcal{R}, \mathcal{W}). \end{aligned} \tag{4.5}$$

Here $\beta = 0$ leads to Picard iterations and $\beta = 1$ leads to Newton's method.

4.3. Mixed Finite Element Discretization. The variational non-linear problem 1 and its linearization in (4.3) is discretized with the Mixed Finite Element method. In particular, we let $\mathcal{R}_h \subset \mathcal{R}$ be the (lowest order) Raviart–Thomas finite element space consisting of vector functions with a continuous normal component across the interfaces between the elements and $\mathcal{W}_h \subset \mathcal{W}$ be the space of piecewise discontinuous polynomials (constant) scalar functions. It is well-known that this choice of finite element spaces satisfies the Ladyzhenskaya-Babuška-Brezzi conditions, and therefore allows for a stable discretization.

To obtain the discrete version of (4.3), let us denote with $\{\boldsymbol{\phi}^j\}_{j=1,\dots,\dim(\mathcal{R}_h)}$ a basis for the space $\mathcal{R}_h$ and $\{\psi^j\}_{j=1,\dots,\dim(\mathcal{W}_h)}$ a basis for the space $\mathcal{W}_h$. With this notation, the finite element solution $(\mathbf{u}_h, p_h)$ can be written as a linear combination of the basis functions $(\boldsymbol{\phi}^j, \psi^j)$. More specifically, letting $\mathbf{U} \in \mathbb{R}^{\dim(\mathcal{R}_h)}$ and $P \in \mathbb{R}^{\dim(\mathcal{W}_h)}$ denote the vectors collecting the finite element degrees of freedom $\mathbf{u}_h^i$, $i = 1, \ldots, \dim(\mathcal{R}_h)$ and p_h^i, $i = 1, \ldots, \dim(\mathcal{W}_h)$, we write

$$\mathbf{u}_h = \sum_{j=1}^{\dim(\mathcal{R}_h)} \mathbf{u}_h^j \boldsymbol{\phi}^j, \qquad p_h = \sum_{j=1}^{\dim(\mathcal{W}_h)} p_h^j \psi^j. \tag{4.6}$$

We introduce the finite element matrices M, B and N whose entries are given by

$$\begin{array}{rll} M_{ij} = & \left(k^{-1}(p_{h,\text{old}})\boldsymbol{\phi}^j, \boldsymbol{\phi}^i\right), & i,j = 1, \ldots, \dim(\mathcal{R}_h), \\ B_{ij} = & \left(\nabla \cdot \boldsymbol{\phi}^j, \psi^i\right), & i = 1, \ldots, \dim(\mathcal{W}_h), j = 1, \ldots, \dim(\mathcal{R}_h), \\ N_{ij} = & \left(\frac{\partial k^{-1}}{\partial p}\Big|_{p=p_{h,\text{old}}} \mathbf{u}_{h,\text{old}}\psi^j, \boldsymbol{\phi}^i\right), & i = 1, \ldots, \dim(\mathcal{R}_h), j = 1, \ldots, \dim(\mathcal{W}_h). \end{array} \tag{4.7}$$

The Galerkin formulation leads to the solution of the sparse linear system

$$\mathbf{AX} = \mathbf{B}, \tag{4.8}$$

where the block matrix $\mathbf{A}$ and block vectors $\mathbf{X}$ and $\mathbf{B}$ read:

$$\mathbf{A} = \begin{bmatrix} M & B^T + \beta \cdot N \\ B & 0 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \delta\mathbf{U} \\ \delta P \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{F} \\ Q \end{bmatrix}, \tag{4.9}$$

where $\beta = 0$ will lead to Picard iterations and $\beta = 1$ will lead to Newton's method.

4.4. Multilevel formulation. Using the AMGe technique in Section 2, we construct coarse spaces $\mathcal{R}_l$ and $\mathcal{W}_l$ for each level of the hierarchy l. The *inf-sup compatibility* of the coarse spaces is a direct consequence of the compatibility of the fine grid spaces $\mathcal{R}_0$ and $\mathcal{W}_0$ and of the commutativity of the diagram

$$\begin{tikzcd} \mathcal{R}_l \arrow[r, "D_l"] \arrow[d, shift right, "\Pi_l^{\mathcal{R}}"'] & \mathcal{W}_l \arrow[d, shift right, "\Pi_l^{\mathcal{W}}"'] \\ \mathcal{R}_{l+1} \arrow[r, "D_{l+1}"] \arrow[u, shift right, "P_l^{\mathcal{R}}"'] & \mathcal{W}_{l+1} \arrow[u, shift right, "P_l^{\mathcal{W}}"'] \end{tikzcd},$$

where, by commutativity, we mean that the identities $\Pi_l^{\mathcal{W}} D_l = D_{l+1}\Pi_l^{\mathcal{R}}$ and $D_l P_l^{\mathcal{R}} = P_l^{\mathcal{W}} D_{l+1}$ hold. Here the $D_i : \mathcal{R}_i \to \mathcal{W}_i$ $(i = l, l+1)$ is the discrete operator representing the mapping $D_i\mathbf{u}_i = \operatorname{div}\mathbf{u}_i \in \mathcal{W}_i$ for all $\mathbf{u}_i \in \mathcal{R}_i$; $P_l = [P_l^{\mathcal{R}}; P_l^{\mathcal{W}}] : (\mathcal{R}_{l+1}, \mathcal{W}_{l+1}) \to (\mathcal{R}_l, \mathcal{W}_l)$ is the prolongation operator from coarse to fine, and $\Pi_l = [\Pi_l^{\mathcal{R}}; \Pi_l^{\mathcal{W}}] : (\mathcal{R}_l, \mathcal{W}_l) \to (\mathcal{R}_{l+1}, \mathcal{W}_{l+1})$ is the projection operator.

Finally, to apply the FAS V-cycle, we let x_l be the unknowns $(\mathbf{u}_l, p_l)$, we define the nonlinear differential operator $\mathcal{A}_l : (\mathcal{R}_l, \mathcal{W}_l) \to (\mathcal{R}_l^*, \mathcal{W}_l^*)$ as

$$\langle A_l(x_l), y_l \rangle = r_l(\mathbf{u}_\mathbf{l}, p_l; \mathbf{v}_l, w_l), \quad \forall y_l = (\mathbf{v}_l, w_l) \in (\mathcal{R}_l, \mathcal{W}_l),$$

and, on the fine grid, we set $f_0 = 0$ to match (4.3).

5. Multilevel Divergence Free preconditioner. Each Newton/Picard step require the solution of a linear system of the form (4.8), where the matrix $\mathbf{A}$ is indefinite (saddle point problem). We use the GMRES method preconditioned by a specialized indefinite AMGe preconditioner (the Multilevel Divergence Free preconditioner - MLDivFree) developed at LLNL for the solution of the mixed formulation of the Darcy equations. More specifically, MLDivFree uses a hierarchy of AMGe coarse spaces to form a preconditioner for symmetric indefinite saddle point problems of the form in (4.8) (when $\beta = 0$). MLDivFree can be summarized in the following three actions:

1. Find $\hat{u}$ such that the divergence constraint $B\hat{u} = q$ is satisfied.
2. Find $u = \hat{u} + C\sigma$ such that $||M(\hat{u} + C\sigma) - f||^2_{M^{-1}} \to \min$, where C is the discretization of the curl operator (also obtained by AMGe).
3. Find p such that $||B^T p - Mu - f||^2_{M^{-1}} \to \min$. This is the dual operation of step 1.

In practice, this is implemented by a symmetric V-cycle with a sophisticated multiplicative smoother. The pre-smoothing involves first solving for each agglomerate a local saddle point problem. Next a divergence free correction is obtained by solving for $\delta u = C(\delta\sigma)$, where $\sigma \in H(curl)$ is computed by applying some smoothing iteration to the linear system $C^T M C\sigma = C^T f$. The post-smoother consists of the same two components but in the reverse order.

In the numerical results section, to allow for a fair comparison among all the nonlinear solvers, we will apply MLDivFree both as a preconditioner of the linear systems in Newton's and Picard's method and in the smoothing phase of FAS.

6. Numerical results.

In this section, scaling experiments are carried out for a structured hexahedral mesh and for an unstructured tetrahedral mesh. For the exact Newton's method and Picard iterations we solve the linear system using preconditioned GMRES up to a relative tolerance of 10^{-8} and absolute tolerance of 10^{-10}. The inexact Newton's method and Picard iterations are based on a Eisenstat-Walker type condition, [6], to determine the relative tolerance for the linear solver GMRES in each nonlinear iteration and prevent *oversolving*. We use the following expression: $\min(0.5, \sqrt{||r^k||_2/||r^0||_2})$. Globalization of the Newton/Picard method is achieved by backtracking. The stopping criterion of the nonlinear solvers used for all experiments is $||r^k||_2 \leqslant \max(\text{rtol}||r^0||_2, \text{atol})$, where $\text{rtol} = 10^{-6}$ and $\text{atol} = 10^{-8}$ are the relative and absolute tolerances. Furthermore, in this section, we use two measures of multigrid performance, namely the arithmetic complexity and the operator complexity. The arithmetic complexity C_a is defined as the ratio of the total number of degrees of freedom on all levels (fine grid and coarse) to the fine grid number of degrees of freedom. In a similar way, the operator complexity C_o is the ratio of the total number of non-zeros (in the mixed system) on all levels to the number of non-zeros on the fine grid. More specifically, we have

$$C_a = \frac{\sum_{l=0}^{\text{levels-1}} \dim(\mathcal{R}_l \times \mathcal{W}_l)}{\dim(\mathcal{R}_0 \times \mathcal{W}_0)} \qquad C_o = \frac{\sum_{l=0}^{\text{levels-1}} \text{nnz}(\mathbf{A}_l)}{\text{nnz}(\mathbf{A}_0)}. \tag{6.1}$$

We stress upon the fact that many methods in practice can achieve C_a close to unity and have acceptable approximation properties. However, it is also of vital importance to ensure that C_o is small (at least sufficiently less than two) since then the coarse systems take up much less memory than the fine grid problem.

6.1. Structured grid scaling.

The first study is a comparison between FAS, Newton's method and Picard iterations on a structured grid. The computational domain is the unit cube, discretized with a structured cartesian hexahedral mesh. The permeability coefficient k_0 is the realization of a lognormal spatially correlated random field displayed in Figure 6.1. It is generated by means of a truncated Karhunen-Loève (KL) expansion with 6 eigenmodes in each direction, a standard deviation of 3 and a correlation length of 0.1. The KL expansion is chosen for its ability to generate a grid-independent permeability field such that scaling experiments can be easily performed. Furthermore, it does have some similarities with actual permeability fields. The parameter α is set to 10.

Five different solver schemes are compared, namely

- (1) FAS with Picard linearization. Each smoothing step is using one V-cycle of the MLDivFree preconditioner.
- (2) Exact and (3) inexact Picard iterations with GMRES preconditioned by MLDivFree.
- (4) Exact and (5) inexact Newton's method with GMRES preconditioned by MLDivFree. Note that for preconditioning, MLDivFree uses the symmetric matrix stemming from the Picard linearization.

Figure 6.2 shows the computational time as a function of degrees of freedom for all solver schemes. The hierarchy of agglomerated meshes is structured (cartesian) and with a coarsening factor of 2 in each direction. The coarsest level is 1 agglomerate and the number of multigrid levels range from 3 (on the coarsest initial mesh) to 7 (on the finest initial mesh) for these experiments. It is evident that all solver schemes have mesh-independent convergence for the given problem. The inexact solvers are faster than the exact solvers and FAS is the fastest overall. Table 6.1 holds more information on the results.

We notice that, for this particular problem, the Picard's method converges in nearly the same number of iterations as Newton's method. We suspect that the suboptimal convergence of Newton is due to the fact that $\alpha = 10$ results in very small basin of attraction for the Newton's method and that backtracking is needed to ensure global converge of the Newton method.

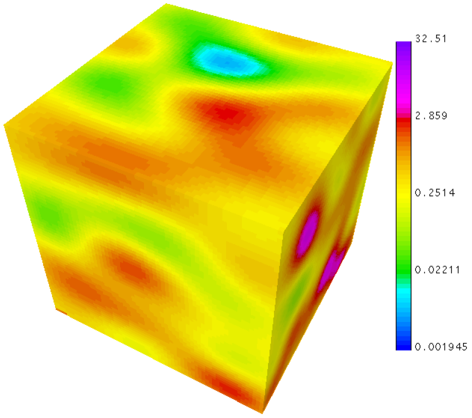

Fig. 6.1: Permeability field used for scaling experiments. Generated by a truncated Karhunen-Loève expansion.

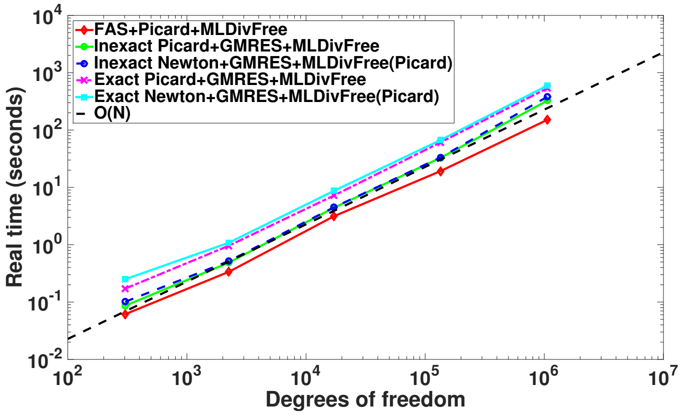

Fig. 6.2: Computational time for various solver schemes as a function of problem size (structured grid).

6.2. Unstructured grid scaling. In this section, we carry out a scaling experiment for an unstructured tetrahedral mesh. A unit cube is meshed with NETGEN, [18, 21], to produce 8 unstructured meshes with increasing resolution. We use the same permeability coefficient k_0 and parameter α as in the structured test case. For all 8 meshes, the performance of FAS is compared to the performance of inexact Picard. We restrict ourselves to these two nonlinear methods, since they have proven to be the fastest. The agglomeration is carried out using the graph partitioner METIS with a post-processing to ensure certain topological requirements are met. The first level of the mesh hierarchy is geometric (i.e. mesh derefinement) and the following levels are algebraic (i.e. METIS). The initial geometric level allows for smaller operator complexity. Figures 6.3 - 6.6 give an example of the topology produced by the procedure. The unstructured coarsening factor (METIS) used in these experiments is 100 finer elements per agglomerated element.

From Table 6.2, it can be seen that both algorithms perform optimally in terms of linear and nonlinear iterations. If we look at Figure 6.7, the computational time is good, but slightly suboptimal due to the operator complexity not remaining constant. This can be remedied by increasing the coarsening factor for the larger problem sizes.

FAS+Picard+MLDivFree	#elements	#linears	#nonlinears	time/time(FAS)
	64	-	5	-
	512	-	4	-
	4096	-	5	-
	32768	-	4	-
	262144	-	4	-
Inexact Picard+GMRES+MLDivFree				
	64	12	9	1.41
	512	11	7	1.45
	4096	13	9	1.41
	32768	12	9	1.71
	262144	12	10	2.16
Inexact Newton+GMRES+MLDivFree(Picard)				
	64	16	9	1.65
	512	12	7	1.54
	4096	13	9	1.43
	32768	12	9	1.73
	262144	12	10	2.52
Exact Picard+GMRES+MLDivFree				
	64	37	9	2.79
	512	33	6	2.85
	4096	34	7	2.30
	32768	37	7	3.17
	262144	35	7	3.55
Exact Newton+GMRES+MLDivFree(Picard)				
	64	59	9	4.07
	512	38	6	3.22
	4096	41	7	2.78
	32768	42	7	3.48
	262144	44	7	3.96

#elements	**#DoFs**	**operator complexity**	**arithmetic complexity**	**#levels**
64	304	1.32	1.17	3
512	2240	1.32	1.16	4
4096	17152	1.31	1.15	5
32768	134144	1.25	1.15	6
262144	1060864	1.18	1.15	7

Table 6.1: Information on the structured grid scaling experiments. #linears is the total number of linear iterations for all nonlinear iterations.

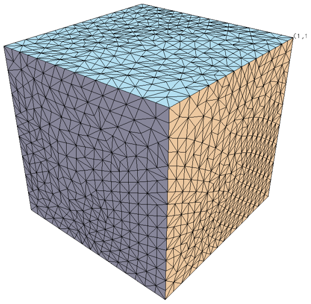

Fig. 6.3: Level 0
Uniformly refined once

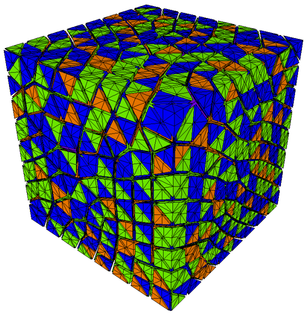

Fig. 6.4: Level 1
Agglomerates are based on the uniform refinement

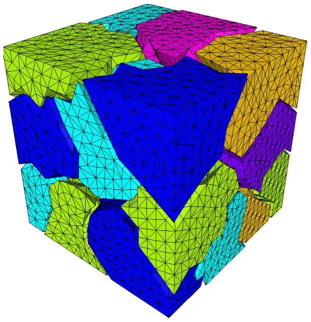

Fig. 6.5: Level 2
Agglomerates are formed by METIS

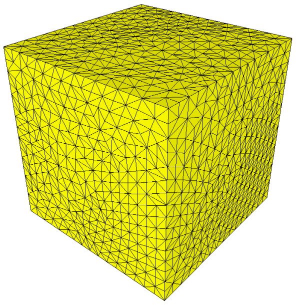

Fig. 6.6: Level 3
Coarsest level is one agglomerate

FAS+Picard+MLDivFree	Experiment	#linears	#nonlinears	time/time(FAS)
	32280	-	5	-
	53832	-	4	-
	67624	-	4	-
	215512	-	4	-
	405632	-	4	-
	496800	-	4	-
	679808	-	4	-
	827144	-	4	-
Inexact Picard+GMRES+MLDivFree				
	32280	13	9	1.42
	53832	13	9	1.66
	67624	13	9	1.59
	215512	11	9	1.33
	405632	11	9	1.20
	496800	12	9	1.32
	679808	12	10	1.28
	827144	13	9	1.18

#elements	#DoFs	operator complexity	arithmetic complexity	#levels
32280	98644	1.19	1.135	4
53832	164268	1.21	1.135	4
67624	207056	1.21	1.136	4
215512	654332	1.23	1.136	5
405632	1227916	1.29	1.140	5
496800	1507968	1.34	1.144	5
679808	2066320	1.34	1.140	5
827144	2500552	1.34	1.140	5

Table 6.2: Information on the unstructured grid scaling experiments. #linears is the total number of linear iterations for all nonlinear iterations.

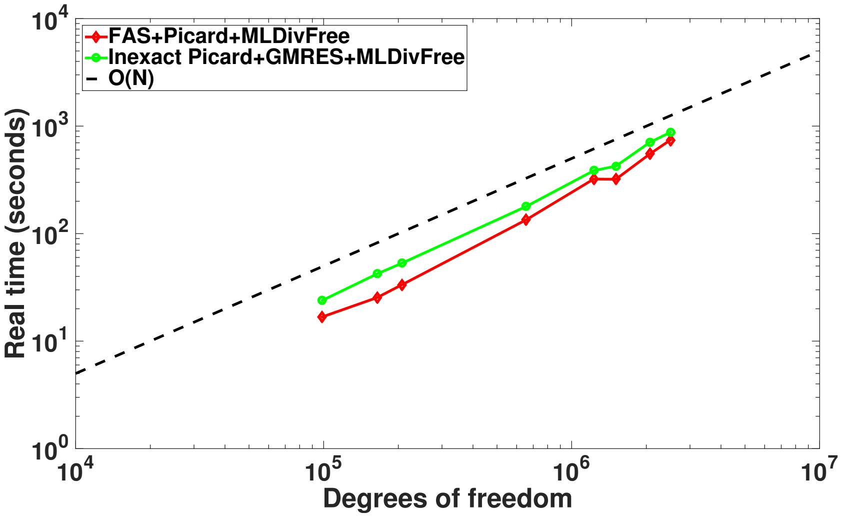

Fig. 6.7: Computational time for FAS and inexact Picard as a function of problem size (unstructured grids generated by NETGEN).

7. Conclusion & perspectives.

AMGe with guaranteed approximation properties has been combined with FAS to demonstrate a scalable nonlinear solver for a challenging saddle point problem. Numerical tests have been performed demostrating the mesh independent convergences of FAS (number of V-cycles). We compared FAS-AMGe to exact and inexact Newton's method and Picard iterations; FAS outperformed the exact methods (4X faster) and also proved slightly faster than the inexact versions. In this paper, only global linearization has been considered. A more thorough study comparing with local linearization techniques would be interesting.

REFERENCES

[1] *MFEM: Modular finite element methods.* mfem.org.

[2] Achi Brandt, *Guide to multigrid development*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., vol. 960 of Lecture Notes in Mathematics, Springer Berlin Heidelberg, 1982, pp. 220–312.

[3] A. Brandt, S. F. Mccormick, and J. W. Ruge, *Sparsity and its Applications; Algebraic multigrid (AMG) for sparse matrix equations*, Cambridge University Press, New York, 1984.

[4] M. L. C. Christensen, U. Villa, A. Engsig-Karup, and P. S. Vassilevski, *Numerical upscaling for incompressible flow in reservoir simulation: An element-based algebraic multigrid (amge) approach*, Lawrence Livermore National Laboratory Technical Report LLNL-JRNL-661295, (September 24, 2014).

[5] M. Dumett, P. Vassilevski, and C. S. Woodward, *A multigrid method for nonlinear unstructured finite element elliptic equations*, LLNL Technical Report UCRL-JC-150513, (2002).

[6] Stanley C. Eisenstat and Homer F. Walker, *Globally convergent inexact newton methods*, SIAM Journal on Optimization, 4 (1994), pp. 393–422.

[7] W. Hackbusch, *Multigrid Methods and Applications*, Springer, 1985.

[8] Van E. Henson, *Multigrid methods for nonlinear problems: an overview*, Proc. SPIE, 5016 (2003), pp. 36–48.

[9] hypre, *High performance preconditioners*. https://www.llnl.gov/casc/hypre/.

[10] Jim E. Jones and Panayot S. Vassilevski, *AMGe based on element agglomeration*, SIAM J. Sci. Comput., 23 (2001), pp. 109–133.

[11] J. E Jones, P. S. Vassilevski, and C. S. Woodward, *Nonlinear Schwarz-fas methods for unstructured finite element problems*, Second M.I.T. Conference on Computational Fluid and Solid Mechanics, Cambridge, MA, June 17-20, 2003.

[12] George Karypis and Vipin Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on scientific Computing, 20 (1998), pp. 359–392.

[13] IV Lashuk and PS Vassilevski, *Element agglomeration coarse raviart–thomas spaces with improved approximation properties*, Numerical Linear Algebra with Applications, 19 (2012), pp. 414–426.

[14] Ilya Lashuk and Panayot S Vassilevski, *The construction of the coarse de rham complexes with improved approximation properties.*, Comput. Meth. in Appl. Math., 14 (2014), pp. 257–303.

[15] Dimitri J. Mavriplis, *Multigrid approaches to non-linear diffusion problems on unstructured meshes*, Numerical Linear Algebra with Applications, 8 (2001), pp. 499–512.

[16] Dimitri J. Mavriplis, *An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers*, Journal of Computational Physics, 175 (2002), pp. 302 – 325.

[17] D. J. Mavriplis and D. J. Mavriplis, *Multigrid techniques for unstructured meshes*, tech. report, in VKI Lecture Series VKI-LS, 1995.

[18] NETGEN, *Netgen - automatic mesh generator*. http://www.hpfem.jku.at/netgen/.

[19] Joseph E. Pasciak and Panayot S. Vassilevski, *Exact de rham sequences of spaces defined on macro-elements in two and three spatial dimensions.*, SIAM J. Scientific Computing, 30 (2008), pp. 2427–2446.

[20] J. W. Ruge and K. Stuben, *Algebraic multigrid (amg)*, In S. F. McCormick, editor, Multigrid Methods, volume 3 of Frontiers in Applied Mathematics, pages 73-130. SIAM, Philadelphia, PA,, (1987).

[21] Joachim Schöberl, *Netgen an advancing front 2d/3d-mesh generator based on abstract rules*, Computing and Visualization in Science, 1, pp. 41–52.

[22] Ulrich Trottenberg, Cornelius W. Oosterlee, and Anton Schuller, *Multigrid*, Academic Press, 2000.

[23] Panayot S. Vassilevski, *Sparse matrix element topology with application to amg(e) and preconditioning*, Numerical Linear Algebra with Applications, 9 (2002), pp. 429–444.