Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

Wegelerstraße 6 • 53115 Bonn • Germany
phone +49 228 73-3427 • fax +49 228 73-7527
www.ins.uni-bonn.de

P. Zaspel

Subspace correction methods in algebraic multi-level frames

INS Preprint No. 1510

June 2015

Subspace correction methods in algebraic multi-level frames

Peter Zaspel

Institute for Numerical Simulation, University of Bonn, Wegelerstraße 6, 53115 Bonn

Abstract

This study aims at introducing new algebraic multi-level solution techniques for linear systems with M-matrices. Previous optimal geometric constructions by multi-level generating systems or multi-level frames are adapted. The new contribution is a purely algebraic construction of multi-level frames. A new class of algebraic multi-level algorithms is derived by applying subspace correction iterative solvers to the algebraic multi-level linear system. These algorithms feature error resilience properties and potential massive parallelism. The proposed work outperforms previous geometric constructions since a black-box, geometry-independent methodology is considered. Moreover, optimality results of geometric constructions are matched. Overall, the new method will be well suited for generic linear algebra libraries for future multi- and many-core systems.

Keywords: Algebraic Multigrid, Subspace Correction, Iterative Solver, Multi-Level Frames

1. Introduction

The solution of large sparse linear systems is the dominant computational part of many solvers for partial differential equations (PDEs). Multigrid solvers are optimal problem-size independent iterative solvers for such systems. Nevertheless, multigrid solvers tend to have limited parallel scalability on extreme-scale high performance computing (HPC) clusters [1]. Moreover, next generation Exascale clusters are expected to have a growing number of (hardware) failures [2]. These issues require us to develop new optimal-complexity linear solvers that are scalable and resilient.

Preprint submitted to Linear Algebra and its Applications *June 11, 2015*

Randomized subspace correction iterative solvers for multi-level frames, cf. [3], give rise to mesh-width independent solvers for e.g. elliptic problems. The randomized, greedy-type solution algorithm from [3] is error-resilient by construction and might expose a high grade of parallelism by some suitable extension. However, it has been only discussed for model-type problems, since knowledge of the geometric structure of the discretized problem geometry is needed for the multi-level construction.

To easily solve problems on complex geometries, we introduce a purely algebraic multi-level construction approach. The multi-level construction is based on classical (Ruge-Stüben) algebraic multigrid (AMG). Projections between grid levels are replaced by algebraic prolongations and restrictions, while the different levels are generated by standard AMG coarsening. The solution of the resulting algebraic multi-level frame linear system by a Gauss-Seidel iterative method is equivalent to some algebraic multigrid method with Gauss-Seidel smoother. Moreover, we construct a new randomize greedy-type algebraic multi-level method by applying randomized greedy subspace correction techniques from [3] to the algebraic multi-level frame system.

Numerical results will be given, showing that the new method matches the geometry-dependent solver results from [3] for equivalent complex-geometry elliptic problems discretized by linear finite elements. The purely algebraic construction makes the proposed method an optimal candidate for general-purpose linear solver libraries. Resilience and potential massive parallelism might be influencal for future linear solver developments in Exascale computing.

In Section 2, we discuss related work. Section 3 reviews standard (geometric) multi-level frames with iterative solvers and subspace correction schemes. Section 4 introduces the new algebraic multi-level frames covering the necessary AMG background. Numerical results are given in Section 5. Section 6 concludes this study with a short outlook.

2. Related work

Theory and numerical treatment of the so-called *multi-level generating system* for (elliptic) PDEs has been first discussed in [4, 5] and a series on follow-up studies [3, 6, 7, 8]. In the generating system approach, a basis for a fixed Galerkin finite element discretization level is replaced by a generating system of multi-level basis functions. Equivalence of the application of standard iterative solvers to the resulting multi-level generating system and optimal-complexity multi-level solvers on single-level discretizations has been intensively discussed in [4, 5]. More recently, multi-level generating systems have been reconsidered as *multi-level frames* [9, 10], accounting for the close connection to wavelet-based techniques [11]. Many of these approaches can be also re-interpreted as additive or multiplicative Schwarz methods [3, 12].

The application of more sophisticated iterative solvers to multi-level frame or generating system discretizations naturally leads to new classes of multigrid/-level type algorithms. In [3, 13], the Gauss-Southwell method [14] has been applied in this context, giving rise to a new greedy-type multi-level method. The new method shows problem-size independent convergence with a high potential for robustness and problem-adaptivity. Moreover, a preceding randomized subspace selection process leads to a randomized iterative subspace correction technique with optimal convergence. This randomized iterative subspace correction technique has a close relationship to randomized (block) Kaczmarz iterative methods [15, 16] and randomized coordinate decent methods [17, 18, 19], which are known from large-scale optimization. Another related field are asynchronous iteration techniques [20]. These techniques provide a theoretical and practical framework for iterative methods that allow for the re-use of iteration results from several previous iterations. Indeed, they allow to define asynchronous update steps in a single iterative process. Therefore, more recent applications of asynchronous iterations are scalable and hardware-aware iterative solvers and multigrid smoothers [21, 22].

Randomized and asynchronous iterative methods promise to overcome *resilience* issues, which are expected to occur for HPC clusters of growing size [2]. While iterative methods are generally well-known to be robust for a certain class of hardware- or software-induced errors, resilient multigrid methods are still subject to current research [23, 24]. *Scalability* of multigrid methods and smoothers might also be improved by randomized and asynchronous iterative methods. Here, recent studies, such as [1], suggest a scalability bottleneck for standard smoothers on large-scale HPC systems. In addition, *problem-adapted* multigrid techniques such as semi-coarsening [25] and line smoothers [26] have been proposed before. Greedy-type approaches should improve over such existing methodologies.

The algebraic or matrix-dependent construction of multi-level or multigrid methods is attractive for optimal-complexity black-box linear algebra libraries. A review of these methods is given in [27]. More recently, two techniques, namely classical Ruge-Stüben AMG [28] with further developments e.g. in [29, 30] and (smoothed) aggregation-based AMG [31] are in main use. While classical AMG constructs hierarchies of variables by *reusing* a subset of the variables from a fine level on the next coarser level, aggregation-based techniques construct new variables on a coarser level by *replacing* a set of fine grid variables. Aggregation-based techniques tend to be easier to implement and parallelize. On the other hand, classical AMG is more robust.

To the best of our knowledge, the only known connection between AMG and frame-based constructions has been discussed in the partially unpublished work [32, 33, 34], where so-called AMGlets, thus algebraically constructed wavelets, are considered. AMGlets primarily aim at generalizing basis function constructions for some classes of differential operators, while the present work focuses on geometry-independent optimal solvers. Greedy-based techniques have been discussed for AMG with respect to the construction of the multigrid hierarchy [35], but not for smoothers, so far. Therefore, we argue that the introduction of an AMG-based multi-level frame technique is new. Moreover, by transferring randomized subspace correction methods to AMG, we introduce a new class of resilient, scalable and problem-adapted algebraic multi-level methods.

3. Multi-level frame systems and their iterative solution

This section introduces the *original* (geometric) multi-level frame construction. A short review of multi-level frames for the solution of elliptic PDEs is given. Some knowledge of the structure of the resulting linear system is collected. Finally, properties of iterative linear solvers for multi-level frame systems are summarized.

3.1. Multi-level frames for the solution of elliptic PDEs

We closely follow [10, 3] for the introduction of multi-level frame discretizations of elliptic PDEs.

For a separable Hilbert space $\mathcal{H}$, its dual space $\mathcal{H}'$ and the duality product $< \cdot, \cdot >$ on $(\mathcal{H}, \mathcal{H}')$, we can introduce a *frame* for $\mathcal{H}$ as the countable collection $\Phi = \{\phi_i : i \in \Delta\} \subset \mathcal{H}$ with

$$C\|f\|^2_{\mathcal{H}'} \leq \sum_{i\in\Delta} |\langle f, \phi_i \rangle|^2 \leq D\|f\|^2_{\mathcal{H}'} \qquad \forall f \in \mathcal{H}'$$

Let $\Omega \subset \mathbb{R}^{\{2,3\}}$ be a piecewise linearly bounded domain. As a model problem, we aim at solving the elliptic partial differential equation

$$\begin{aligned} -\Delta u &= g \quad \text{in}\,\Omega \\ u &= 0 \quad \text{on}\,\partial\Omega \end{aligned}$$

by a Galerkin approach. We therefore choose $\mathcal{H} = H_0^1(\Omega)$ and introduce the usual bilinear form

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, dx, \quad a : \mathcal{H} \times \mathcal{H} \to \mathcal{H}'$$

and the corresponding linear functional

$$F(v) = \int_\Omega gv \, dx, \quad F \in \mathcal{H}' \, .$$

For a standard linear finite element discretization, we introduce an appropriate triangulation or partition $\mathcal{T}_l$, $l \geq 0$ of Ω with element sizes $h \approx 2^{-l}$. Then,

$$V_l := \text{span}\{\varphi_{l,k} | k \in \Delta_l\}$$

is the finite-dimensional subspace of linear Lagrange finite element basis functions over $\mathcal{T}_l$. Finally, we have to solve the linear system

$$A_l \vec{u}_l = \vec{f}_l$$

with

$$A_l = (a_{l;k,k'})_{k,k'}, \quad a_{l;k,k'} := a(\varphi_{l,k}, \varphi_{l,k'}) \quad \text{and} \quad f_{l;k} = F(\varphi_{l,k})$$

in order to get an approximation for the given PDE in V_l.

For a *multi-level frame* approximation, we instead introduce the *series* of nested subspaces

$$V_0 \subset V_1 \subset \ldots \subset V_l \subset \ldots \subset L^2(\Omega)$$

for nested triangulations. For geometric multi-level frames, the coarsest subspace V_0 is usually expected to resolve the boundary exactly, which might be a strong limitation.

The objective of multi-level frames is the approximation of the PDE as in a multi-resolution analysis. That is, we introduce the (countable) collection

$$\Phi = \{\varphi_{l,k} | k \in \Delta_l, l \in \mathbb{N}_0\} .$$

Following [10, Theorem 5], this collection defines a frame in $H^1(\Omega)$. By replacing the standard finite element basis with the (finite) collection or *generating system*

$$\Phi_L = \{\varphi_{l,k} | k \in \Delta_l, l \in \{0, \ldots, L\}\}$$

of maximum level L, cf. [3], we are able to introduce a multi-level discretization on level L, with a new linear system

$$\bar{A}_L \vec{\vec{u}}_L = \vec{\vec{f}}_L . \tag{1}$$

Here, the system matrix is composed of matrix blocks as

$$\bar{A}_L = (A_{l,l'})_{1 \leq l,l' \leq L} , \quad A_{l,l'} = \left(a_{l,l';k,k'}\right)_{k \in \Delta_l, k' \in \Delta_{l'}} , \quad a_{l,l';k,k'} = a(\varphi_{l,k}, \varphi_{l',k'})$$

and the right-hand side is given as

$$\vec{\vec{f}}_L = \left(\vec{f}_l\right)_{1 \leq l \leq L} , \quad \vec{f}_l = (f_{l;k})_{k \in \Delta_l}, \quad f_{l;k} = F(\varphi_{l,k}) .$$

By construction, the matrix of the multi-level frame system (1) is singular. However, cf. [10], the right-hand side lies in the image of the multi-level system matrix. For such problems, Krylov subspace solvers converge to a (non-unique) solution vector $\vec{\vec{u}}^L$. The final unique solution on level L is constructed by projection on the solution space V_L.

3.2. Structure of the multi-level system

In [4, 13], we learned that the multi-level system (1) has some specific structure. We can easily rewrite it in terms of the standard finite element stiffness matrix on level L and appropriate prolongation and restriction operators P_l and R_l that transfer functions between the different approximation spaces V_l as

$$P_l : V_l \to V_{l+1} \quad \text{and} \quad R_l : V_l \to V_{l-1} .$$

In the following, these linear operators are understood as matrices with respect to the appropriate basis or generating system. For the standard multilevel frame construction, P_l and R_l are chosen as in an appropriate standard geometric multigrid method. We also use the more general notation $P_l^{l+1} := P_l$ and $P_l^{l-1} := R_l$.

The reformulation of the multi-level problem (1) further involves to introduce the transfer matrices

$$S_L := \begin{bmatrix} P_L^0 \\ P_L^1 \\ \vdots \\ P_L^{L-1} \\ P_L^L \end{bmatrix}, \quad S^L := \left[P_0^L P_1^L \dots P_{L-1}^L P_L^L \right]$$

with concatenated restriction operators $P_L^l = P_L^{L-1} \cdot \ldots \cdot P_{l+1}^l$ and $P_L^L = I_L$ the identity matrix on level L. We can then rewrite (1) as

$$\bar{A}_L \vec{\bar{u}}_L = S_L A_L S^L \vec{\bar{u}}^L = S_L \vec{f}_L = \vec{\bar{f}}_L \, . \tag{2}$$

A solution on level L can be derived from the multi-level solution $\vec{\bar{u}}_L$ by

$$\vec{u}_L = S^L \vec{\bar{u}}_L \, .$$

3.3. Iterative solvers on multi-level frame systems

Following [5], the iterative solution of the multi-level system (1) for a maximum level of L corresponds to the solution of the standard single-level finite element problem on level L by a sophisticated, typically optimally preconditioned, iterative solver.

Solving the system (1) by a Jacobi-preconditioned conjugate gradient iterative solver corresponds to a BPX-preconditioned CG solver for the standard problem. Furthermore, the standard Gauss-Seidel iteration for the linear system $A\vec{x} = \vec{b}$, $A \in \mathbb{R}^{N \times N}$, which is given in Algorithm 1, is equivalent to some standard geometric multigrid V-cycle with Gauss-Seidel smoother [5, 3].

In [14], a greedy version of the Gauss-Seidel method was introduced. It is called *Gauss-Southwell* method. For a given iterate $\vec{x}^{(n')}$, it computes the residual $\vec{r}^{(n')} = \vec{b} - A\vec{x}^{(n')}$. Here, it picks the variable index $i^\star$ with maximum absolute residual

$$i^\star := \underset{i \in \{1,\dots,N\}}{\operatorname{argmax}} \left| r_i^{(n')} \right| \, .$$

Algorithm 1 Gauss-Seidel method

```
Require: $A \in \mathbb{R}^{N\times N}$, $\vec{b} \in \mathbb{R}^N$, $\vec{x}_{init} \in \mathbb{R}^N$, $N_{iter}$
function GaussSeidel
    $\vec{x}^{(0)} := \vec{x}_{init}$
    for $n \in \{0, \dots, N_{iter} - 1\}$ do
        for $i \in \{1, \dots, N\}$ do
            $\vec{x}_i^{(n+1)} := \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(n+1)} - \sum_{j=i+1}^{N} a_{ij} x_j^{(n)} \right)$
    return $\vec{x}^{(N_{iter})}$
```

Algorithm 2 Gauss-Southwell method

```
Require: $A \in \mathbb{R}^{N\times N}$, $\vec{b} \in \mathbb{R}^N$, $\vec{x}_{init} \in \mathbb{R}^N$, $N_{iter}$
function GaussSouthwell
    $\vec{x}^{(0)} := \vec{x}_{init}$
    for $n' \in \{0, \dots, NN_{iter} - 1\}$ do
        $\vec{r}^{(n')} := \vec{b} - A\vec{x}^{(n')}$
        $i^\star := \underset{i \in \{1,\dots,N\}}{\operatorname{argmax}} \left| r_i^{(n')} \right|$
        $\vec{x}_{i^\star}^{(n'+1)} := \frac{1}{a_{i^\star i^\star}} \left( b_{i^\star} - \sum_{j=1, j\neq i^\star}^{i^\star-1} a_{i^\star j} x_j^{(n')} \right)$
    return $\vec{x}^{(NN_{iter})}$
```

Then, a correction step is applied such that we have a zero residual for variable $x_{i^\star}^{(n'+1)}$ in the next iteration. In [13], an up to $O(\log N)$ complexity approach for this update step is given. A total of N of these one-variable-update steps is usually considered as one iteration, cf. Algorithm 2. In [3] and [13], it is shown that an iterative solution of the multi-level frame system by the Gauss-Southwell method is at least as fast as the Gauss-Seidel method, in terms of iterations. In many cases, the Gauss-Southwell algorithm clearly outperforms the Gauss-Seidel method.

Another approach discussed in [3] is a modification of the Gauss-Southwell method. The modified approach *randomly* picks a new variable index instead of optimizing over all variables. We call this approach *random Gauss-Seidel* method and assume an equal distribution over all variables. Numerical results in [3] show similar (optimal) solution performance for the multi-level frame system, in terms of asymptotic complexity, but a larger constant. This limitation is removed by an alternative approach discussed in [3], which we

Algorithm 3 k-random block-Gauss-Seidel method

```
Require: $A \in \mathbb{R}^{N\times N}$, $\vec{b} \in \mathbb{R}^N$, $\vec{x}_{init} \in \mathbb{R}^N$, $N_{iter}$, $k$
function kRandomBlockGaussSeidel
    $\vec{x}^{(0)} := \vec{x}_{init}$
    for $n' \in \{0, \ldots, NN_{iter} - 1\}$ do
        $\vec{r}^{(n')} := \vec{b} - A\vec{x}^{(n')}$
        $I :=$ uniformRandomSubset $(k, \{1, \ldots, N\})$
        $i^\star := \underset{i \in I}{\mathrm{argmax}} \left| r_i^{(n')} \right|$
        $\vec{x}_{i^\star}^{(n'+1)} := \frac{1}{a_{i^\star i^\star}} \left( b_{i^\star} - \sum_{j=1, j\neq i^\star}^{i^\star - 1} a_{i^\star j} x_j^{(n')} \right)$
    return $\vec{x}^{(NN_{iter})}$
```

call *k-random block-Gauss-Seidel* method. It randomly picks k variables and performs the greedy Gauss-Southwell-type correction on that variable subset, cf. Algorithm 3. In [3], a numerical result for an elliptic problem suggests that the k-random block-Gauss-Seidel method ouptperforms the standard Gauss-Seidel method for $k \geq 3$. For $k = 1$, this method is identical to the random Gauss-Seidel method.

4. Algebraic multi-level frames

In our new purely algebraic approach, we aim at solving general linear systems of type

$$A\vec{x} = \vec{b}$$

with $A \in \mathbb{R}^{N\times N}$ and $\vec{x}, \vec{b} \in \mathbb{R}^N$ by a multi-level method. We further require A to be an M-matrix, to be able to apply classical AMG in its original form. Our aim is then to introduce an algebraically constructed variable hierarchy $\mathcal{D}_0 \subset \mathcal{D}_1 \subset \ldots \subset \mathcal{D}_L$ with $\mathcal{D}_L := \{1, \ldots, N\}$. This hierarchy and corresponding interpolation and restriction transfer operators will give rise to the algebraic multi-level system. It will be based on the structure considerations from Section 3.2.

4.1. Algebraic coarsening and transfer operators

As motivated before, we use coarse level selection techniques and prolongation/restriction operators from classical Ruge-Stüben AMG. Therefore, we give a brief review of the necessary basic facts from algebraic multigrid, cf. [36, Appendix A].

In AMG, a multigrid hierarchy is constructed from the structure and entries of the underlying system matrix. Geometric properties are ignored. Notation usually follows graph theory by identifying variables as graph nodes or points and non-zero non-diagonal entries in the system matrix as weighted edges between these nodes.

Let $A \in \mathbb{R}^{N\times N}$ be a given M-matrix that is a stiffness or system matrix on (the finest) level L. Each variable then corresponds to one index in an index set $\mathcal{D}_L := \{1, \ldots, N\}$. Variables on coarser levels of the (algebraic) multigrid hierarchy are collected in subsets $\mathcal{D}_l$ with $\mathcal{D}_0 \subset \mathcal{D}_1 \subset \ldots \subset \mathcal{D}_L$. Classical Ruge-Stüben AMG classifies variables on each level l into disjoint sets of coarse grid variables C^l and fine grid variables F^l, such that $\mathcal{D}_l = C^l \mathbin{\dot\cup} F^l$. The coarse grid variables are reused on the next coarser level, i.e. $\mathcal{D}_{l-1} := C^l$.

To formulate an algorithm for the choice of fine and coarse grid points, – we call this choice *C/F splitting* – we need some further notation. The neighborhood of a variable $i \in \mathcal{D}_l$ is given by

$$\mathcal{N}_i^l := \left\{ j \in \mathcal{D}^l \,\middle|\, j \neq i, a_{ij}^l \neq 0 \right\} ,$$

where the a_{ij}^l are the matrix entries of the (subsequently constructed) coarse grid operator on level l. A variable i is *strongly negatively coupled* to a variable j if we have the relation

$$-a_{ij}^l \geq \varepsilon_{str} \max_k |a_{ik}^l|$$

for a given, fixed $0 < \varepsilon_{str} < 1$. We collect these variables in

$$S_i^l := \left\{ j \in \mathcal{N}_i^l \,\middle|\, i \text{ strongly negatively coupled to } j \right\}$$

and further define ${S_i^l}^\top := \left\{ j \in \mathcal{D}_l \,\middle|\, i \in S_j^l \right\}$. Algorithm 4 then summarizes the standard coarsening or C/F splitting algorithm of Ruge-Stüben AMG, cf. [36, Appendix A].

Next, we consider the algebraic construction of prolongation and restriction operators. In difference to the geometric case, prolongation operators $\mathcal{P}_l^{l+1} \in \mathbb{R}^{|\mathcal{D}_{l+1}|\times|\mathcal{D}_l|}$ from algebraic multigrid are matrices by definition. As usual, we further assume $\mathcal{P}_{l+1}^l = {\mathcal{P}_l^{l+1}}^\top$, that is, restriction matrices are constructed from given prolongation matrices. Therefore, it is enough just to

Algorithm 4 Standard coarsening algorithm

```
Require: level $l$, $\mathcal{D}_l$, $S^l$, $S^{l^\top}$
function AMGstandardCoarsening
    $F^l := \emptyset, C^l := \emptyset, U^l := \mathcal{D}_l$
    for $i \in U^l$ do
        $\lambda_i^l := \left| S_i^{l^\top} \cap U^l \right| + 2 \left| S_i^{l^\top} \cap F^l \right|$
    while $\exists i$ s.th. $\lambda_i^l \neq 0$ do
        find $i_{\max} := \operatorname{argmax}_i \lambda_i^l$
        $C^l := C^l \cup \{i_{\max}\}$
        $U^l := U^l \setminus \{i_{\max}\}$
        for $j \in \left( S_i^{l^\top} \cap U^l \right)$ do
            $F^l := F^l \cup \{j\}$
            $U^l := U^l \setminus \{j\}$
        for $i \in U^l$ do
            $\lambda_i^l := \left| S_i^{l^\top} \cap U^l \right| + 2 \left| S_i^{l^\top} \cap F^l \right|$
    return $C^l, F^l$
```

define prolongation, which is also called *interpolation* in AMG. Here, we have further notation

$$C_i^l := C^l \cap \mathcal{N}_i^l, \quad F_i^l := F^i \cap \mathcal{N}_i^l, \quad \tilde{C}_i^l := C^l \cap S_i^l, \quad \tilde{F}_i^l := F^l \cap S_i^l\,.$$

For the so-called *direct interpolation*, interpolation matrices are constructed as follows: Coarse grid variable are identically transferred form coarse to fine grid. However, fine grid variables $e_i^l, i \in \mathcal{F}^l$ have to be interpolated appropriately. This is done using the set of interpolatory variables $I_i^l := \tilde{C}_i^l$ that are all strongly connected coarse grid variables. The interpolation rule of direct interpolation is then given by

$$e_i^l = \sum_{j \in I_i^l} w_{ij}^l e_j^l, \quad w_{ij}^l = -\alpha_i^l \frac{a_{ij}^l}{a_{ii}^l}, \quad \alpha_i^l = \frac{\sum_{k \in \mathcal{N}_i^l} a_{ik}^l}{\sum_{j \in I_i^l} a_{ij}^l}\,.$$

Standard interpolation additionally considers strong connections between fine grid nodes. To this end, the original (coarse grid) operator matrix A_l is expanded to a matrix $\hat{A}_l$, such that we replace for each given fine grid variable

$i \in F^l$, the variable $j \in \tilde{F}_i^l$ as

$$e_j \longrightarrow \sum_{k \in \mathcal{N}_j^l} a_{jk}^l e_k^l / a_{jj}^l \,.$$

The new set of interpolatory variables is $\hat{I}_i^l = \tilde{C}_i^l \cup (\bigcup_{j \in \tilde{F}_i^l} \tilde{C}_j^l)$. Standard interpolation is finally constructed by applying direct interpolation to the extended matrix $\hat{A}_l$.

In some cases, an additional Jacobi smoothing step is executed on the system matrix, leading to the so-called *Jacobi interpolation*, cf. [36, Appendix A]. Usually, *truncation* is applied to interpolation or prolongation matrices, in order to reduce the number of non-zero entries of the system matrix on the next coarser level. Truncation removes matrix entries beyond a relative threshold of ε_{tr}.

4.2. Algebraic multi-level system

Let us remember our intention to solve the linear system

$$A\vec{x} = \vec{b}\,, \tag{3}$$

with $A \in \mathbb{R}^{N \times N}$ an M-matrix and $\vec{b} \in \mathbb{R}^N$ an appropriate right-hand side. We can now use Algorithm 4 to introduce a hierarchy $\mathcal{D}_0 \subset \mathcal{D}_1 \subset \ldots \subset \mathcal{D}_L$ of variable indices, with $\mathcal{D}_L = \{1, \ldots, N\}$ the variable index set of the original linear system (3) and

$$\mathcal{D}_l = C^l \,\dot\cup\, F^l, \quad \mathcal{D}_{l-1} := C^l \quad \forall l \in \{1, \ldots, L\}$$

the splitting into coarse and fine grid variables as in AMG. We furthermore introduce the same interpolation or prolongation operators $\mathcal{P}_l^{l+1} \in \mathbb{R}^{|\mathcal{D}_{l+1}| \times |\mathcal{D}_l|}$ and restriction operators $\mathcal{P}_{l+1}^l$ as in AMG. Depending on the given linear system, we might use direct interpolation, standard interpolation, Jacobi interpolation or combinations of these. We now construct an *algebraic multi-level system* using algebraic transfer matrices

$$\mathcal{S}_L := \begin{bmatrix} \mathcal{P}_L^0 \\ \mathcal{P}_L^1 \\ \vdots \\ \mathcal{P}_L^{L-1} \\ \mathcal{P}_L^L \end{bmatrix}, \quad \mathcal{S}^L := \left[\mathcal{P}_0^L \mathcal{P}_1^L \ldots \mathcal{P}_{L-1}^L \mathcal{P}_L^L\right] \,.$$

The algebraic multi-level system is then given by

$$\mathcal{S}_L A \mathcal{S}^L \vec{\vec{x}} =: \quad \bar{A}\vec{\vec{x}} = \vec{\vec{b}} \quad := \mathcal{S}_L \vec{b}\,, \tag{4}$$

with

$$\bar{A} \in \mathbb{R}^{\bar{N}\times\bar{N}}, \quad \vec{\vec{x}}, \vec{\vec{b}} \in \mathbb{R}^{\bar{N}}, \quad \bar{N} = \sum_{l=0}^{L} |\mathcal{D}_l|\,.$$

As in the standard multi-level frame case, this system is not uniquely solvable. However, a non-unique solution $\vec{\vec{x}}$ can be projected back to a unique solution of the original system (3) by

$$\vec{x} = \mathcal{S}^L \vec{\vec{x}}_L\,.$$

4.3. Performance and complexity considerations

By construction, the algebraic multi-level linear system is larger than the original linear system (3). This increases the complexity of linear solvers that are applied to the multi-level system, if we express the complexity in terms of the number of unknowns N of (3). However, at the same time, optimized greedy-type solvers and improved scalability in parallelizations are expected to improve the measured runtime a lot. Therefore, a clear overall runtime improvement over standard AMG is expected for a highly optimized parallel algebraic multi-level frame solver.

Numerical results in this article reflect a feasibility study of the proposed approach. Therefore runtime comparisons are considered future work. Moreover, it might be possible to reduce the overall computational complexity and runtime by new on-the-fly construction and compression techniques. This is also future work.

5. Numerical results

We study the solution properties of the iterative methods from Section 3.3 applied to the new algebraic multi-level frame construction. The model problem considered here is the Poisson problem

$$\begin{aligned} -\Delta u &= 1 \quad \text{in}\,\Omega\,, \\ u &= 0 \quad \text{on}\,\partial\Omega \end{aligned}$$

on the complex geometry Ω shown in Figure 1. It is the unit square with five circular holes.

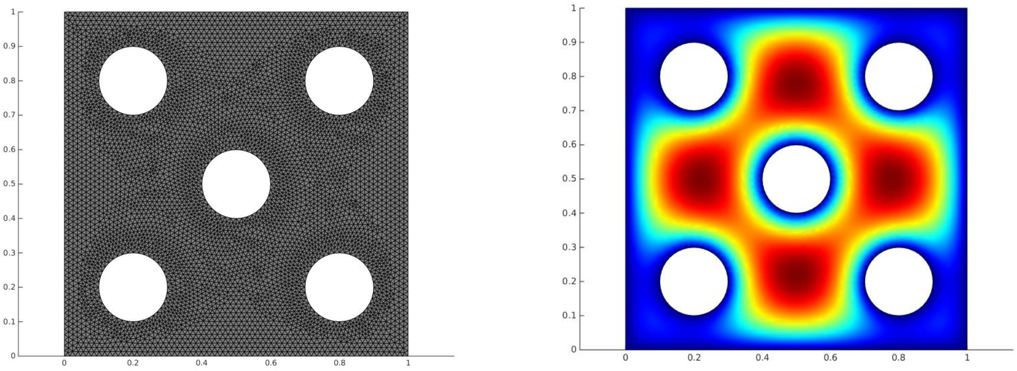

Figure 1: Visualization of the triangulated domain (left) and the solution field of the discussed model problem (right).

The model problem is discretized by first-order finite elements. For our convergence studies, we define resolution levels L such that the triangulation $\mathcal{T}_L$ has a maximum triangle size of $h_{max} = 2^{-L}$. Triangulation and stiffness matrix assembly is done by COMSOL. Figure 1 shows on the left-hand side a triangulation for level $L = 6$. Stiffness matrices and load vectors were extracted using the COMSOL LiveLink for MATLAB extension and the command `mphmatrix`. We use *eliminated* matrices and load vectors, thus potential null spaces in the original stiffness matrix were removed by COMSOL, beforehand. The algebraic multi-level system is constructed in MATLAB.

In our MATLAB-based numerical studies, we iteratively solve the algebraic multi-level frame system up to a normalized residual norm of 10^{-9}. Normalization is done with the norm of the right-hand side load vector. The initial guess for the solution is a random vector. C/F splitting with strength parameter $\varepsilon_{str} = 0.25$ and standard interpolation with an additional Jacobi interpolation step and truncation ($\varepsilon_{tr} = 0.25$) is applied. The algebraic multigrid hierarchy has been coarsened until a maximum of 10 coarse grid variables was left.

Remember that the model problem is given on a complex geometry. Here, a standard (non-algebraic) multi-level frame construction would require a

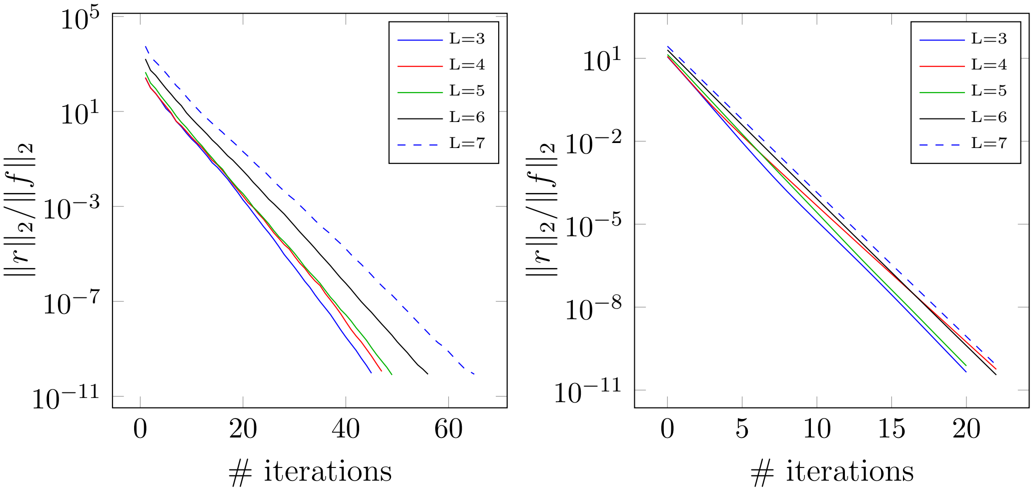

Figure 2: Relative residual decay for the solution of the algebraic multi-level linear system by a Jacobi-preconditioned CG method (left) and by the Gauss-Seidel method (right) for different resolution levels L.

considerable effort to construct special problem-dependent coarsening, prolongations and restrictions. It might be even necessary to exactly resolve the boundary on the coarsest grid. In contrast, our algebraic approach works without any modification.

Figure 2 shows on the left-hand side the residual decay for an iterative solution of the algebraic multi-level system by a Jacobi-preconditioned CG solver. This is equivalent to an algebraic-type BPX preconditioned solve for the standard finite element problem. A small, problem-size dependent increase in the number of iterations is visible. It is expected to stagnate for lager problems sizes. On the right-hand side of the same figure, the Gauss-Seidel iterative solver is used to solve the multi-level system. Here, optimal problem-independent convergence is shown. The method is equivalent to some algebraic multigrid method with Gauss-Seidel smoother. The Gauss-Seidel results match the results given in [3] for the geometric multi-level frame construction.

Results for the Gauss-Southwell iterative method, applied to the algebraic multi-level system, are given on the left-hand side of Figure 3. Remember that a single Gauss-Southwell step corrects a single variable. Therefore, we here denote $\bar{N}$ correction steps as a single iteration of the Gauss-Southwell method. Following [13], one such iteration can be computed with a computa-

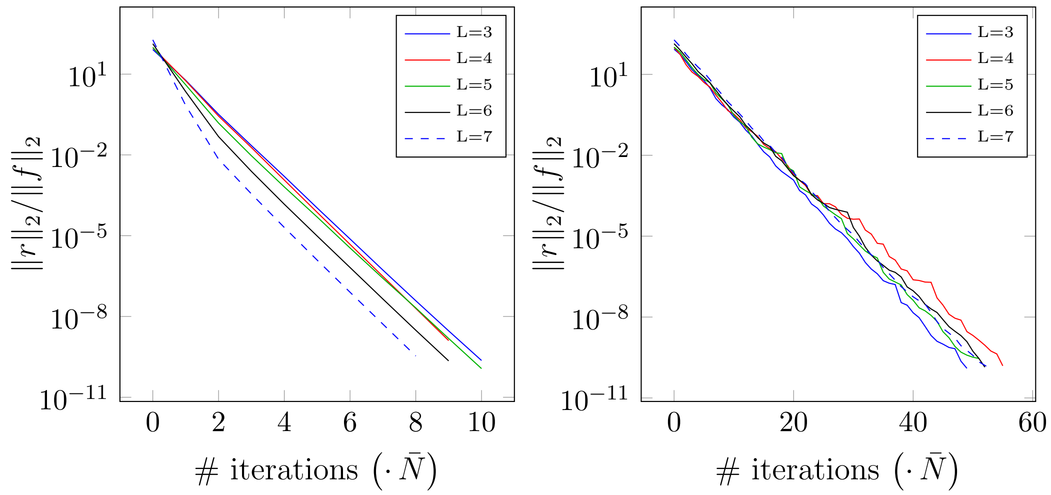

Figure 3: Relative residual decay for the solution of the algebraic multi-level linear system by the Gauss-Southwell method (left) and by the random Gauss-Seidel method (right) for different resolution levels L.

tional complexity of $O(\bar{N} \log \bar{N})$. Optimal problem-independent convergence is achieved. Moreover, the number of iterations is only half of the number of Gauss-Seidel iterations.

The right-hand side of Figure 3 gives convergence results for the random Gauss-Seidel method with the same definition of an iteration as in the Gauss-Southwell case. Here again, problem-size independent convergence is achieved. Note that this method performs corrections steps in a purely random ordering. Therefore, this approach is resilient by construction. The random Gauss-Seidel method needs about five times the amount of iterations of the Gauss-Southwell method and about 2.5 times the number of iterations of the standard Gauss-Seidel method. This relation roughly holds across all levels. All results shown in Figure 3 match convergence results from [3] in a qualitative way.

Finally, we compare the k-random block-Gauss-Seidel method for different block sizes with the standard Gauss-Seidel method and the Gauss-Southwell method, cf. Figure 4. This is done for a fixed resolution level $L = 7$. Our results qualitatively match those in [3]. For $k \geq 3$, the k-random block-Gauss-Seidel method performs at least as good as the Gauss-Seidel method, i.e. an algebraic multigrid method with Gauss-Seidel smoother in the standard formulation. In case of growing k, the necessary number of

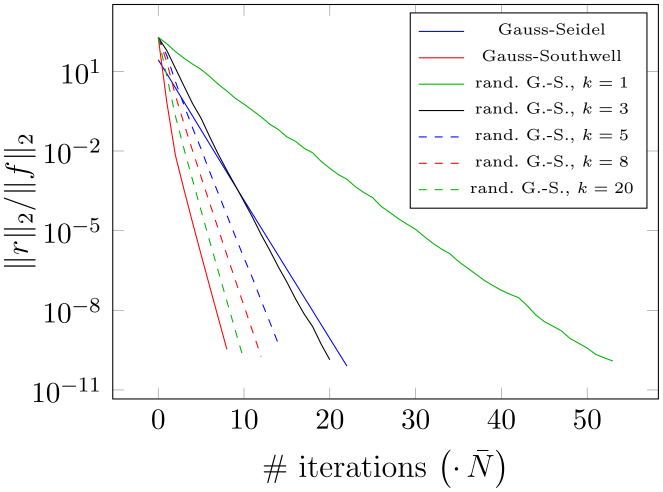

Figure 4: Comparison of the different relative residual decays for a complex geometry Poisson problem on level $L = 7$, solved by the Gauss-Seidel method, the Gauss-Southwell method and the k-random block-Gauss-Seidel method.

iterations seems to converge towards results of the Gauss-Southwell method.

6. Conclusions

We have introduced an algebraic multi-level frame construction. It transforms the multi-level discretization approach, known as multi-level frames, to a purely algebraic solver technique. In case of an elliptic model problem, subspace correction iterative methods applied to the algebraic multi-level frame system converge in a problem-independent way, i.e. achieve a similar performance as standard algebraic multigrid. Moreover, the proposed approach exposes structure that will make it an optimal candidate for error-resilience. Compared to the geometric construction shown in [3], we achieve (qualitatively) identical results, even on a complex geometry. Due to its algebraic nature, the proposed method is an optimal candidate for generic linear algebra libraries.

In the future, it is planned to discuss improvements for performance and computational complexity. New on-the-fly construction and compression techniques may reduce the runtime and computational complexity. At the same time, the extreme parallelism and error-resilience of the proposed method is expected to be exemplified by providing a multi-level algebraic frames library based on work in [13]. This library might to be able to outperform classical algebraic multigrid approaches on multi- and many-core architectures.

Acknowledgement

The author was partially supported by the project EXAHD of the DFG priority program 1648 Software for Exascale Computing (SPPEXA).

References

[1] A. Baker, R. Falgout, T. Gamblin, T. Kolev, M. Schulz, U. Yang, Scaling algebraic multigrid solvers: On the road to Exascale, in: C. Bischof, H.-G. Hegering, W. E. Nagel, G. Wittum (Eds.), Competence in High Performance Computing 2010, Springer Berlin Heidelberg, 2012, pp. 215–226.

[2] F. Cappello, A. Geist, W. Gropp, S. Kale, B. Kramer, M. Snir, Toward Exascale resilience: 2014 update, Supercomputing frontiers and innovations 1 (1).

[3] M. Griebel, P. Oswald, Greedy and randomized versions of the multiplicative Schwarz method, Linear Algebra and its Applications 7 (2012) 1596–1610.

[4] M. Griebel, Multilevelmethoden als Iterationsverfahren über Erzeugendensystemen, Teubner Skripten zur Numerik, Vieweg+Teubner Verlag, 1993.

[5] M. Griebel, Multilevel algorithms considered as iterative methods on semidefinite systems, SIAM Int. J. Sci. Stat. Comput. 15(3) (1994) 547–565.

[6] M. Griebel, P. Oswald, On additive Schwarz preconditioners for sparse grid discretizations, Numerische Mathematik 66 (1) (1993) 449–463.

[7] M. Griebel, P. Oswald, Tensor product type subspace splittings and multilevel iterative methods for anisotropic problems, Advances in Computational Mathematics 4 (1) (1995) 171–206.

[8] M. Griebel, A. Hullmann, P. Oswald, Optimal scaling parameters for sparse grid discretizations, Numerical Linear Algebra with Applications 22 (1) (2015) 76–100.

[9] S. Dahlke, M. Fornasier, T. Raasch, R. Stevenson, M. Werner, Adaptive frame methods for elliptic operator equations: The steepest descent approach, Adv. Comput. Math 27 (2004) 27–63.

[10] H. Harbrecht, R. Schneider, C. Schwab, Multilevel frames for sparse tensor product spaces, Numerische Mathematik 110 (2) (2008) 199–220.

[11] W. Dahmen, Wavelet and multiscale methods for operator equations, Acta Numerica 6 (1997) 55–228.

[12] M. Griebel, P. Oswald, On the abstract theory of additive and multiplicative Schwarz algorithms, Numerische Mathematik 70 (2) (1995) 163–180.

[13] R. Thesen, Effiziente adaptive Lösung von Multilevelsystemen, diploma thesis, Institute for Numerical Simulation, University of Bonn (aug. 2010).

[14] R. Southwell, Relaxation methods in engineering science - A treatise on approximate computation, Oxford Univ. Press, Oxford, 1940.

[15] A. Zouzias, N. M. Freris, Randomized extended Kaczmarz for solving least squares, SIAM Journal on Matrix Analysis and Applications 34 (2) (2013) 773–793.

[16] P. Oswald, W. Zhou, Convergence estimates for Kaczmarz-type methods, Lin. Algebra Appl., submitted.

[17] Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, SIAM Journal on Optimization 22 (2) (2012) 341–362.

[18] A. Beck, L. Tetruashvili, On the convergence of block coordinate descent type methods, SIAM Journal on Optimization 23 (4) (2013) 2037–2060.

[19] J. Mairal, Optimization with first-order surrogate functions, in: Proceedings of the International Conference on Machine Learning (ICML, 2013.

[20] A. Frommer, D. B. Szyld, On asynchronous iterations, Journal of Computational and Applied Mathematics 123 (12) (2000) 201 – 216, Numerical Analysis 2000. Vol. III: Linear Algebra.

[21] H. Anzt, S. Tomov, M. Gates, J. Dongarra, V. Heuveline, Block-asynchronous multigrid smoothers for GPU-accelerated systems, Procedia Computer Science 9 (0) (2012) 7 – 16, Proceedings of the International Conference on Computational Science, ICCS 2012.

[22] H. Anzt, S. Tomov, J. Dongarra, V. Heuveline, Weighted block-asynchronous iteration on GPU-accelerated systems, in: I. Caragiannis, M. Alexander, R. Badia, M. Cannataro, A. Costan, M. Danelutto, F. Desprez, B. Krammer, J. Sahuquillo, S. Scott, J. Weidendorfer (Eds.), Euro-Par 2012: Parallel Processing Workshops, Vol. 7640 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 145–154.

[23] M. Casas, B. R. de Supinski, G. Bronevetsky, M. Schulz, Fault resilience of the algebraic multi-grid solver, in: Proceedings of the 26th ACM International Conference on Supercomputing, ICS '12, ACM, New York, NY, USA, 2012, pp. 91–100.

[24] M. Huber, B. Gmeiner, U. Rüde, B. I. Wohlmuth, Resilience for Exascale enabled multigrid methods, CoRR abs/1501.07400.

[25] J. Dendy, Jr., M. Ida, J. Rutledge, A semicoarsening multigrid algorithm for SIMD machines, SIAM Journal on Scientific and Statistical Computing 13 (6) (1992) 1460–1469.

[26] C. Oosterlee, F. Gaspar, T. Washio, R. Wienands, Multigrid line smoothers for higher order upwind discretizations of convection-dominated problems, Journal of Computational Physics 139 (2) (1998) 274 – 307.

[27] K. Stüben, A review of algebraic multigrid, Journal of Computational and Applied Mathematics 128 (12) (2001) 281 – 309, Numerical Analysis 2000. Vol. VII: Partial Differential Equations.

[28] J. Ruge, K. Stüben, Algebraic multigrid (AMG), in: S. McCormick (Ed.), Multigrid Methods, Frontiers in Applied Mathematics, Vol. 5, SIAM, Philadelphia, 1986.

[29] M. Griebel, B. Metsch, D. Oeltz, M. A. Schweitzer, Coarse grid classification: A parallel coarsening scheme for algebraic multigrid methods, Numerical Linear Algebra with Applications 13 (2–3) (2006) 193–214.

[30] B. Metsch, Algebraic Multigrid (AMG) for Saddle Point Systems, PhD thesis, Institute for Numerical Simulation, University of Bonn (jul. 2013).

[31] P. Vank, J. Mandel, M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing 56 (3) (1996) 179–196.

[32] F. Kiefer, Multiskalen-Verfahren für Konvektions-Diffusions Probleme, PhD thesis, Institute for Numerical Simulation, University of Bonn (jul. 2001).

[33] M. Griebel, F. Kiefer, Generalized hierarchical basis multigrid methods for convection-diffusion problems, SFB Preprint 720, Sonderforschungsbereich 256, Institut für Angewandte Mathematik, Universität Bonn (2001).

[34] T. Gerstner, F. Kiefer, A. Kunoth, Wavelet and multigrid methods for convection–diffusion equations, in: H.-J. Neugebauer, C. Simmer (Eds.), Dynamics of Multiscale Earth Systems, Lecture Notes in Earth Sciences 97, Springer, 2003, pp. 123–134.

[35] S. MacLachlan, Y. Saad, A greedy strategy for coarse-grid selection, SIAM Journal on Scientific Computing 29 (5) (2007) 1825–1853.

[36] U. Trottenberg, A. Schuller, Multigrid, Academic Press, Inc., Orlando, FL, USA, 2001.