Function-based Algebraic Multigrid method for the 3D Poisson problem on structured meshes

Ali Dorostkar[1]

Department of Information Technology, Uppsala University, Sweden, ali.dorostkar@uu.se

Abstract. Multilevel methods, such as Geometric and Algebraic Multigrid, Algebraic Multilevel Iteration, Domain Decomposition-type methods have been shown to be the methods of choice for solving linear systems of equations, arising in many areas of Scientific Computing. The methods, in particular the multigrid methods, have been efficiently implemented in serial and parallel and are available via many scientific libraries.

The multigrid methods are primarily used as preconditioners for various Krylov subspace iteration methods. They exhibit convergence that is independent or nearly independent on the number of degrees of freedom and can be tuned to be also robust with respect to other problem parameters. The methods likewise possess optimal computational complexity. As a drawback, of particular importance when solving very large scale problems, we point out their high demand for computer memory. Since the methods utilize hierarchical structures, where the amount of computations decreases and that of communication increases, their parallel implementation might exhibit lesser scalability. Further, as the implementation usually relies on sparse matrix-vector multiplications, this may also decrease parallel performance for very large problems.

In this work we utilize a different framework to construct multigrid methods, based on an analytical function representation of the matrix, that may keep the amount of computation high and local, and may reduce significantly the memory requirements. The approach is particularly suitable for modern computer architectures. An implementation of the latter for the three-dimensional discrete Laplace operator is derived and implemented. The same function representation technology is used to construct smoothers of sparse approximate inverse type.

1 Introduction

In scientific computations, Poisson's or Laplace equation is among most studied, analysed and tested partial differential equations.

The general form of the equation is

$$\begin{aligned} &\mathcal{L}(u) = f \text{ in } \Omega \subset \mathbb{R}^d \\ &u\,|_{\Gamma_D} = g_0, \\ &\tfrac{\partial u}{\partial n}\,|_{\Gamma_N} = g_1, \end{aligned} \tag{1}$$

where $\Gamma_D \cup \Gamma_N = \partial\Omega,\ d \in \{1, 2, 3\}$.

Here $\mathcal{L}$ is the Laplace operator and it comes in the following most well-known forms:

$$\mathcal{L}(u) \equiv -\Delta u \qquad \text{(isotropic Laplace)} \tag{2}$$

$$\mathcal{L}(u) \equiv -\sum_{i=1}^{d} \varepsilon_i \frac{\partial^2 u}{\partial x_i^2} \qquad \text{(anisotropic Laplace)} \tag{3}$$

$$\mathcal{L}(u) \equiv -\nabla \cdot (k(\mathbf{x})\nabla u) \qquad \text{(variable coefficients Laplace)} \tag{4}$$

On one side we may see (1) as a simple benchmark problem, used to study the properties of various discretization methods, solution techniques and preconditioning strategies in the context of iterative solution methods. On the other side, the significance of this problem in various scientific disciplines is monumental.

The forms (2)-(4), appear in various applications such as the stationary heat transfer, the pressure equation in the reduced form of the Navier-Stokes equations, in elasto-static problems for torsion load. It is also used in potential flow analysis, in mechanical engineering, where both

stream function and velocity potential function are obeying the Laplace equation. Further, the solution for Airy's stress function in plane stress problems, appearing in modelling of bending of isotropic plates, turns out to be a combination of general solutions of Laplace and Poisson's equations. The 3D Glacial Isostatic Adjustment model includes the geo-potential that also obeys Poisson's equation. In biological tissues, the local electric field distribution can be determined by solving the Laplace equation, at least in the cases when the micro-structure of the tissue is not evolving.

The vast amount of application fields of the Laplace and Poisson's equations has driven the need to develop fast and reliable black-box solvers for the discrete versions of the equations. The topic is very well studied, however, we focus on a reason to seek yet a different approach to solve systems with the discrete Laplace operator.

The largest interest lies in the three-dimensional case where the degrees of freedom become too large to use either of the direct solution methods, Fast Fourier transform-based methods and such. Experience shows that various versions of multilevel and multigrid methods are among the most efficient solvers for the problem at hand.

Although in theory multigrid methods possess numerically and computationally optimal properties, in practice they exhibit some drawbacks. We consider the algebraic multigrid (AMG) as a reference. For very large problems, the construction of AMG may become prohibitive due to very high memory requirements, sometimes too large to fit in the distributed memory of a modern HPC cluster (cf. [25]).

To achieve numerical efficiency, AMG constructs a sequence of coarser levels, where it operates on matrices of decreasing size. Thus, the number of arithmetic operations per fine-level degree of freedom decreases and the distance to communicate over increases. AMG involves non-local communications when implemented in parallel distributed manner. This communication pattern has a negative impact on the overall performance, as shown in, for instance, [4].

Both factors are not in favour of the modern computer architectures, where computation is cheap and memory references and communication are expensive. This performance bottleneck makes the idea to use the AMG framework, based on a particular analytic function representation of the matrix, particularly plausible for 3D problems. The idea itself is not new. It has been introduced as early as in 1991 in [11]. The essence of the idea is the following. Consider the class of structured matrices, arising from a discretization of differential equations using regular quadrilateral or hexahedral meshes and local methods, such as finite elements (FE), finite differences (FD), finite volumes (FV), or isogeometric analysis (IgA). Under mild assumption, these matrices fall in the class of the so-called Generalized Locally Toeplitz (GLT) matrices and can be characterized by an analytical function, referred to as the symbol of the matrix, cf. [13]. Based on the symbol, one can construct all the ingredients of a multigrid method and since the matrix entries can be computed via the symbol, this approach opens the possibility to obtain a matrix-free MG, saving memory usage and increasing the computational intensity of the corresponding implementation.

The GLT-based MG technique has undergone developments in many directions. A short literature survey reveals the following achievements. In [12] a symbol-based MG framework is introduced and two-grid method convergence is shown. The tests are performed on Poisson's equation in 2D. In [6,7], the method is applied for the convection-diffusion-reaction equation, discretized in the IgA framework. It utilizes the Kronecker structure of B-splines. The theoretical results are in a general d-dimensional setting. The numerical tests are performed in Matlab for 1D and 2D problems. The work in [8] analyses a symbol-based MG for a second order two point boundary value problem, approximated by spline polynomials of third degree in 1D. The construction of an MG preconditioner for graph Laplacians is treated in [24]. Special attention is paid to projection operators, attempting to preserve the structural properties of the original matrices. Paper [9] provides an extensive summary of the GLT-MG approach in several general cases. Worth to mention (cf., e.g. [1]) is that, in contrast to the classical V-cycle (A)MG, V-cycle GLT-MG exhibits optimal convergence property too.

Possibly because the developed theory for the symmetric positive definite case is rather general and space-dimension independent, so far the case for constructing GLT-MG for the 3D Laplacian has not been completed and implemented. This is the topic of the current paper.

The contents of this report is structured as follows. In Section 2 we construct the symbol representation of the anisotropic Laplacian in 3D, discretized by standard conforming piecewise trilinear basis functions. In Section 3 we describe the implementation of the related GLT-AMG method. Further, we show in Section 4 that one can use the GLT technology to compute sparse approximate inverses of the level matrices, to be used as smoothers, or as preconditioners in the smoothing steps. Section 5 contains an analysis of computer resources demands for the GLT-AMG method and the possible savings in memory, together with increase in number of computations to better match the characteristics of multicore computer architectures or GPUs. In Section 6 some numerical illustrations of the GLT-AMG performance are shown, as well as a comparison with that of AGMG (cf. [20–22]).

2 Function representation of the anisotropic Laplace operator

We consider Poisson's equation in 3D with anisotropic Laplace operator,

$$-\varepsilon_1 \frac{\partial^2 u}{\partial x_1^2} - \varepsilon_2 \frac{\partial^2 u}{\partial x_2^2} - \varepsilon_3 \frac{\partial^2 u}{\partial x_3^2} = f \quad \text{in } \Omega \tag{5}$$

where $\Omega \subset \mathbb{R}^3$ and $\varepsilon_i, i = 1, 2, 3$ are positive constants. For simplicity, we assume that the coefficients are scaled appropriately and are less or equal to 1. For describing the straightforward application of the GLT framework we let Ω to be the unit cube, discretized with an equidistant structured mesh with stepsize h. Without loss of generality we assume that homogeneous Dirichlet boundary conditions are imposed on the whole boundary $\partial\Omega$ of Ω.

To discretize (5), we choose standard conforming trilinear FEM. The variational formulation is also very standard and is therefore omitted. The resulting algebraic linear system to solve is

$$A\mathbf{u} = \mathbf{f},$$

where A is sparse, symmetric and positive definite. For our purposes, it is important to analyse its structure and how it is viewed as a Toeplitz (or nearly Toeplitz) matrix. To this end, in order to derive an analytical function to be associated with A, we need to introduce some major concepts from the GLT theory.

2.1 GLT and symbols of matrices

We briefly introduce the notion of Toeplitz matrices and their symbols, for a more complete discussion see [13].

Denote by $f(\theta_1, \cdots, \theta_d)$ a d-variate complex-valued integrable function, defined over the domain $Q^d = [-\pi, \pi]^d, d \geq 1$. Denote by f_k the Fourier coefficients of f,

$$f_k = \frac{1}{(2\pi)^d} \int_{Q^d} f(\theta) e^{-i\,(k,\theta)}\, d\theta,\ k = (k_1, \cdots, k_d) \in \mathbb{Z}^d,\ i^2 = -1, \tag{6}$$

where $(k, \theta) = \sum_{j=1}^{d} k_j \theta_j$, $n = (n_1, \cdots, n_d)$, and $N(n) = n_1 \cdots n_d$. Following the multi-index notation in [26], with each f we can associate a sequence of Toeplitz matrices $\{T_n\}$, $T_n = \{f_{k-\ell}\}_{k,\ell=\mathbf{e}^T}^{n} \in \mathbb{C}^{N(n)\times N(n)}$, $\mathbf{e} = [1, 1, \cdots, 1] \in \mathbb{N}^d$.

The function f is referred to as the *generating function*, also known as the *symbol*, of T_n. Using a more compact notation we write $T_n = T_n(f)$. An evaluation of the symbol $f(\theta)$ over an equispaced grid gives an approximation of the eigenvalues of $T_n(f)$. Moreover, the error in the approximation of the eigenvalues approaches zero as $n \to \infty$.

A d-dimensional partial differential problem with constant coefficients, discretized on quadrilateral finite elements of degree p is a d-level block-valued Toeplitz matrix with blocks of size $s = p^d$. In case of problems where the coefficients are non-constant and continuous, the matrix sequence $T_n(f)$ is of Generalized Locally Toeplitz type with symbol f.

Remark 1. An important feature of the GLT sequences is that a matrix obtained by a sequence of algebraic operations on GLT matrices is itself a GLT matrix. The symbol of the new GLT matrix is a function obtained by applying the same algebraic operations on the corresponding symbols of the original matrices.

2.2 The symbol of the 3D anisotropic Laplacian

To construct the symbol for the discrete counterpart of (5), we interpret the matrix A as the summation of three matrices A_i of the same size. The matrix A_i corresponds to the case where $\varepsilon_i = 1,\ \varepsilon_j = 0\ j \neq i$. We remove the effect of the grid size by multiplying A with $h/9$, which gives a simpler symbol expression. By analysing the matrices A_i we see that they are of GLT type with symbols f_i, given in (7).

$$\begin{aligned} f_1(\theta_1,\theta_2,\theta_3) &= (4+2cos(\theta_1))(1+0.5cos(\theta_2))(2-2cos(\theta_3)),\\ f_2(\theta_1,\theta_2,\theta_3) &= (4+2cos(\theta_1))(2-2cos(\theta_2))(1+0.5cos(\theta_3)),\\ f_3(\theta_1,\theta_2,\theta_3) &= (2-2cos(\theta_1))(4+2cos(\theta_2))(1+0.5cos(\theta_3)). \end{aligned} \tag{7}$$

Hence, the matrix A is a GLT sequence $T_n(f)$ with the following expression,

$$f(\theta_1,\theta_2,\theta_3) = \frac{h}{9}\big(\varepsilon_1 f_1(\theta_1,\theta_2,\theta_3) + \varepsilon_2 f_2(\theta_1,\theta_2,\theta_3) + \varepsilon_3 f_3(\theta_1,\theta_2,\theta_3)\big). \tag{8}$$

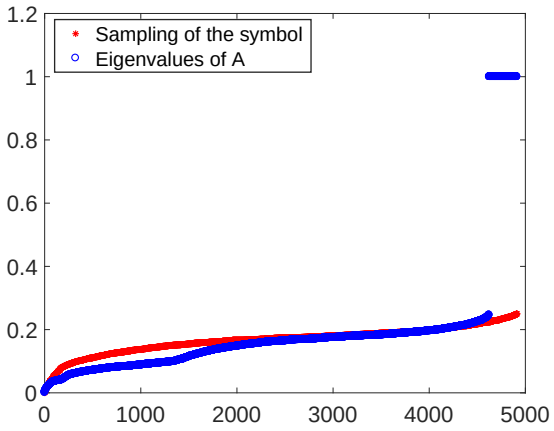

Fig. 1. The sorted equispaced sampling of the symbol f vs the true eigenvalues of A

Figure 1 illustrates the quality of the symbol f. We see that the equispaced sampling of f gives a good approximation of the eigenvalues of A up to a finite number of outliers. The outliers usually correspond to the effect of the boundary conditions on A (c.f. [13] for more details).

3 Construction of an Algebraic Multigrid method, based on the GLT function representation of a matrix

Simple iteration methods, such as Jacobi or Gauss-Seidel, usually dampen the error in the solution vector more rapidly in the eigenvector directions corresponding to the larger eigenvalues (high frequencies) of A (cf. [23]). In these methods, the components of the error corresponding to the lower frequencies are not reduced with the same speed as high frequencies. Hence, a better strategy is to work with a hierarchy of matrices, from coarse to the original dimensions, where the low

frequency errors are mapped to higher frequencies on the coarser meshes. On the coarse levels the error corresponding to these low frequencies can be smoothed more rapidly with simple, cheap iterative methods. In case when the hierarchy of the matrices is obtained from a hierarchy of grids, the method is called geometric multigrid (MG). On the other hand when the coarse level matrices are created purely algebraically, the method is known as the algebraic multigrid (AMG) (see, for instance, [23], for a detailed description of AMG).

There are a few components necessary for the construction of AMG , namely, prolongation and restriction operators which are used to move through the hierarchy, matrices on each coarse level, a solver for the coarse grid, and pre-/post-smoothers. The coarsest grid solver can be a direct or an iterative method used on the coarsest level in the AMG algorithm. Smoothers are cheap solution methods in charge of damping the error modes on each level.

The choice of prolongation and restriction operators for GLT sequences as well as theoretical justifications have already been studied in various articles (see [10] and the references therein).

The prolongation operator P_i is defined as the product between a Toeplitz matrix $T_{\mathbf{n}_i}(p_i)$ and the transpose of a k-level cutting matrix $H_{\mathbf{n}_i}$. Here, p_i is a k-variate first order trigonometric polynomial and the cutting matrix $H_{\mathbf{n}_i}$ is obtained as the Kronecker product of k 1-level cutting matrices $H^1_{n_i}$

$$H^1_{n_i} = \begin{bmatrix} 0 & 1 & 0 & & & \\ & 0 & 1 & 0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & 0 & 1 & 0 \end{bmatrix}, \quad H^1_{n_i} \in \mathbb{R}^{n_{i+1} \times n_i}.$$

Assuming that f_i has a zero of order $2q$ in θ^0_i, we choose p_i to be

$$p_i(\theta) = c \cdot \prod_{j=1}^{k} [1 + \cos(\theta_j - (\theta^0_i)_j)]^q. \tag{9}$$

We define the level matrices as $A_i = P^H_{i-1} A_{i-1} P_{i-1}$, $i = 1, \ldots, m$, $A_0 = A$. The restriction operator R_i is defined simply as the transpose of the prolongation operator P_i. The symbol f_i of the coarse level matrix A_i is of the form

$$f_i(\theta) = \frac{1}{2^k} \sum_{\eta \in \mathcal{C}(\theta)} p_i^2(\eta) f_i(\eta), \tag{10}$$

where $\mathcal{C}(\theta)$ is the set of the so-called corner points defined as $\mathcal{C}(\theta) = \{\eta | \eta_j \in \{\theta_j, \theta_j + \pi\}\}$.

Having defined coarse level matrices, prolongation, and restriction operators, we are ready with the construction of GLT-MG.

4 Choosing a smoother for the GLT-based AMG

Next, we introduce a method which, taking advantage of the GLT theory, provides a computationally cheap smoother for AMG.

The choice of the smoother for AMG is problem-dependent. The question how to choose it for anisotropic problems is discussed in the literature, for instance in [27].

Since A is a symmetric positive definite (spd) matrix in the anisotropic Poisson problem, we choose the preconditioned conjugate gradient (PCG) as a smoother. Further, we choose a sparse approximate inverse (SPAI) of A_i as the preconditioner for the smoother on the ith level.

SPAI is a branch of preconditioning techniques, where we construct $D \approx A^{-1}$. The preconditioned system is then $DAx = Db$. To use an approximation of A^{-1} as a preconditioner has the added benefit that the application of the preconditioner is reduced to a matrix-vector multiplication. The true inverse of a sparse matrix is usually dense. Thus, to overcome this limitation, a sparsity pattern is often prescribed, restricting the amount and placement of the non-zero elements of D. Constructing SPAI has been the topic in many research articles, starting with the work by

L. Kolotilina and co-workers ([19]). The original idea is to compute a high quality approximation of the exact inverse of a matrix with a prescribed sparsity structure which is "good" in some sense.

How to define a "good" sparsity structure for the approximate inverse of a general matrix remains an open problem. Intuitively we can think that good sparsity patterns are those that include "the most important", "dominant" part of the inverse, "the position of the largest entries" in the inverse (cf. [2,5,14–16,18,23]). However, a rigorous way to determine those may not be a feasible task for general matrices. Significant contributions in the area that have lead to practical algorithms, can be found in [3,16].

A notable SPAI preconditioner is the modified SPAI (MSPAI) ([16]) which is shown to perform well for various types of matrices. Moreover, the sparsity pattern of the approximate inverse can be both prescribed or dynamically constructed. Further, MSPAI is implemented and distributed as high performance parallel library ([17]). However, in this article, we have not performed any comparison with MSPAI.

For the considered class of problems we know that the entries in the true inverse of the discrete Laplacian exhibit fast decay away from the main diagonal and, therefore, a band structure of the approximate inverse is an appropriate choice.

Our idea to construct SPAI comes from the property of GLT sequences stated in Remark 1. We form a GLT sequence based on the following symbol,

$$h(\theta) = \frac{1}{f(\theta) + \gamma}, \quad G = T_n(h). \tag{11}$$

Here γ is a non-negative scalar which guarantees that $h(\theta)$ is bounded and integrable. We obtain the elements of the inverse by using (11) in (6). The sparsity pattern is set by prescribing a drop tolerance, a bandwidth or both.

After constructing G based on the symbol of the inverse, we can add an additional step to possibly improve its quality. We utilize the Frobenius norm minimization technique and compute a diagonal matrix D_G, such that $G + D_G$ minimizes the Frobenius norm

$$\|I - (G + D_G)A\|_W^2 .$$

In the numerical tests, W is the identity matrix.

5 Analysis of the required computer resources when using GLT-based AMG

The properties of a Toeplitz matrix make it suitable for parallel computations. From the definition of a Toeplitz matrix we know that the elements along descending diagonals are constant. Moreover, from (6) we are able to compute these elements directly from the symbol of the Toeplitz matrix. Thus, computing the first element of each diagonal is enough to construct the matrix or its approximate inverse.

Storing a Toeplitz matrix is flexible and can be performed in two ways.

- Compute the first element of each diagonal in each level of a d-level Toeplitz matrix and store it in an array. The memory consumption of the matrix is then of order $O(\prod_{k=1}^{d} b_k)$ where b_k is the chosen bandwidth for the kth level of the Toeplitz matrix.
- Avoid storing any entries and compute every element as soon as it is needed. In this way the method becomes matrix-free and everything in the algorithm is converted to computations. We note that there is no need for bookkeeping. That is, we don't need to store the positions of the non-zero elements are present. This is due to the fact that we impose a bandwidth on SPAI and hence for any needed element we know exactly if it should be computed or not.

Note that the computation of the elements should be performed at least once for the first storage strategy and many times in the case of the second, matrix free, method. To that end, the complexity of the numerical integration plays a role in the performance of the GLT-MG. Particularly for non-smooth, multidimensional symbols, the integration itself may make the second approach unusable.

6 Numerical experiments

In this section we provide a few numerical experiments to show the performance of the method. In the first case we show a GLT-MG method with one Gauss-Seidel iteration as its pre-/post-smoothing step. The prolongation and restriction operators are constructed based on the discussions in the previous section. The number of iterations and the final error in the solution is compared to AGMG. Table 1 provides the results. We see that GLT-MG is robust and optimal and performs fewer iterations than AGMG.

Table 1. GLTMG with GS as a smoother compared to AGMG

	$\varepsilon = (1,1,1)$				$\varepsilon = (1,1,10^{-4})$			
	GLT-MG$_{\text{GS}}$		AGMG		GLT-MG$_{\text{GS}}$		AGMG	
size	Iter	Error	Iter	Error	Iter	Error	Iter	Error
125	2	1.26e-09	1	2.67e-16	3	2.62e-06	1	1.10e-15
729	2	6.22e-09	7	8.70e-09	3	1.47e-06	11	4.33e-08
4913	2	1.06e-08	7	1.42e-08	3	7.11e-05	13	2.98e-07
35937	2	1.24e-08	11	8.09e-08	3	4.92e-05	13	1.35e-06
274625	2	1.30e-08	11	3.75e-07	3	3.96e-05	14	1.85e-06

Next, we replace the GS smoother with 10 iterations of PCG preconditioned with the symbol-based SPAI. Table 2 shows the number of iterations compared to AGMG. We see that although the number of iterations are growing with size, still our method needs less iterations than AGMG. The efficient implementation of the method in `C++` is under development.

Table 2. GLTMG + SPAI as smoother compared to AGMG; $\varepsilon = (1,1,1)$

Size	GLTMG	AGMG
125	2	1
729	3	7
4913	5	7

7 Conclusions

In this paper we construct an algebraic multigrid preconditioner based on the Generalized Locally Toeplitz theory and solve the three-dimensional anisotropic Poisson problem. The matrices arising from a finite element discretization on a structured grid belong to the GLT class of matrices. Taking advantage of the Toeplitz structure of the so-constructed matrices, we provide an algebraic multigrid preconditioner with a low memory footprint and high computational intensity. we design a sparse approximate inverse preconditioner for the coarse level matrices and use it in the pre-/post-smoothers. The construction of SPAI is completely parallelizable and the memory consumption is controllable. We show that our proposed AMG method is numerically robust and competitive with the AGMG method.

Acknowledgements

The author is grateful to Maya Neytcheva for providing constructive comments and insights and to Darina Neytcheva for helping with preparing the text.

References

1. A. Aricó, M. Donatelli, and S. Serra-Capizzano. V-cycle optimal convergence for certain (multilevel) structured linear systems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):186–214, 2004.
2. O. Axelsson. *Iterative solution methods*. Cambridge University Press, 1996.
3. M. Benzi and M. Tuma. A comparative study of sparse approximate inverse preconditioners. *Applied Numerical Mathematics*, 30(2):305–340, 1999.
4. J. R. Bull, A. Dorostkar, S. Holmgren, A. Kruichinina, M. Neytcheva, D. Nikitenko, N. Popova, P. Shvets, A. Teplov, V. Voevodin, and V. Voevodin. Multidimensional performance and scalability analysis for diverse applications based on system monitoring data. In *Parallel Processing and Applied Mathematics :*, Lecture Notes in Computer Science, 2017. to appear.
5. E. Chow. A priori sparsity patterns for parallel sparse approximate inverse preconditioners. *SIAM Journal on Scientific Computing*, 21(5):1804–1822, 2000.
6. M. Donatelli, C. Garoni, C. Manni, S. Serra-Capizzano, and H. Speleers. Robust and optimal multi-iterative techniques for iga galerkin linear systems. *Computer Methods in Applied Mechanics and Engineering*, 284(Supplement C):230 – 264, 2015. Isogeometric Analysis Special Issue.
7. M. Donatelli, C. Garoni, C. Manni, S. Serra-Capizzano, and H. Speleers. Symbol-based multigrid methods for galerkin b-spline isogeometric analysis. *SIAM Journal on Numerical Analysis*, 55(1):31–62, 2017.
8. M. Donatelli, M. Molteni, V. Pennati, and S. Serra-Capizzano. Multigrid methods for cubic spline solution of two point (and 2d) boundary value problems. *Applied Numerical Mathematics*, 104(Supplement C):15 – 29, 2016. Sixth International Conference on Numerical Analysis – Recent Approaches to Numerical Analysis: Theory, Methods and Applications (NumAn 2014).
9. M. Donatelli and S. Serra-Capizzano. Multigrid methods for (multilevel) structured matrices associated to a symbol and related applications. 2013.
10. M. Donatelli, S. Serra-Capizzano, and D. Sesana. Multigrid methods for Toeplitz linear systems with different size reduction. *BIT Numerical Mathematics*, 52(2):305–327, 2012.
11. G. Fiorentino and S. Serra-Capizzano. Multigrid methods for Toeplitz matrices. *Calcolo*, 28(3-4):283–305, 1991.
12. G. Fiorentino and S. Serra-Capizzano. Multigrid methods for symmetric positive definite block Toeplitz matrices with nonnegative generating functions. *SIAM Journal on Scientific Computing*, 17(5):1068–1081, 1996.
13. C. Garoni and S. Serra-Capizzano. Generalized locally Toeplitz sequences: theory and applications. *Springer (to appear). The preliminary (unpublished and incomplete) version of this book is available, under a different title, as Technical Report*, 23:2015–023, 2015.
14. M. J. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM Journal on Scientific Computing*, 18(3):838–853, 1997.
15. T. Huckle. Approximate sparsity patterns for the inverse of a matrix and preconditioning. *Applied Numerical Mathematics*, 30(2):291–303, 1999.
16. T. Huckle and A. Kallischko. Frobenius norm minimization and probing for preconditioning. *International Journal of Computer Mathematics*, 84(8):1225–1248, 2007.
17. T. Huckle, M. Sedlacek, and A. Kallischko. Mspai, `https://www5.in.tum.de/wiki/index.php/MSPAI`.
18. S. Kharchenko, L. Y. Kolotilina, A. Nikishin, and A. Y. Yeremin. A robust ainv-type method for constructing sparse approximate inverse preconditioners in factored form. *Numerical linear algebra with applications*, 8(3):165–179, 2001.
19. L. Y. Kolotilina and A. Y. Yeremin. Factorized sparse approximate inverse preconditionings i. theory. *SIAM Journal on Matrix Analysis and Applications*, 14(1):45–58, 1993.
20. A. Napov and Y. Notay. An algebraic multigrid method with guaranteed convergence rate. *SIAM Journal on Scientific Computing*, 34(2):A1079–A1109, 2012.
21. Y. Notay. An aggregation-based algebraic multigrid method. *Electronic Transactions on Numerical Analysis*, 37:123–146, 2010.
22. Y. Notay. Aggregation-based algebraic multigrid for convection-diffusion equations. *SIAM Journal on Scientific Computing*, 34(4):A2288–A2316, 2012.
23. Y. Saad. *Iterative methods for sparse linear systems*. Siam, 2003.
24. S. Serra-Capizzano and C. Tablino-Possio. A note on algebraic multigrid methods for the discrete weighted laplacian. *Computers & Mathematics with Applications*, 60(5):1290 – 1298, 2010.
25. E. Turan and P. Arbenz. Large scale micro finite element analysis of 3d bone poroelasticity. *Parallel Computing*, 40(7):239 – 250, 2014. 7th Workshop on Parallel Matrix Algorithms and Applications.

26. E. E. Tyrtyshnikov. A unifying approach to some old and new theorems on distribution and clustering. *Linear Algebra and its Applications*, 232(0):1 – 43, 1996.
27. J. van Lent and S. Vandewalle. Multigrid waveform relaxation for anisotropic partial differential equations. *Numerical Algorithms*, 31(1):361–380, Dec 2002.