Schur Complement Domain Decomposition in conjunction with Algebraic Multigrid methods based on Generic Approximate Inverses

P.I. Matskanidis
Department of Electrical and Computer Engineering,
School of Engineering, Democritus University of
Thrace, University Campus, Kimmeria,
GR 67100 Xanthi, Greece
Email: pascmats@ee.duth.gr

G.A. Gravvanis
Department of Electrical and Computer Engineering,
School of Engineering, Democritus University of
Thrace, University Campus, Kimmeria,
GR 67100 Xanthi, Greece
Email: ggravvan@ee.duth.gr

Abstract—**For decades, Domain Decomposition (DD) techniques have been used for the numerical solution of boundary value problems. In recent years, the Algebraic Multigrid (AMG) method has also seen significant rise in popularity as well as rapid evolution. In this article, a Domain Decomposition method is presented, based on the Schur complement system and an AMG solver, using generic approximate banded inverses based on incomplete LU factorization. Finally, the applicability and effectiveness of the proposed method on characteristic two dimensional boundary value problems is demonstrated and numerical results on the convergence behavior are given.**

I. Introduction

DOMAIN decomposition includes a significant range of computing techniques for the numerical solution of Partial Differential Equations (PDEs). Domain decomposition techniques are based on splitting the computational domain into smaller subdomains, with or without overlap. The problems in the subdomains are independent, thus rendering domain decomposition methods suitable for parallelization. Domain decomposition techniques can themselves be used as stationary iterative schemes, as well as preconditioners in order to accelerate the convergence of other iterative methods, specifically Krylov subspace methods [14].

Domain decomposition methods can be split into two categories: overlapping and non-overlapping methods. In overlapping DD methods, often referred to as Schwarz methods due to Schwarz's work in 1870 [22], the subdomains overlap by more than the interface. The overlapping methods have a simple algorithmic structure, since there is no need to solve special interface problems between neighbouring subdomains. This feature differentiates overlapping from non-overlapping DD methods [5],[23]. Overlapping methods operate by an iterative procedure, where the PDE is repeatedly solved within every subdomain. For each subdomain, the artificial internal boundary condition is provided by its neighbouring subdomains. The convergence of the solution on these internal boundaries ensures the convergence of the solution in the entire solution domain.

In non-overlapping DD methods, also referred to as iterative substructuring methods, the subdomains intersect only on their interface. Non-overlapping methods can furthermore be distinguished in primal and dual methods. Primal methods, such as BDDC [7], enforce the continuity of the solution across the subdomain interface by representing the value of the solution on all neighbouring subdomains by the same unknown. In dual methods, such as FETI [10], the continuity is further enforced by the use of Lagrange multipliers. Hybrid methods, such as FETI-DP [8],[9],[16], have also been introduced.

In the past decades, the development of multigrid methods has also been critical for the numerical solution of PDEs. An essential component of the multigrid method is the relaxation scheme, which efficiently reduces high frequency components of the error, however is inefficient at reducing the lower frequency ones [2]. Transferring the problem to a coarser grid, those low frequency errors become more oscillatory and can be effectively damped by a stationary iterative method. Recursive application of this process produces the multigrid methods [18].

The algebraic multigrid algorithm (AMG) was first introduced over twenty years ago [1],[20]. Unlike geometric multigrid, the algebraic multigrid method does not require knowledge of the geometry of the problem to define its components. This is the reason AMG is perfectly suited for unstructured grids, both in two and three dimensions, and complicated domains. Specifically, by considering a linear system Au=f, the AMG method requires only the coefficient matrix A and the right-hand side vector f. As a result, AMG solvers can easily be integrated into existing problem solving environments as standard solvers or preconditioners [18].

Consider a linear system Au=f, where $A=(a_{i,j}), i,j\in[1,n]$ is an (n×n) coefficient matrix. A "grid" is a set of indices of the variables, thus the first grid is $\Omega=\{1,2,\ldots,n\}$. Since AMG is independent of the geometry of the problem, the coarser grids, where the successive corrections to the solu-

978-1-4673-4471-5/$25.00 © 2013, IEEE

tion will be obtained, have to be constructed by the coarsening process, which is an essential component of the AMG algorithm.

The components needed for AMG, where superscripts indicate the level with 1 being the finest level [4],[27], are the following:

- Grids $\Omega^1 \supset \Omega^2 \supset \ldots \supset \Omega^N$ ($\Omega^1=\Omega$) containing the following two disjoint subsets:
  Coarse points set (C-points): $C^k, k=1,\ldots,N-1$.
  Fine points set (f-points): $F^k, k=1,\ldots,N-1$.
- Grid operators: $A^1, A^2, \ldots, A^N$, where $A^1=A$.
- Interpolation and restriction operators:
  $I_{k+1}^{k}, k=1,\ldots,N-1$; $I_k^{k+1}, k=1,\ldots,N-1$.
- A smoother (relaxation scheme) for each level.

AMG consists of two main phases: the setup phase, where the above components are created, and the solution phase that utilizes the components in the recursively defined multigrid cycle.

In this article, a domain decomposition method is presented, based on the Schur complement system. An Algebraic Multigrid method is used to solve the resulting linear systems for each domain, based on the use of generic approximate banded inverses, derived from the ILU(0) factorization [13],[18]. In section II, the Schur complement method is showcased, while in section III, the AMG method is presented.

Finally in section IV, the applicability of the new proposed scheme on two dimensional boundary value problems is demonstrated and numerical results on the convergence behavior and performance are given.

II. The Schur Complement Method

In this section, the Schur complement domain decomposition method is presented.

The Schur complement method is the earliest version of non-overlapping DD methods. Methods such as Dirichlet-Neumann and Neumann-Neumann are essentially the Schur complement method with the use of particular preconditioners.

Let us consider the Poisson equation on a region Ω, with zero Dirichlet data given on $\partial\Omega$, the boundary of Ω. Let us also suppose that Ω is partitioned into two non-overlapping subdomains Ω_i:

$$\overline{\Omega}=\overline{\Omega_1 \cup \Omega_2}, \Omega_1 \cap \Omega_2 = \emptyset, \Gamma = \partial\Omega_1 \cap \partial\Omega_2$$

as shown in Fig. 1 [25].

Assuming the boundaries of the subdomains are Lipschitz continuous, we consider the problem:

$$-\Delta u(x,y) = f \quad (x,y)\in\Omega \tag{1}$$

$$u(x,y)=0 \quad (x,y)\in\partial\Omega \tag{1.a}$$

Considering a triangulation of the domain Ω and a finite element approximation of the problem (1), assuming that subdomains consist of unions of elements, leads to a linear system

$$\mathrm{Au=f} \tag{2}$$

with a symmetric, positive definite matrix A. Partitioning the degrees of freedom to those internal to Ω_1, Ω_2 and those interior of Γ, the matrix A and vectors u, f can be expressed as:

$$A=\begin{bmatrix} A_{II}^{(1)} & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma} \end{bmatrix}, \mathrm{u}=\begin{bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_\Gamma \end{bmatrix}, \mathrm{f}=\begin{bmatrix} f_I^{(1)} \\ f_I^{(2)} \\ f_\Gamma \end{bmatrix} \tag{3}$$

The first step of many iterative domain decompositions methods eliminates the unknowns in the interior of the subdomains $u_I^{(i)}$. This leads to a block factorization of the matrix A (3) [25]:

$$A=\mathrm{LR}=\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ A_{\Gamma I}^{(1)} A_{II}^{(1)^{-1}} & A_{\Gamma I}^{(2)} A_{II}^{(2)^{-1}} & I \end{bmatrix} \begin{bmatrix} A_{II}^{(1)} & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ 0 & 0 & S \end{bmatrix} \tag{4}$$

and the resulting linear system:

$$\begin{bmatrix} A_{II}^{(1)} & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ 0 & 0 & S \end{bmatrix} u = \begin{bmatrix} f_I^{(1)} \\ f_I^{(2)} \\ g_\Gamma \end{bmatrix} \tag{5}$$

where I is the identity matrix and $S=A_{\Gamma\Gamma}-A_{\Gamma I}^{(1)} A_{II}^{(1)^{-1}} A_{I\Gamma}^{(1)} - A_{\Gamma I}^{(2)} A_{II}^{(2)^{-1}} A_{I\Gamma}^{(2)}$ is the Schur complement matrix relative to the unknowns on Γ.

Defining the local Schur complements by

$$S^{(i)} := A_{\Gamma\Gamma}^{(i)} - \sum_{i=1}^{2} A_{\Gamma I}^{(i)} A_{II}^{(i)^{-1}} A_{I\Gamma}^{(i)} \tag{6}$$

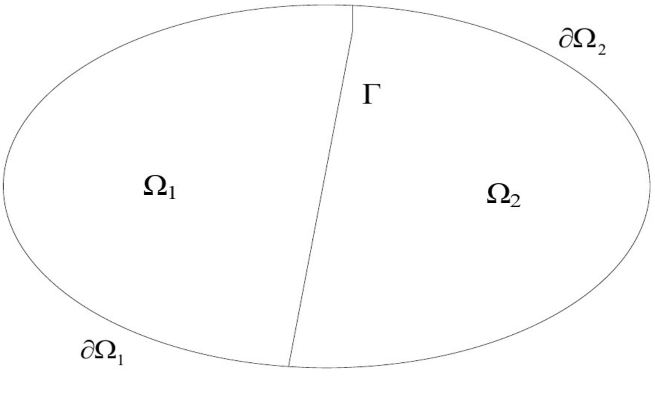

Fig. 1 Ω partitioned into two non-overlapping subdomains.

we find the Schur complement system for u_Γ to be [25]:

$$\mathrm{S}u_\Gamma = g_\Gamma \tag{7}$$

with

$$S = S^{(1)} + S^{(2)} \tag{8}$$

$$g_{\Gamma} := g_{\Gamma}^{(1)} + g_{\Gamma}^{(2)} = \left(f_{\Gamma}^{(1)} - A_{\Gamma I}^{(1)} A_{II}^{(1)^{-1}} f_{I}^{(1)}\right) + \left(f_{\Gamma}^{(2)} - A_{\Gamma I}^{(2)} A_{II}^{(2)^{-1}} f_{I}^{(2)}\right) \tag{9}$$

Once u_Γ is found by solving (7), the internal components can be found by using (5):

$$u_{I}^{(i)} = A_{II}^{(i)^{-1}} \left(f_{I}^{(1)} - A_{I\Gamma}^{(i)} u_{\Gamma}\right) \tag{10}$$

Equations (8) and (9) can be extended to a generic case, where there are more than two domains.

In the method presented in this paper, equation (7) is solved by using a typical direct solver, such as Gaussian Elimination. An AMG solver, based on the Generic Approximate Banded Inverse (GenAbI) algorithm [13], is used in order to solve the equations (10) for each domain.

III. The AMG Solver

In this section, we present the AMG method [13],[18] used for the solution of the systems that arise from the Schur complement method described in the previous section.

The key part of AMG's setup phase is the coarse-grid selection, which is the process of creating the degrees of freedom of a coarse-level problem. The goal of coarse-grid selection is to determine the sets C and F of coarse-grid and fine-grid points respectively, as well as a small set $C_i \subset C$ of interpolating points for each fine-grid point [20],[24]. This is called a C/F splitting, where C-points are variables that exist on both the fine and coarse levels and F-points are variables only on the fine level. Interpolation can then be defined as follows:

$$\left(I_{k+1}^{k} u^{k+1}\right)_{i} = \begin{cases} u_{i}^{k+1}, \mathrm{i} \in C \\ \sum\limits_{j \in C_i} w_{\mathrm{ij}} u_{j}^{k+1}, \mathrm{i} \in F \end{cases} \tag{11}$$

An important concept in the coarse grid selection is that of strong influence and strong dependence. It is highly likely that not all matrix coefficients are equally important to the selection of the coarse grids and thus only those that are "large enough" should be considered [18],[27].

It should be stated that point i depends on point j if a_{ij} is sufficiently large, denoting that in order to satisfy the i-th equation of the system, the node u_i is affected more from node u_j than other neighbouring nodes. We can then define the set of dependencies for point i as:

$$S_{i} = \left\{ j \neq \mathrm{i}, -a_{\mathrm{ij}} \geq \theta \max_{k \neq i} \left(-a_{\mathrm{ik}}\right) \right\} \tag{12}$$

where θ is called strength threshold and is important for its influence on stencil size and convergence [27]. A typical value for θ is 0.25. The set of influences for point i can be defined as the transpose of the dependencies set.

The concept of strong influence/dependence in conjunction with the following two heuristics is vital to creating a valid coarse grid [4],[20],[27]:

- ***H1***: For each point j that strongly influences a fine-grid point i, j is either a coarse-grid point or strongly depends on a coarse-grid point that also strongly influences i.
- ***H2***: C is a maximal set with the property that no C-point influences another C-point.

Condition H1 ensures the quality of the interpolation and condition H2 restricts the size of coarser grids.

The coarsening schemes of early AMG methods are based on the Ruge-Stüben (RS) coarsening method [20]. The Ruge-Stüben coarsening is the classical coarse-grid selection algorithm, based on enforcing heuristic H1, while implicitly using heuristic H2 as a guideline. It is a two-pass process, where the first pass selects a maximal independent set guaranteeing that every fine-grid point strongly depends on at least one coarse-grid point. Further details on the two-pass RS coarsening are given in [20].

Since RS coarsening selects only a single C-point in each iteration, its main drawback is its sequential nature. However, RS exhibits optimal scalability and convergence behavior for a variety of problems.

In recent years, there has been significant progress in the development of parallel coarse grid selection schemes, most of which are based on their sequential predecessors, such as CLJP [4] and PMIS [6]. An overview of such coarsening schemes and their performance is presented in [27].

The interpolation formula used for the purposes of this article is direct interpolation, in which the weights w_{ij} are defined as follows:

$$w_{\mathrm{ij}} = -\left(\frac{\sum_{k \in N_i} a_{\mathrm{ik}}}{\sum_{l \in C_i} a_{\mathrm{il}}} \right) \frac{a_{\mathrm{ij}}}{a_{\mathrm{ii}}} \tag{13}$$

This formula is easy to implement and only requires immediate neighbours of i, however, it generally leads to worse convergence rates compared to interpolation formulas that use extended neighbourhoods [27]. The interpolation operator can now be computed according to (11).

The restriction operator is defined as the transpose of the interpolation operator through the Galerkin condition [19]. Thus the next coarser level matrix is defined as the triple matrix product of the restriction operator, the finer grid matrix and the interpolation operator. When the current grid is considered "coarse enough", then the setup phase terminates and AMG proceeds to the solution phase.

The cycle strategy is an essential component of any multigrid algorithm and refers to the sequence in which the various grids are visited and the respective coarse grid corrections are obtained. The common cycle strategy is the V–cycle algorithm and is shown in Fig. 2 [18].

The solution can be achieved by successive applications of the cycle according to arbitrary termination criterion. The proposed multigrid scheme descends to the coarsest level and then, the multigrid method ascends to the finer levels and corrects the respective solution [18].

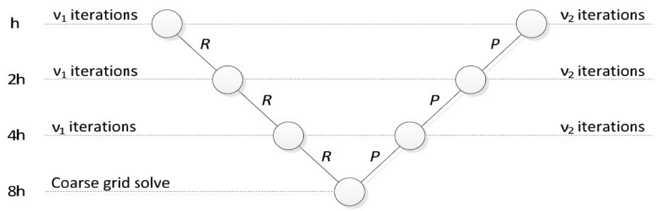

Fig. 2 The V–Cycle with finest grid Ω^h, coarsest grid Ω^{8h}, and v_1,v_2 pre–smoothing and post–smoothing steps, respectively.

An important component in multigrid methodology is the stationary iterative solver, namely smoother, that can be described by the following recurrence relation [13],[18]:

$$x_{(i+1)}^k = x_{(i)}^k + M^k r^k, \mathrm{r}^k = f^k - A^k x_{(i)}^k \quad (14)$$

where f^k,A^k are the right-hand side and coefficient matrix (at the k-th coarse level) and $x_{(i)}^k$ is the solution vector at the i-th iterative step. Equation (14) describes a family of stationary iterative methods, according to the choice of the M^k matrix.

Generic approximate banded inverses in conjunction with the general iterative method (14) can be used as smoothers for multigrid schemes, by choosing $M^k = (M^k)^{\delta \mathrm{l}}$, where $(M^k)^{\delta \mathrm{l}}$ is a class of approximate inverses. The class of smoothing methods proposed can be described as follows:

$$x_{(i+1)}^k = x_{(i)}^k + \omega \left(M^k\right)_r^{\delta \mathrm{l}} \left(f^k - A^k x_{(i)}^k\right) \quad (15)$$

where ω is the damping parameter with $0 < \omega \leq 1$ [13].

Let us assume the incomplete LU factorization, such that

$$A \approx \mathrm{LU} + R \quad (16)$$

where L and U are upper and lower matrices of the same nonzero structure as the lower and upper parts of coefficient matrix A respectively and R is some error matrix. This is the so-called incomplete LU factorization with zero fill-in, or more commonly ILU(0) factorization [21].

Let $M^{\delta \mathrm{l}} = (\mu_{\mathrm{i,j}}), \mathrm{i} \in [1,\mathrm{n}], \mathrm{j} \in [1 - \delta\mathrm{l} + 1, \mathrm{i} + \delta\mathrm{l} - 1]$ be the generic approximate banded inverse of the coefficient matrix A. The elements of a class of banded forms of the generic approximate inverse, by retaining δl and δl-1 elements in the lower and upper parts, can be computed by solving recursively the following systems [11],[13],[17]:

$$M^{\delta \mathrm{l}} L = U^{-1} \text{ and } \mathrm{UM}^{\delta \mathrm{l}} = L^{-1} \quad (17)$$

Then, the elements of the approximate inverse are computed by the Generic Approximate Banded Inverse (GenAbI) algorithm [13].

Specific information on the smoothing and approximation property for the GenAbI algorithm, as well as the use of the DOUR scheme [15] in order to dynamically compute the relaxation parameter ω for the smoothing scheme can be found in [12],[13],[18].

IV. Numerical Results

In this section we examine the effectiveness of the new proposed scheme, namely Domain Decomposition-Algebraic Multigrid method in conjunction with the Generic Approximate Banded Inverse matrix.

The convergence factor depends on the required number of iterations for convergence [2],[3],[26]. The convergence factor with respect to the 2-norm is defined as:

$$q = \sqrt[m]{\|r_m\|_2 / \|r_0\|_2} \quad (18)$$

where r_m is the residual vector at the m-th iteration. The termination criterion for the AMG solver is $\|r_m\|_2 < 10^{-10} \|r_0\|_2$ and the numbering of the grid is lexicographical. The maximum number of iterations was set to 200 iterations.

The strength threshold θ was set to 0.25 for the AMG solver. The values for the pre-smoothing and post-smoothing steps were set to v_1, v_2=2. The coarsest level, with its maximum amount of variables allowed set to 15, was solved using the BiCGSTAB method.

The coarsening scheme used in the first problem was the CLJP algorithm, while for the second problem the PMIS coarsening technique was utilized. It should also be mentioned that for the second problem, the AMG solver was modified to use the V-cycle as a preconditioner for the BiCGSTAB method in order to accelerate convergence.

Model Problem I: The model problem to be solved with the proposed scheme is the Poisson equation:

$$-\Delta \mathrm{u}(\mathrm{x,y}) = f \quad (19)$$

$$u(\mathrm{x,y}) = 0 \quad (\mathrm{x,y}) \in \partial\Omega \quad (19.a)$$

discretized with the finite element method, where f(x,y)=1, Ω is $[0,1] \times [0,2]$ and $\partial\Omega$ denotes the boundary of Ω. The domain Ω was split into 8 subdomains, as shown in Fig. 3.

In Table I, the convergence factors and convergence behavior are presented for various values of the order of the linear system n and “retention” parameter δl of the generic approximate banded inverse. Additionally, the convergence factors and convergence behavior for the same values of n and δl using AMG as a standalone solver without domain decomposition techniques are given. In Table II, the performance, in seconds, of both the DD/AMG and standalone AMG methods is showcased for various values of the order of the linear system n and “retention” parameter δl of the generic approximate banded inverse.

Model Problem II: Let us consider the following elliptic PDE:

$$-\Delta \mathrm{u}(\mathrm{x,y}) + \alpha(\mathrm{x,y}) u(\mathrm{x,y}) = f(\mathrm{x,y}) \quad (20)$$

$$u(\mathrm{x,y}) = 0 \quad (\mathrm{x,y}) \in \partial\Omega \quad (20.a)$$

discretized with the finite element method, where α(x,y)=-40x+70y, $f(x,y)=19x-44y^2$, Ω is the unit square and $\partial\Omega$ denotes the boundary of Ω. The domain Ω was split into 16 subdomains, as shown in Fig. 4.

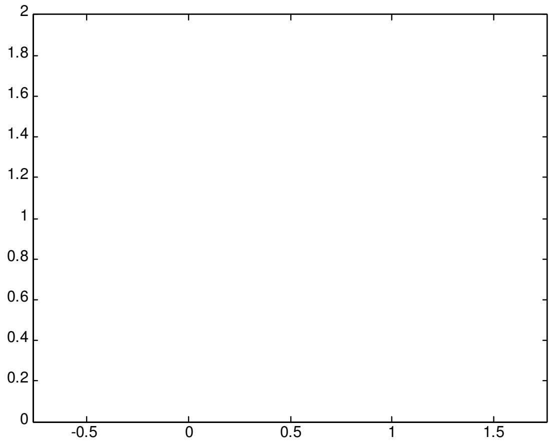

Fig. 3 Domain Ω split into 8 subdomains for model problem I.

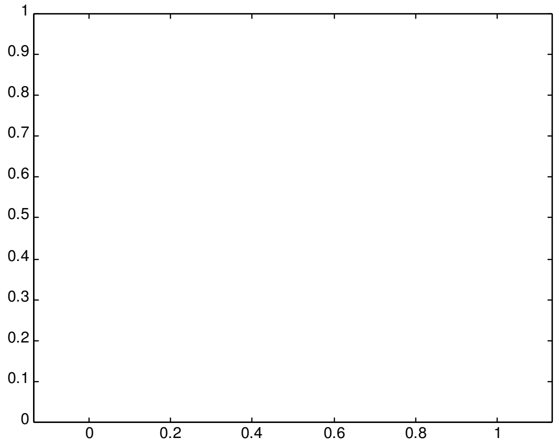

Fig. 4 Domain Ω split into 16 subdomains for model problem II.

In Table III, convergence behavior for various values of the order of the linear system n and "retention" parameter δl of the generic approximate banded inverse is presented. Additionally, convergence behavior for the same values of n and δl using AMG-BiCGSTAB as a standalone solver without domain decomposition techniques is given. In Table IV, the performance, in seconds, of both the DD/AMG-BiCGSTAB and standalone AMG-BiCGSTAB methods is showcased for various values of the order of the linear system n and "retention" parameter δl of the generic approximate banded inverse.

It should be noted that, for both problems, the convergence and performance results for the DD/AMG method are the average of the results taken from all domains, since the domains were not identical.

As already expected considering past results [18], increasing the value of the "retention" parameter δl leads to improved convergence behavior. Additionally, we notice that solving the linear systems arising from the subdomains is significantly more efficient, both in performance and convergence.

One of the drawbacks of AMG methods is the computational work added by the setup phase in addition to the solution phase. The resulting smaller linear systems from the domain decomposition method significantly reduce the workload for both phases. Considering that these systems can be solved in parallel since domain decomposition methods are well suited for parallel computing, the combination of domain decomposition with AMG can be very efficient.

Finally, it should be stated that the effectiveness and applicability of the new proposed method will be shown when applied to more general problems, such as quasilinear boundary-value problems or convection-diffusion problems.

V. Conclusions

A Schur complement domain decomposition method, utilizing an AMG solver, based on generic approximate banded inverse matrices, for solving the resulting subdomain systems was presented. The use of domain decomposition techniques was proven to be effective by improving the performance and convergence behavior of Algebraic Multigrid method. Since domain decomposition methods lead to smaller linear systems, arising from each subdomain, the load for both the setup and solution phase of the AMG solver is significantly reduced.

References

[1] A. Brandt, S.F. McCormick and J.W. Ruge, "Algebraic multigrid (AMG) for sparse matrix equations", in *Sparsity and Its Applications*, D.J. Evans (ed), Cambridge University Press, 1984.

[2] L.W. Briggs, V. Henson and S.F. McCormick, *A multigrid tutorial*, SIAM, 2000.

[3] O. Bröker, M.J. Grote, C. Mayer and A. Reusken, "Robust parallel smoothing for multigrid via sparse approximate inverses", *SIAM Journal on Scientific Computing*, vol. 23(4), pp. 1396–1417, 2001.

[4] A. Cleary, R. Falgout, V. Henson and J. Jones, "Coarse-grid selection for parallel algebraic multigrid", in *Proc. Fifth International Symposium on Solving Irregularly Structured Problems in Parallel*, Vol. 1457, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 104–115, 1998.

[5] T.F. Chan and T.P. Mathew, "Domain decomposition algorithms", In *Acta Numerica 1994*, Cambridge University Press, pp. 61-143,1994.

[6] H. De Sterck, U. Yang and J. Heys, "Reducing complexity in parallel algebraic multigrid preconditioners", *SIAM J. Mat. Anal. and Appl.*, vol. 27, pp. 1019-1039, 2006.

[7] C.R. Dohrmann, "A preconditioner for substructuring based on constrained energy minimization", *SIAM J. Sci. Comput.*, vol. 25, pp. 246-258, 2003.

[8] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen, "FETI-DP: a dual-primal unified FETI method – part I: A faster alternative to the two-level FETI method", *Int. J. Numer. Meth. Engng,* vol. 50, pp. 1523-1544, 2001.

[9] C. Farhat, M. Lesoinne, and K. Pierson, "A scalable dual-primal domain decomposition method", *Numer. Linear Algebra Appl.*, vol. 7, pp. 687-714, 2000.

[10] C. Farhat and F.X. Roux, "A method of finite element tearing and interconnecting and its parallel solution algorithm", *Int. J. Numer. Meth. Engng,* vol. 32, pp. 1205-1227, 1991.

[11] G.A. Gravvanis, Explicit Approximate Inverse Preconditioning Techniques, *Archives of Computational Methods in Engineering,* vol. 9(4), pp. 371-402, 2002.

[12] G.A. Gravvanis, C.K. Filelis-Papadopoulos and P.I. Matskanidis, "A survey on the parallelization issues of geometric and algebraic multigrid methods based on generic banded approximate inverses", in Computational Technology Reviews vol. 7, B.H.V. Topping and P. Ivanyi (eds), pp. 65-98, Saxe-Coburg Publications, 2013.

[13] G.A. Gravvanis, C. K. Filelis-Papadopoulos and P. I. Matskanidis, “Algebraic multigrid methods based on Generic Approximate Matrix Techniques”, Report TR/ECE/ASC-AMA/2012/13, submitted.

[14] A. Greenbaum, *Iterative methods for solving linear systems*, SIAM, Philadelphia, PA, USA, 1997.

[15] R. Haelterman, J. Viederndeels and D. Van Heule, “Non–stationary two–stage relaxation based on the principle of aggregation multi-grid”, *Computational Fluid Dynamics 2006*, Part 3, Springer, pp. 243–248, 2009.

[16] A. Klawonn, O.B. Widlund, and M. Dryja, “Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients”, *SIAM J. Numer. Anal.*, vol. 40, pp.159-179, 2002.

[17] E.A. Lipitakis and D.J. Evans, “Explicit semi-direct methods based on approximate inverse matrix techniques for solving boundary-value problems on parallel processors”, *Math. and Computers in Simulation*, vol. 29, pp. 1-17, 1987.

[18] P.I. Matskanidis and G.A. Gravvanis, “On the algebraic multigrid method based on generic approximate banded inverses”, in *Proc. 2012 Panhellenic Conference on Informatics (PCI 2012)*, pp. 211-216, IEEE Computer Society, 2012.

[19] S.F. McCormick, “Multigrid methods for variational problems: general theory for the V-cycle”, *SIAM J. Numer. Anal.*, vol. 22, pp. 634-643, 1985.

[20] J.W. Ruge and K. Stuben, “Algebraic multigrid (AMG)”, in *Multigrid Methods*, S.F. McCormick (ed), vol. 3, Frontiers in Applied Mathematics, SIAM, pp. 73–130, 1987.

[21] Y. Saad, *Iterative methods for sparse linear systems*, PWS Publishing, 1996.

[22] H.A. Schwarz, “Über eine grenzübergang durch alternirendes verfahren, Gesammelete Mathematische Abhandlungen”, Springer-Verlag, 2, pp. 133-143, 1980. First published in *Viertel-jaahrsschift der Naturforschenden Gesellschaft* in Zürich, vol.15, pp. 272-286, 1870.

[23] B.F. Smith, P.E. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

[24] K. Stuben, “Algebraic multigrid (AMG): An introduction with applications”, in *Multigrid*, U. Trottenberg, C. Oosterlee, and A. Schuller (eds), Academic Press, 2001.

[25] A. Toselli and O.B. Widlund, *Domain Decomposition methods – Algorithms and Theory*, Springer, 2005.

[26] U. Trottenberg, C.W. Osterlee and A. Schuller, *Multigrid*, Academic Press, 2000.

[27] U.M. Yang, “Parallel Algebraic Multigrid Methods - High Performance Preconditioners”, in *Numerical Solution of Partial Differential Equations on Parallel Computers*, A.M. Bruaset and A. Tveito (eds), vol. 51, Springer-Verlag, pp. 209-236, 2006.

TABLE I.
CONVERGENCE BEHAVIOUR OF THE DD/AMG METHOD FOR MODEL PROBLEM I

		n=24353		n=96833		n=386177	
Method	**δl**	**q**	**its**	**q**	**its**	**q**	**its**
DD/AMG	**1**	0.2152	15	0.2556	17	0.3074	20
	2	0.2102	15	0.2467	17	0.3024	20
	50	0.1946	14	0.2374	16	0.2873	19
AMG	**1**	0.3483	22	0.4137	27	0.4881	33
	2	0.3427	22	0.4090	26	0.4839	32
	50	0.3142	20	0.3794	24	0.4667	30

TABLE II.
PERFORMANCE, IN SECONDS, OF THE DD/AMG METHOD FOR MODEL PROBLEM I

		n=24353	n=96833	n=386177
Method	**δl**	**time**	**time**	**time**
DD/AMG	**1**	0.0971	0.4499	2.4231
	2	0.0990	0.4653	2.4991
	50	0.2994	1.3975	7.1642
AMG	**1**	1.1680	6.025	33.2329
	2	1.1954	6.2288	33.7139
	50	3.339	15.4747	87.9342

TABLE III.
CONVERGENCE BEHAVIOUR OF THE DD/AMG-BICGSTAB METHOD FOR MODEL PROBLEM II

		n=11457	**n=45441**	**n=180993**
Method	**δl**	**its**	**its**	**its**
DD/AMG-BiCGSTAB	**1**	10	11	14
	2	9	10	13
	50	7	9	11
AMG-BiCGSTAB	**1**	15	21	34
	2	13	17	27
	50	12	15	23

TABLE IV.
PERFORMANCE, IN SECONDS, OF THE DD/AMG-BICGSTAB METHOD FOR MODEL PROBLEM II

		n=11457	**n=45441**	**n=180993**
Method	**δl**	**time**	**time**	**time**
DD/AMG-BiCGSTAB	**1**	0.0174	0.0704	0.3441
	2	0.0182	0.0706	0.3618
	50	0.0399	0.1962	1.0097
AMG-BiCGSTAB	**1**	0.3936	2.2724	17.0591
	2	0.3854	2.003	14.9678
	50	1.2891	5.5477	32.5779