

Krylov-based algebraic multigrid for edge elements

François Musy, Artem Napov, Yvan Notay, Ronan Perrussel, Riccardo Scorretti

▶ **To cite this version:**

François Musy, Artem Napov, Yvan Notay, Ronan Perrussel, Riccardo Scorretti. Krylov-based algebraic multigrid for edge elements. Compumag 2009, Nov 2009, Florianópolis, Brazil. pp.1072. hal-00412347

HAL Id: hal-00412347
https://hal.science/hal-00412347

Submitted on 1 Sep 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Krylov-based algebraic multigrid for edge elements

François Musy[1], Artem Napov[2], Yvan Notay[2], Ronan Perrussel[3] and Riccardo Scorretti[3]
[1] Institut Camille Jordan, CNRS UMR5208; [3] Laboratoire Ampère, CNRS UMR5005
Université de Lyon, École Centrale de Lyon, F-69134 Écully, France
[2] Service de Métrologie Nucléaire, Université Libre de Bruxelles, B-1050 Brussels, Belgium
mail to: ronan.perrussel@ec-lyon.fr

***Abstract*—This work tackles the evaluation of a multigrid cycling strategy using inner flexible Krylov subspace iterations. It provides a valuable improvement to the Reitzinger and Schöberl algebraic multigrid method for systems coming from edge-element discretizations.**

I. Introduction

In finite element method libraries, the linear system solvers play a key role in terms of performances in the computing time and in the memory consumption. Multigrid methods are among the most efficient iterative linear system solvers for elliptic problems [1]. In this work, we focus on the discretization with the lowest order edge element of a curl-curl equation

$$\operatorname{curl}\delta\operatorname{curl}\mathbf{U}+\gamma\mathbf{U}=\mathbf{f}\text{ on }\Omega\subset\mathbb{R}^d\,(d=2\text{ or }3),\quad(1)$$

which gives rise to a linear system $Ax=f$. Several Algebraic MultiGrid (AMG) algorithms have been proposed for "efficiently" solving this system [2]–[7]. We use the main ideas of the algorithm proposed by Reitzinger and Schöberl (RS), who were the first to propose an edge prolongation matrix satisfying a *commutativity property* [3]. Our motivation comes from the fact that this algorithm has the fastest setup time and gives the sparsest prolongation and coarse grid matrices among the available methods. However, it also gives the poorest rate of convergence, leading, in the literature, to a non-optimal multigrid solver. We combine the RS algorithm ideas with a Krylov-based multigrid cycle in order to recover classical multigrid performance. Numerical experiments are performed on 2D problems.

II. Components of the algorithm

A. Recursive Krylov-based multigrid cycle

The multigrid preconditioning algorithm on grid k (denoted by MGp) is given by *Algorithm 1* where matrix A_k represents the discrete problem on grid k (grid 0 is the coarsest) and P_k is the prolongation matrix from grid $k-1$ to grid k.

Algorithm 1: **INPUT** r_k, **OUTPUT** $z_k=\mathrm{MGp}(r_k)$

1) Relax using smoother M_k: $z_k \leftarrow M_k^{-1}r_k$.
2) Restrict residual: $r_{k-1}\leftarrow P_k^T(r_k-A_kz_k)$.
3) Compute an approximate solution $\tilde{\theta}_{k-1}$ to:
$$A_{k-1}\theta_{k-1}=r_{k-1}.\quad(2)$$
4) Prolongate coarse-grid correction: $z_k\leftarrow z_k+P_k\tilde{\theta}_{k-1}$.
5) Relax using smoother M_k: $z_k\leftarrow z_k+M_k^{-1}(r_k-A_kz_k)$.

If $k-1$ is zero, the solution of (2) is exact. If not, a common strategy is to perform once $\mathrm{MGp}(r_{k-1})$ (V-cycle) or twice (W-cycle) to approximately solve (2). Following [8], we propose to consider a K_2-cycle which is a W-cycle completed by the two first iterations of a flexible Krylov subspace method. An example with the flexible conjugate gradient is given by *Algorithm 2*.

Algorithm 2: **INPUT** r_{k-1}, **OUTPUT** $\tilde{\theta}_{k-1}=\mathrm{CS}(r_{k-1})$

1) First iteration:
$d_{k-1}\leftarrow\mathrm{MGp}(r_{k-1})$; $\alpha_{k-1}\leftarrow\frac{r_{k-1}^Td_{k-1}}{d_{k-1}^TA_{k-1}d_{k-1}}$;
$\tilde{\theta}_{k-1}\leftarrow\alpha_{k-1}d_{k-1}$; $r_{k-1}\leftarrow r_{k-1}-\alpha_{k-1}A_{k-1}d_{k-1}$.
2) Second iteration:
$c_{k-1}\leftarrow\mathrm{MGp}(r_{k-1})$;
$d_{k-1}\leftarrow d_{k-1}-\frac{c_{k-1}^TA_{k-1}d_{k-1}}{d_{k-1}^TA_{k-1}d_{k-1}}c_{k-1}$;
$\tilde{\theta}_{k-1}\leftarrow\tilde{\theta}_{k-1}+\frac{r_{k-1}^Td_{k-1}}{d_{k-1}^TA_{k-1}d_{k-1}}d_{k-1}$.

The iteration cost in time and memory of both K_2- and W-cycle is roughly equivalent. Moreover, their convergence rate are *theoretically similar* [8]. Nevertheless, in practice the K_2-cycle has a better convergence rate than the W-cycle [8].

B. Prolongation matrix and smoother

Reitzinger and Schöberl proposed to construct an edge prolongation matrix P^{edg} satisfying a commutativity property:

$$P^{\mathrm{edg}}G^H=G^hP^{\mathrm{nod}}.\quad(3)$$

In this equality, P^{nod} is a nodal prolongation matrix obtained from a nodal auxiliary matrix and G^h and G^H are respectively fine and coarse edge-node incidence matrix.

At the finest level, G^h is given by the relation between vertices and edges on the finite element mesh. At the same level, the nodal auxiliary matrix B, following the proposition in [9], contains information about the edge-node incidence, the lengths of the edges and coefficient δ from (1). The construction of P^{nod} is then performed by the double pairwise-aggregation algorithm proposed in [10]. Once P^{nod} is known, the definition of the coarse edge incidence matrix G^H and the edge prolongator P^{edg} are straightforward. To apply the construction recursively, coarse grid matrices are obtained by Galerkin product:

$$A^H=(P^{\mathrm{edg}})^TAP^{\mathrm{edg}},\quad B^H=(P^{\mathrm{nod}})^TBP^{\mathrm{nod}}.\quad(4)$$

The smoother is a symmetric version of the smoother proposed in [11] in a geometric multigrid context. It uses on each grid the corresponding edge-node incidence matrix.

III. NUMERICAL RESULTS

The behavior of the method with an increasing size of the problem and several kinds of parameters δ and γ is studied.

A. Structured mesh

The examples are taken from [12]. The domain is a unit square and Dirichlet boundary conditions are enforced. The mesh with triangles is structured but this fact is not used by the solver. For the parameters, three situations are considered:

1) Homogeneous parameters: $\delta = \gamma = 1$.
2) Oscillating with discontinuities for $\delta = f(x, y)$ and $\gamma = 1$. Function f have the following definition:

$$f = C(2 + \sin(40\pi x))^2(2 + \cos(40\pi y))^2$$

$$\text{with } C = \begin{cases} 10 \text{ in }]0, 0.5[\times]0, 0.5[, \\ 10^4 \text{ in }]0.5, 1[\times]0, 0.5[, \\ 10^{-1} \text{ in }]0, 0.5[\times]0.5, 1[, \\ 10^2 \text{ in }]0.5, 1[\times]0.5, 1[. \end{cases}$$

3) Oscillating with discontinuities for $\delta = f(x, y)$ and for $\gamma = f(y, x)$.

The behavior of the iterative method is evaluated by computing the average convergence rate σ_{est} in energy norm:

$$\sigma_{\text{est}} = \left(\frac{\text{er}_k^t A \text{er}_k}{\text{er}_0^t A \text{er}_0} \right)^{1/(2k_f)}$$

with er_k the error at the k-th iteration and k_f the iteration where the stopping criterion is reached.

An examination of the convergence rate of the two-grid algorithm provides information to predict the behavior of the multigrid cycle. In Table I, the two-grid convergence rate is quasi-independent of the size of the problem and is bounded away from 1. Case 3 is the worst situation probably because the aggregation does not take into account variation of γ; it may explain the bad convergence rate for one particular mesh in this case (in bold). The number of unknowns is divided roughly by 4 between fine and coarse grids which is the best trade-off between coarsening and the overall arithmetic complexity.

With such convergence rates for the two-grid solver, it is illustrated in Table II that the convergence rates of the W-cycle (and of course of the V-cycle) deteriorates as the number of grids increases. On the contrary, the convergence rate of the K_2-cycle has a remarkable stability.

TABLE I
RESULTS OBTAINED WITH A TWO-GRID SOLVER.

	d.o.f. fine grid	736	3008	12160	48896
Case 1	d.o.f. coarse grid	184	751	3040	12224
	σ_{est}	0.62	0.68	0.70	0.71
Case 2	d.o.f. coarse grid	187	788	3087	12404
	σ_{est}	0.64	0.72	0.69	0.69
Case 3	d.o.f. coarse grid	187	788	3087	12404
	σ_{est}	0.61	0.72	**0.87**	0.68

TABLE II
COMPARISON BETWEEN A W-CYCLE AND A K_2-CYCLE SOLVER. CASE 3 ON THE MESH WITH 48896 D.O.F..

# grids	3	4	5	6	7
d.o.f. coarsest grid	3279	844	216	55	11
K_2-cycle, σ_{est}	0.68	0.68	0.68	0.68	0.68
W-cycle, σ_{est}	0.83	0.89	0.93	0.95	0.97

B. Unstructured mesh

Coefficients δ and γ are those defined in Case 3. The domain is still a unit square but an unstructured mesh, refined at the center of the square, is considered. Table III confirms the results obtained for the structured meshes.

TABLE III
COMPARISON BETWEEN A TWO-GRID, A W-CYCLE AND A K_2-CYCLE SOLVER.

d.o.f. finest grid	2043	8988	36717
two-grid, σ_{est}	0.64	0.61	0.61
# grids	5	6	5
d.o.f. coarsest grid	6	9	216
K_2-cycle, σ_{est}	0.64	0.62	0.61
W-cycle, σ_{est}	0.90	0.80	0.95

The change of cycling proposed for the RS algorithm seems to make this algorithm robust and optimal in 2D. The proposed algorithm can be straightforwardly extended to 3D problems but our actual Matlab implementation does not enable us to propose results on challenging problems and also to discuss computational time and memory requirements. We are working on an implementation in Fortran90 in order to propose valuable comparisons on realistic problems.

REFERENCES

[1] D. W. Hackbusch, *Multi-Grid Methods and Applications*, 1985.
[2] R. Beck, "Algebraic multigrid by components splitting for edge elements on simplicial triangulations," *Preprint SC 99-40, ZIB*, Dec. 1999.
[3] S. Reitzinger and J. Schöberl, "An algebraic multigrid method for finite element discretizations with edge elements," *Numerical Linear Algebra with Applications*, vol. 9, no. 3, pp. 223–238, 2002.
[4] J. J. Hu, R. S. Tuminaro, P. B. Bochev, C. J. Garasi, and A. C. Robinson, "Toward an h-independent algebraic multigrid method for Maxwell's equations," *SIAM Journal on Scientific Computing*, vol. 27, no. 5, pp. 1669–1688, 2006.
[5] T. Boonen, G. Deliège, and S. Vandewalle, "On algebraic multigrid methods derived from partition of unity nodal prolongators," *Numerical Linear Algebra with Applications*, vol. 13, no. 2-3, pp. 105–131, 2006.
[6] R. Hiptmair and J. Xu, "Nodal auxiliary space preconditioning in H(curl) and H(div) spaces," *SIAM J. Numer. Anal.*, vol. 45, no. 6, pp. 2483–2509, 2007.
[7] F. Musy, L. Nicolas, and R. Perrussel, "Compatible coarse nodal and edge elements through energy functionals," *SIAM Journal on Scientific Computing*, vol. 29, no. 3, pp. 1315–1337, 2007.
[8] Y. Notay and P. S. Vassilevski, "Recursive krylov-based multigrid cycles," *Numerical Linear Algebra with Applications*, vol. 15, p. 473–487, 2008.
[9] M. Kaltenbacher and S. Reitzinger, "Algebraic multigrid methods for nodal and edge based discretizations of maxwell's equations," *International Compumag Society Newsletter*, vol. 9, no. 2, pp. 15–23, 2002.
[10] Y. Notay, "An aggregation-based algebraic multigrid method," Université Libre de Bruxelles, Brussels, Belgium, Tech. Rep. GANMN 08–02, 2008, http://homepages.ulb.ac.be/ ynotay.
[11] R. Hiptmair, "Multigrid method for Maxwell's equations," *SIAM Journal on Numerical Analysis*, vol. 36, no. 1, pp. 204–225 (electronic), 1999.
[12] J. Jones and B. Lee, "A multigrid method for variable coefficient maxwell's equations," *SIAM Journal on Scientific Computing*, vol. 27, no. 5, pp. 1689–1708, 2006.