A flexible 2-level Neumann-Neumann method for structural analysis problems

Petter E. Bjørstad[1] and Piotr Krzyżanowski[2]

[1] Institute of Informatics, University of Bergen, 5020 Bergen, Norway,
petter@ii.uib.no
[2] Institute of Applied Mathematics, Warsaw University, Banacha 2, 02-097 Warszawa, Poland,
przykry@mimuw.edu.pl

Abstract. We discuss a new approach for the construction of the second-level Neumann-Neumann coarse space. Our method is based on an inexpensive and parallel analysis of the lower part spectrum of each subdomain stiffness matrix. We show that the method is flexible enough to converge fast on nonstandard decompositions and various types of finite elements used in structural analysis packages.

1 Introduction

In this paper, we consider a coarse space construction method for Neumann-Neumann domain decomposition methods used in the iterative substructuring solution of finite element problems with a symmetric and positive definite stiffness matrix K. Below we shall briefly introduce the concepts and notation that will be used in the rest of the paper.

Let the finite element triangulation of the physical domain Ω be decomposed into N non-overlapping subdomains Ω_i, $i = 1, \ldots, N$. For the ith subdomain, let us first number the inner nodes followed by the interface (subdomain boundary) nodes. As a result, the stiffness matrix $K^{(i)}$ for subdomain Ω_i can be written as

$$K^{(i)} = \begin{pmatrix} K_{II}^{(i)} & K_{IB}^{(i)} \\ K_{BI}^{(i)} & K_{BB}^{(i)} \end{pmatrix} \tag{1}$$

We call the matrices $K^{(i)}$ Neumann matrices, they are symmetric, but those that have no external boundary condition may be singular. We also call such subdomains that have a singular Neumann matrix for floating substructures. Let us consider an iterative substructuring algorithm [13, Chapter 4] which uses the decomposition introduced above. After elimination of the internal unknowns, an interface problem of the form $Su = g$, where $S = \sum_{i=1}^{N} R^{(i)} S^{(i)} (R^{(i)})^T$ remains to be solved, where $R^{(i)}$ denotes the prolongation operator associated with subdomain Ω_i and $S^{(i)}$ is the local Schur complement of $K^{(i)}$ with respect to the interface unknowns $K_{BB}^{(i)}$,

$$S^{(i)} = K_{BB}^{(i)} - K_{BI}^{(i)} \cdot [K_{II}^{(i)}]^{-1} \cdot K_{IB}^{(i)}. \tag{2}$$

This interface problem with a symmetric and positive definite matrix S is then solved iteratively by the Preconditioned Conjugate Gradient (PCG) method. Note that we need not assemble S, because the iterative algorithm only requires the action of the local Schur complement matrices $S^{(i)}$; moreover, the $S^{(i)}$ matrices may either be computed directly, or they can be expressed implicitly through vector multiplication using the expression (2).

The Neumann-Neumann method [4] is an efficient way of preconditioning the interface operator S under very mild assumptions on the decomposition of the domain. This method has been significantly improved by Mandel, who introduced a second-level variant of the Neumann-Neumann method, also known as the Balancing Domain Decomposition Algorithm [10].

The second-level Neumann-Neumann method and its modifications have been studied extensively during the past years, see for example [2] [5], [6], [8], [9], [11], These methods have proved very successful, partly because they have been tailored to the problem at hand. They require additional knowledge related to the problem being solved: which elements are being used, what is the geometry of the decomposition, and so on.

The goal of the method we discuss here is a bit different. Instead of trying to solve a particular problem with optimal complexity, we rather try to design a flexible and robust coarse space able to solve a variety of problems efficiently, but without detailed additional information.

Our coarse space construction is based on a relatively inexpensive analysis of the lower part of the spectrum of the local stiffness matrices. The algorithm has been implemented in a domain decomposition software framework developed at Parallab within the European Union research project PARASOL. Its current version, called DD 3.0, is available from Parallab (see [1], where preliminary results related to the overall efficiency of the method have been reported).

2 Motivation for a more flexible approach

Following [10], we define the coarse grid problem for the Neumann-Neumann preconditioner by local coarse space contributions, $Z^{(i)} = \text{span}\{Z_k^{(i)} : k = 1, \ldots, n_i\}$ for each substructure, $i = 1, \ldots, N$. The entries of the global coarse space matrix are then constructed from the local contributions using the formula $(D^{(j)} R^{(j)} Z_l^{(j)})^T \cdot S \cdot D^{(i)} R^{(i)} Z_k^{(i)}$, where $\{D^{(i)}\}_{i=1}^N$ is a diagonal scaling. The local coarse space vectors $Z^{(i)}$ should contain the null space of the local Schur operators $S^{(i)}$: $\ker S^{(i)} \subset Z^{(i)}$, see [10] for further details.

There are cases when the deficiency of the local stiffness matrix (that is, $\dim \ker S^{(i)}$) is not known *a priori* and also, as we shall see, situations where the overall convergence of the Neumann-Neumann method depends on the distribution of the nonzero eigenvalues of the local Schur complement.

Fig. 1. Eigenvalue distribution of a local matrix $S^{(6)}$ (left) and of the global Schur complement matrix S (right). For the local matrices, the 6 lowest eigenvalues correspond to the rigid body motions. There is a cluster of 22 intermediate eigenvalues in both $S^{(6)}$ and $K^{(6)}$. Also, note that these modes are not present in S, having a smallest eigenvalue of order 10^8.

The former may happen when a single substructure consists of an unknown number of disconnected parts or when substructures have an unknown number of *hanging* parts.

Structural mechanics equations discretized with many shell elements may require more flexibility in how the size of the local coarse space is adapted to the problem. Their $S^{(i)}$ and $K^{(i)}$ matrices may contain – in addition to a cluster of zero eigenvalues related to the rigid body modes – a cluster of relatively small, "intermediate" eigenvalues that influence the convergence rate of the Neumann-Neumann method. It appears that this paper is the first to propose an adaptive algorithm that can construct an effective coarse space for problems of this kind.

Example 1 (Roof problem) Consider a cylindrical roof-like example, discretized with triangular shell elements with 6 degrees of freedom per node. This, and other shell examples, have been generated using the structural analysis package SESAM by Det Norske Veritas (Norway). These elements are a mixture of usual shells and membranes in order to improve the approximation properties.

The equations have been discretized on a 60x60 triangular finite element mesh. The roof is made of elastic material with Young modulus $= 0.21 \cdot 10^{12}$, Poisson ratio 0.3 and density $7.85 \cdot 10^3$ and fixed along the boundary; it has been manually decomposed into 16 subdomains of rectangular shape. The roof thickness was set to 1% of its leading dimension.

In addition to a 6-dimensional null space of the corresponding $S^{(i)}$ and $K^{(i)}$, each "floating" subdomain's $S^{(i)}$ and $K^{(i)}$ features a cluster of intermediate eigenvalues which is *not* present in the global Schur operator S, see Figure 1.

3 Adaptively constructed eigenvector based coarse space

Our coarse space construction method is based on a relatively inexpensive eigenanalysis of both Neumann and Dirichlet stiffness matrices. In our approach we make it possible to enrich the coarse space adaptively so that substructure low

energy modes are included in the second level solve of the Neumann-Neumann algorithm. The idea of enlarging the coarse space is in some sense similar to the method [3] used to improve the convergence rate of the Algebraic Multigrid Method. We note that the coarse spaces proposed and analyzed in [8] and [9] are also enriched coarse spaces. Below there is the algorithm we use to compute the local coarse space contributions $Z^{(i)}$:

Algorithm 1
For subdomain i,

1. Compute an approximation of the smallest eigenvalue $\lambda_{min}^{(i)}$ of $K_{II}^{(i)}$.
2. Based on this information, set the threshold value $\Lambda^{(i)}$.
3. Calculate those eigenvectors $\{\tilde{Z}_k^{(i)}\}$ of $S^{(i)}$ which correspond to eigenvalues smaller than $\Lambda^{(i)}$.
4. Define the basis of the local coarse space $Z^{(i)}$ as the space spanned by the eigenvectors $\{\tilde{Z}_k^{(i)}\}$ computed in Step 3.

In Step 1, one step of Rayleigh quotient iteration (RQI) is performed in order to determine the magnitude of the smallest eigenvalue of $K_{II}^{(i)}$. We set the threshold value $\Lambda^{(i)}$ in Step 2 according to (θ is a relaxation parameter):

$$\Lambda^{(i)} := \theta \cdot \max_i \{\lambda_{min}^{(i)}\}. \tag{3}$$

A drawback of setting $\Lambda^{(i)}$ according to (3) is that it requires synchronization between the subdomains which will contribute to a reduction in the overall parallel efficiency, [1].

In Step 3 of the algorithm, we compute all eigenvectors $\tilde{Z}_k^{(i)}$ of $S^{(i)}$ which correspond to eigenvalues not exceeding $\Lambda^{(i)}$. The number of such eigenvectors may vary significantly across the subdomains, and may be large, especially in the case of shell problems. Knowing the threshold value, we can adapt the eigenanalysis of $S^{(i)}$ and thus reduce the computational work involved almost exclusively to those eigenvectors that finally will be included in the coarse space. As described in [1], for a given subdomain i, the Schur complement $S^{(i)}$ is computed explicitly in the Neumann-Neumann algorithm only when the Schur matrix is moderately sized; when the interface is large, the Schur complement is available only implicitly through relation (2). It turns out that in this case we can quite safely switch to finding the lowest eigenvectors of the Neumann matrix $K^{(i)}$. In order to compute the required eigenvectors of B_i, where $B_i = S^{(i)}$ or $K^{(i)}$, we use a (block) variant of the well known RQI, calculating the eigenvectors in chunks of M vectors until we surpass the threshold.

In Step 4 we define the basis of the local coarse space $Z^{(i)}$ as the set of restrictions of $\tilde{Z}_k^{(i)}$ to the interface of Ω_i.

3.1 Computational complexity and implementation

We found out that 2 RQI iterations are enough to get the eigenvectors with sufficient accuracy even if d_i is quite large. Then the dominant computational

cost of the local coarse space evaluation is not more than $2(1 + d_i)$ local solves with $S^{(i)}$ or $K^{(i)}$. This usually is more than required by the crosspoint method. For example, taking $d_i = 6$, the estimate for the number of the rigid body modes of a shell, our method will require 14 local solves per subdomain, while the crosspoint method constraining all displacement degrees of freedom as in [9, Remark 3.6] will use 9 or 12 local solves per subdomain, depending on the subdomain shape (triangle or rectangle). When it comes to the shell examples from DNV (Example 1) the comparison is different, since our method will require a large d_i, depending on the particular problem; in Example 1 we have $d_i \approx 30$.

Finally, note that the subproblems $S^{(i)}$ or $K^{(i)}$ on floating subdomains are still singular. To overcome the difficulty we use a solver (MUMPS) which is able to deal with rank deficient matrices and additionally may use a small regularization parameter α, replacing problems with $S^{(i)}$ by nonsingular problems with $S^{(i)} + \alpha I$.

4 Numerical experiments and discussion

The testcases have been provided by Parallab (Norway), Det Norske Veritas (Norway) and MSC.Software (Germany) within the European ESPRIT IV research project PARASOL. The convergence criterion has been to reduce the Euclidean norm of the residual by a factor of 10^6. In our convergence reports, **iter** denotes the number of iterations required to converge; by **cdps** we denote the average number of coarse degrees of freedom per subdomain, that is, **cdps** = (the order of the coarse matrix)/N. We have mostly experimented with the eigenvector based coarse space construction method described in Section 3. We refer to this method as to an `ADAPT` method. We have used 2 iterations of the inverse power method and a common threshold $\Lambda^{(i)}$ defined by (3) with $\theta = 10^{-3}$. To get more insight into the behavior of the eigenvector method, we have also used a simplified variant of this method, where the number d_i of the lowest eigenvectors included in the definition of the local coarse space $Z^{(i)}$ is *a priori* prescribed, using a common value $d_i = d$ fixed for all subdomains. We call this variant a `DETER(`d`)` algorithm. When possible, we compared our method with the crosspoint method, [9], denoted here by `CROSS`.

Example 2 (2D elasticity) In this case, we used a unit square piece of an elastic material with Young modulus $= 0.21 \cdot 10^{12}$, Poisson ratio 0.3 and density $7.85 \cdot 10^3$. The square was then decomposed automatically with METIS [7], resulting in subdomains containing hanging elements, so that $\dim \ker S^{(i)} \leq 4$. We iterated with `ADAPT`, `DETER()` with several parameters, and with the `CROSS` method. Since there are only displacement degrees of freedom, the crosspoint method we experimented with was exactly the same as the one proposed in [12]. Table 1 shows the number of iterations required by each method to converge.

As expected, `DETER(4)` converged in a small number of iterations, but `DETER(3)` encountered severe difficulties. The reason was that, contrary to `DETER(4)`, not all null space modes were resolved by `DETER(3)` option. The difference between

Table 1. 16x16 2D elasticity iterations (Example 2).

Partition	DETER(3)	DETER(4)	ADAPT	CROSS	first-level
k-way	-	19	16	-	103
recursive	61	20	16	75	99

Table 2. Iterations and coarse degrees of freedom per subdomain for Example 1.

Method	DETER(6)	DETER(28)	ADAPT	CROSS	CROSS+10^{-4}
iter	-	27	33	312	20
cdps	6	28	19.5	13.5	13.5

the number of iterations for different partitioning options can be partly explained by different shapes of the subdomains. The ADAPT method converged very well. It correctly detected the increase in the deficiency of $S^{(i)}$, which resulted in calculating a multiple of 3 local coarse vectors for some of the subdomains. Nevertheless, the average local coarse space dimension still remained low: we observed **cdps** = 5.

Let us notice the difference in the CROSS option behavior. In the case of recursive partitioning, we were lucky to get crosspoints at all vertices of the hanging element, so that it was taken care of at the coarse level. In the k-way case however, the null space mode introduced by a single hanging element could not be recognized by the CROSS option and was left on the first level, leading to a divergence of the method.

Example 1 – continued. We compared the convergence of the second-level Neumann-Neumann method for various coarse spaces. In this test case, every floating subdomain had 4 crosspoints in the corners, so it was fair to compare the CROSS option with the others. The performance of CROSS, DETER() and ADAPT coarse spaces is summarized in Table 2.

Observe that the Neumann-Neumann method with minimal coarse space, DETER(6), was not convergent. Good convergence rate was restored with quite a large coarse space DETER(28), where the number of the lowest eigenmodes moved to the second level was set equal to $28 = 6+22$ – that is, to the largest number of rigid body and intermediate modes per subdomain (see Figure 1). The adaptively constructed eigenvector based coarse space, ADAPT, still converged satisfactorily. Its dimension was even smaller than DETER(6), reflecting the fact that boundary subdomains had fewer null space modes to move up to the second level. It turned out that leaving one more intermediate mode per subdomain on the first level, roughly doubles the number of iterations.

On the other hand, notice the remarkable fact that using a carefully chosen[1] regularization parameter α (see Section 3.1), we were able to restore very good convergence properties of the `CROSS` method on this example.

Example 3 (Tubular joint) Let us end with a quite complex structural mechanics problem, on which the `ADAPT` method converges again without virtually any knowledge of the underlying problem geometry and discretization. The problem was a tubular joint testcase from Det Norske Veritas (100,000 unknowns) decomposed into 58 subdomains of varying shape and size. The model was discretized using both solid and shell finite elements, accompanied with special transitional elements between shells and solids. Moreover, subdomain stiffness/Schur complement matrices vary strongly in their size.

The `ADAPT` method was given only the partitioned stiffness matrices. No additional data (such as: element/unknowns types, decomposition details, geometry) was provided to the solver. Nevertheless, with the `ADAPT` method, the iteration converged within an impressive number of only 22 iterations. The coarse space constructed by the `ADAPT` method reached a reasonable size with **cdps** = 32. We also note that the distribution of the coarse degrees of freedom was quite non-uniform across the subdomains, ranging from 10 to 80.

The above test indicates that our method is able to deal with combined shell–transitional–solid element types, retaining good convergence properties.

5 Conclusions

Our new coarse space approach has been designed to increase the flexibility of the second-level Neumann-Neumann while preserving the efficiency. The coarse space is constructed through an algebraic process, using a relatively low cost eigenanalysis of the subdomain stiffness matrices. On several test examples, we have shown the potential strengths of the method, in particular, the ability to handle strange decompositions and to solve structural mechanics problems on a variety of finite elements using a minimum amount of information regarding the problem itself.

Usually, the new method's convergence rate under regular decomposition was comparable to that of custom methods. The potential weakness of the method is that, on certain shell problems, it leads to somewhat larger coarse spaces and thus does not scale well.

6 Acknowledgments

The authors wish to thank Prof. M.Dryja and Prof. J.Mandel for comments and discussion. They also want to thank the PARASOL project partners: A.C.Damhaug from DNV (Norway) and S.Mayer from MSC.Software (Germany) for providing

[1] Presently, we can do it only by the examination of the whole spectrum of S, which is obviously unacceptable in practical applications.

the test examples and excellent cooperation. We also thank Dr. J.Koster from Parallab (Norway) for making the latest version of the DD solver available. The second author's work was supported through the Universitetet i Bergen (Norway) and partially by the Polish Scientific Research Committee grant KBN 2P03A02116.

References

1. P. E. Bjørstad, J. Koster, and P. Krzyżanowski, *Domain decomposition solvers for large scale industrial finite element problems*, in Applied Parallel Computing. New Paradigms for HPC in Industry and Academia, T. Sorevik, F. Manne, R. Moe, and A. Gebremedhin, eds., vol. 1947 of Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 374–384. Proceedings of the 5th International Workshop, PARA2000 Bergen, Norway, June 18-20, 2000.
2. S. C. Brenner and L.-y. Sung, *Balancing domain decomposition for nonconforming plate elements*, Numer. Math., 83 (1999), pp. 25–52.
3. M. Brezina, C. Heberton, J. Mandel, and P. Vanek, *An iterative method with convergence rate chosen a priori*, Tech. Report 140, University of Colorado in Denver, USA, April 1999. An earlier version has also been presented at the 1998 Copper Mountain Conference on Iterative Methods.
4. Y.-H. De Roeck and P. Le Tallec, *Analysis and test of a local domain decomposition preconditioner*, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. Kuznetsov, G. Meurant, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, PA, 1991, pp. 112–128.
5. M. Dryja and O. B. Widlund, *Additive Schwarz methods for elliptic finite element problems in three dimensions*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations (Norfolk, VA, 1991), SIAM, Philadelphia, PA, 1992, pp. 3–18.
6. ———, *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Comm. Pure Appl. Math., 48 (1995), pp. 121–155.
7. G. Karypis and V. Kumar, *METIS – Serial Graph Partitioning Algorithm*. Available electronically via http://www-users.cs.umn.edu/~karypis/metis/metis.
8. P. Le Tallec, J. Mandel, and M. Vidrascu, *Balancing domain decomposition for plates*, in Domain decomposition methods in scientific and engineering computing (University Park, PA, 1993), Amer. Math. Soc., Providence, RI, 1994, pp. 515–524.
9. ———, *A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems*, SIAM J. Numer. Anal., 35 (1998), pp. 836–867.
10. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Methods Engrg., 9 (1993), pp. 233–241.
11. J. Mandel and M. Brezina, *Balancing domain decomposition for problems with large jumps in coefficients*, Math. Comp., 65 (1996), pp. 1387–1401.
12. J. Mandel and P. Krzyżanowski, *Robust Balancing Domain Decomposition*, August 1999. Presentation at The Fifth US National Congress on Computational Mechanics, University of Colorado at Boulder, CO.
13. B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain decomposition*, Cambridge University Press, Cambridge, 1996. Parallel multilevel methods for elliptic partial differential equations.