

## Nonoverlapping Domain Decomposition Algorithms for the $p$ -version Finite Element Method for Elliptic Problems

Ion Bică

### 1. Introduction

The nonoverlapping domain decomposition methods form a class of domain decomposition methods, for which the information exchange between neighboring subdomains is limited to the variables directly associated with the interface, i.e. those common to more than one subregion. Our objective is to design algorithms in  $3D$  for which we can find an upper bound on the *condition number*  $\kappa$  of the preconditioned linear system, which is independent of the number of subdomains and grows slowly with  $p$ . Here,  $p$  is the maximum degree of the polynomials used in the  $p$ -version finite element discretization of the continuous problem. In this paper, we survey some of the results obtained in [2].

Iterative substructuring methods for the  $h$ -version finite element,  $2D$   $p$ -version, and spectral elements have been previously developed and analyzed by several authors [3, 4], [6], [1],[13, 14], [11], [5], and [7, 8, 9].

However, some very real difficulties remained when the extension of these methods and their analysis to the  $3D$   $p$ -version finite element method were attempted, such as a lack of extension theorems for polynomials. The corresponding results are well known for Sobolev spaces, but their extension to finite element spaces is quite intricate. In our technical work, we use and further develop extension theorems for polynomials given in [1], [12], and [10] in order to prove the following bound on the condition number of our algorithm:

$$(1) \quad \kappa \leq C(1 + \log p)^4.$$

We believe that two logs can be dropped and a bound, similar to the ones in [3, 4], [6], and [13, 14], can be obtained.

In Section 2, we describe the model problem we are solving and the basis functions of the finite element space which are best suited for our algorithm. Section 3 contains a brief description of the preconditioner on which the algorithm is based. In Section 4, we compute the local and global condition numbers of our algorithm and make specific recommendations on the best choice of preconditioners.

---

1991 *Mathematics Subject Classification*. Primary 65N30; Secondary 41A10, 65N35, 65N55.

This work was supported in part by the National Science Foundation under Grants NSF-CCR-9503408 and in part by the U.S. Department of Energy under contract DE-FG02-92ER25127.

## 2. Continuous and discrete problems

We consider the following problem formulated variationally: Find  $u \in V$  such that

$$(2) \quad a(u, v) = \int_{\Omega} \rho(x) \nabla u \nabla v \, dx = f(v) \quad \forall v \in V.$$

Here,  $V$  is a subspace of  $H^1(\Omega)$ , determined by boundary conditions,  $\Omega$  is a polyhedral region triangulated with tetrahedra  $\bar{\Omega}_i$ ,  $\bar{\Omega} = \cup \bar{\Omega}_i$ . We denote by  $\Gamma$  the interface between subdomains,  $\Gamma = \cup \partial\Omega_i \setminus \partial\Omega$ . We assume that the boundary conditions are of the same type within each face of any tetrahedron that is part of the boundary. The coefficient  $\rho(x) > 0$  can be discontinuous across the interface between the subdomains, but varies only moderately within each  $\Omega_i$ . Without further decreasing the generality, we assume  $\rho(x) = \rho_i$  on  $\Omega_i$ . The bound (1) holds for arbitrary jumps in  $\rho_i$ .

We discretize the problem by the  $p$ -version finite element method. The finite element space  $V^p$  consists of the continuous functions on  $\Omega$  which are polynomials of total degree  $p$  in each  $\Omega_i$ . We end up with a system  $Kx = b$ , where the stiffness matrix  $K$  is built from local stiffness matrices, by subassembly. We will now define the basis functions on a reference tetrahedron  $\Omega_{ref}$ . There are many ways to do so, and a proper choice is the key to obtaining an efficient iterative method. We present here only a general description. We distinguish between four types of basis functions, associated with the vertices, edges, faces, and interior of  $\Omega_{ref}$ .

1. A *vertex basis function* has value one at a vertex and vanishes on the face opposite to that vertex. There is only one vertex function per vertex.
2. An *edge basis function* vanishes on the two faces which do not share the edge. The traces of the edge functions associated with the same edge are chosen to be linearly independent on that edge. There are  $p - 1$  such functions per edge.
3. A *face basis function* vanishes on the other three faces. The traces of the face functions associated with the same face are chosen to be linearly independent on that face. There are  $(p - 1)(p - 2)/2$  such functions per face.
4. An *interior basis function* vanishes on  $\partial\Omega_{ref}$ . There are  $(p - 1)(p - 2)(p - 3)/6$  interior functions and they are linearly independent.

The total number of vertex, edge, face, and interior functions is  $(p+1)(p+2)(p+3)/6$ . It is easy to see that they form a basis for  $P^p(\Omega_{ref})$ , the space of polynomials of total degree  $p$  on  $\Omega_{ref}$ . The union of the closed edges is the *wire basket* of  $\Omega_{ref}$ .

It turns out that if we use some standard vertex and edge functions [16], the preconditioned system that defines our algorithm is very ill conditioned; cf. [2, Section 5.1.1]. We therefore construct *low energy* vertex and edge functions; see [2, Chapter 4] which result in the bound (1). They satisfy certain stability properties related to their values on the edges and boundary of the reference tetrahedron. Their construction is based on the extension theorems in [1], [12], [10]. To avoid technical details here, we only remark that the low energy functions with highly oscillatory traces on the wire basket decay much more rapidly away from the edges than the standard ones that have the same trace. See Fig. 1 for a comparison of low energy and standard (high energy) basis functions.

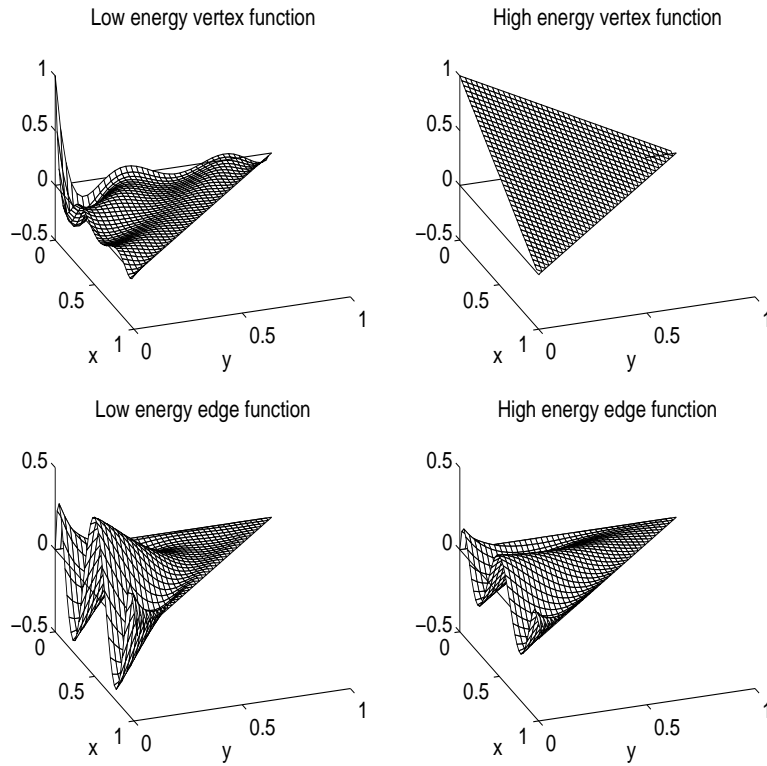


FIGURE 1. Basis functions on the face  $F_3$  of the reference tetrahedron  $\Omega_{ref}$

We next eliminate the interior degrees of freedom and end up with a system  $Sx_\Gamma = b_\Gamma$ , where  $S$  is built from local Schur complements, also by subassembly.

### 3. A wire basket algorithm

This algorithm is similar to the wire basket algorithm defined in [6, Section 6.2] and [13, 14, Section 6]. An interesting theoretical feature of this algorithm is that the bound on the condition number of the global preconditioned system is the same as the local one.

We use a preconditioned conjugate gradient algorithm, with the preconditioner built from blocks that correspond to subspaces. In its simplest form, this algorithm is a block-Jacobi method. We define it in the variational framework known as the abstract Schwarz theory; see, e.g., Smith, Bjørstad, and Gropp [15].

The coarse space  $V_W$  is the space spanned by the vertex and edge functions and contains the constants. The construction of such a space is quite intricate; cf [6, Section 6.2], [13, 14, Section 6], [2, Section 4.3].

All the local spaces are associated with individual faces of the tetrahedra. For each face  $F_k$ , we define the face space  $V_{F_k}$  as the space of functions in  $V^p$ , that vanish on all the faces of the interface  $\Gamma$  except  $F_k$ . We obviously have

$$V^p = V_W + \sum_k V_{F_k}.$$

TABLE 1. Local condition numbers, wire basket algorithm, low energy vertex and edge functions

$p$	Constants not in the wire basket			Constants in the wire basket		
	$\lambda_{min}$	$\lambda_{max}$	$\kappa$	$\lambda_{min}$	$\lambda_{max}$	$\kappa$
4	0.1921	1.8000	9.3691	0.1331	2.2549	16.9416
5	0.1358	1.7788	13.1022	0.1063	2.3890	22.4775
6	0.1033	1.8203	17.6186	0.0842	2.4503	29.1136
7	0.0864	1.8205	21.0818	0.0753	2.4996	33.2026
8	0.0740	1.8407	24.8854	0.0655	2.5374	38.7335
9	0.0656	1.8476	28.1508	0.0601	2.5668	42.6989
10	0.0590	1.8588	31.4892	0.0541	2.5911	47.9346

To each subspace  $V = V_W$  or  $V_{F_k}$ , we associate an operator  $T_V$  defined by

$$\hat{a}_V(T_V u, v) = a(u, v) \quad \forall v \in V.$$

Here,  $\hat{a}_V(\cdot, \cdot)$  is a positive definite, symmetric bilinear form on the subspace  $V$ .

Each bilinear form  $\hat{a}_V(\cdot, \cdot)$  uniquely defines the operator  $T_V$  and vice-versa. We say that  $T_V$  is an approximate projection. If  $\hat{a}_V(\cdot, \cdot) = a(\cdot, \cdot)$  then  $T_V = P_V$ , the  $a(\cdot, \cdot)$ -orthogonal projection on  $V$ . For specific subspaces, we choose  $T_V$  to be almost spectrally equivalent to  $P_V$ , but cheaper to compute.

On the coarse space, we can use the exact solver  $a(\cdot, \cdot)$  or, more economically, an inexact solver based on the bilinear form

$$(3) \quad \hat{a}_W(u, u) = (1 + \log p) \sum_i \inf_{c_i} \|u - c_i\|_{L^2(W_i)}^2.$$

On each face space, we choose  $\hat{a}_{F_k}(\cdot, \cdot) = a(\cdot, \cdot)$ .

The *additive Schwarz method* (ASM) is defined by the operator

$$(4) \quad T_a = T_W + T_{F_1} + \dots + T_{F_{nf}},$$

where  $nf$  is number of faces in  $\Gamma$ . The equation  $T_a u = g_a$ , where  $g_a = T_W u + T_{F_1} u + \dots + T_{F_{nf}} u$  can be solved by the conjugate gradient method, which can be viewed as a preconditioned conjugate gradient method for the initial system  $Sx_\Gamma = b_\Gamma$ .

The preconditioner for the additive Schwarz method, using the exact solver on  $V_W$ , has the following matrix form:

$$(5) \quad S_{prec} = \begin{pmatrix} S_{WW} & 0 & 0 & 0 \\ 0 & S_{F_1 F_1} & 0 & 0 \\ 0 & 0 & S_{F_2 F_2} & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix}.$$

If we use the inexact solver  $\hat{a}_W(\cdot, \cdot)$ , the block  $S_{WW}$  is replaced by

$$\hat{S}_{WW} = (1 + \log p) \left( M - \sum_i \frac{(M^{(i)} z^{(i)}) \cdot (M^{(i)} z^{(i)})^T}{z^{(i)T} M^{(i)} z^{(i)}} \right),$$

where  $M^{(i)}$  is the mass matrix of the wire basket  $W_i$ , and  $z^{(i)}$  is the vector containing the coefficients of the constant function 1. The mass matrix  $M$ , for our particular choice of vertex and edge functions, is tridiagonal.

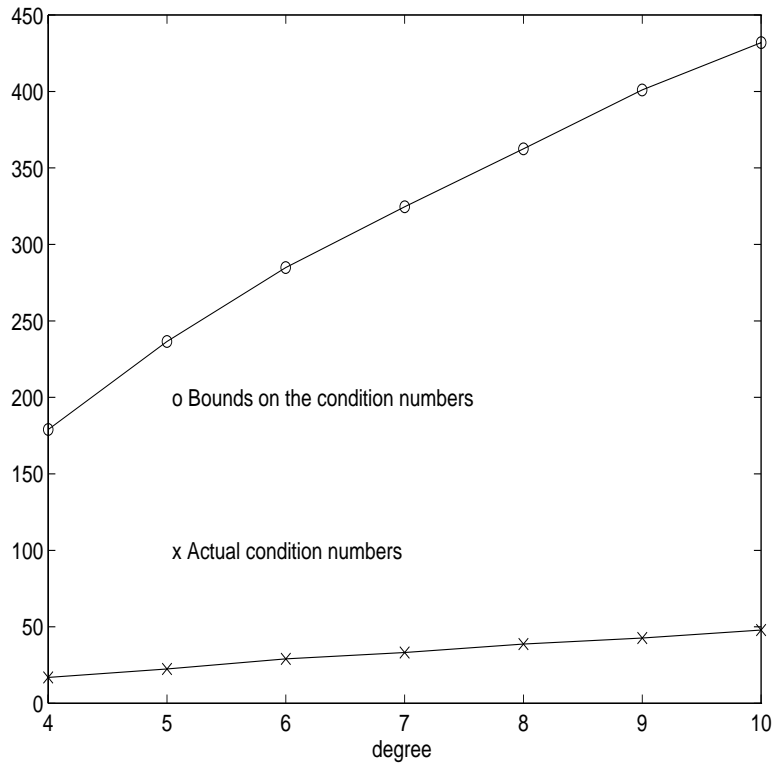


FIGURE 2. Condition numbers and bounds on them, for the wire basket algorithm

The *symmetrized multiplicative method* (MSM) is defined by the operator

$$T_m = I - (I - T_W)(I - T_{F_1}) \cdots (I - T_{F_{n_f}}) \cdots (I - T_{F_1})(I - T_W).$$

The *hybrid method* (HSM) is defined by the operator

$$T_h = I - (I - T_W)(I - T_{F_1} - \cdots - T_{F_{n_f}})(I - T_W).$$

The matrix form of the preconditioners defined by the operators  $T_m$  and  $T_h$  is not block-diagonal.

#### 4. Numerical experiments

We start by computing the local condition number of the preconditioned Schur complement, on a reference tetrahedron; see Table 1. We obtain lower condition numbers in case when the constants are not in the coarse space space. However, as we mentioned in Section 3, we must add the constants to the coarse space. We do this at the expense of increasing the condition numbers by a factor of 1.5 – 1.6. Next, we look at the bounds on these condition numbers, as given by the theory. To this end, we compute all the constants in the inequalities used in the proof (1); see [2, Section 5.1.2]. The asymptotic logarithmic growth of these bounds is more visible than that of the actual condition numbers; see Fig. 2.

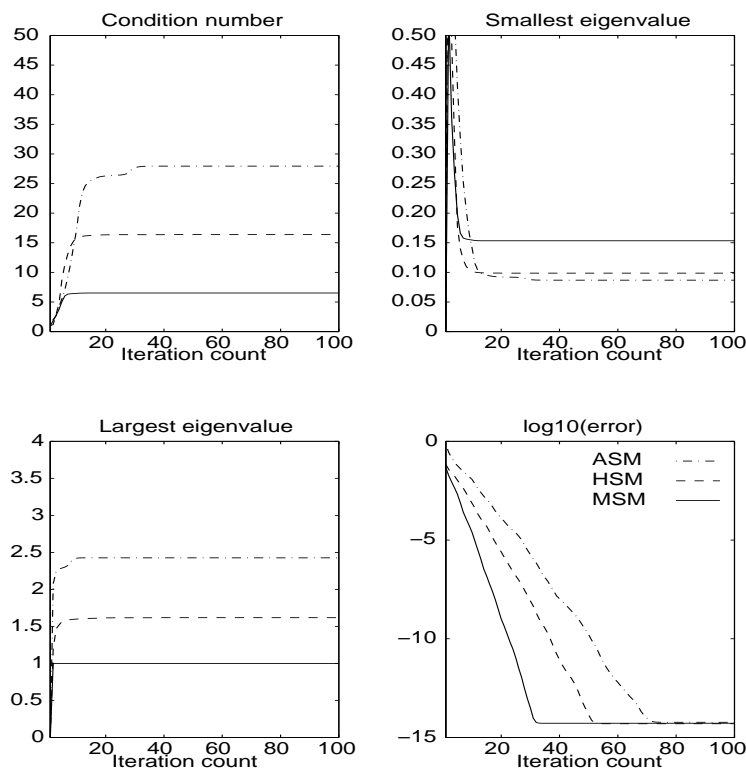


FIGURE 3. Comparison of additive, hybrid, and multiplicative methods,  $p = 6$ , exact solver on the wire basket.

We now move to global experiments. The number of iterations is fixed beforehand. We compare the performances of the additive, hybrid, and multiplicative methods, see Fig. 3. The extreme eigenvalues are computed via the Lanczos iteration. We have performed experiments on a cubic region that consists of 192 identical tetrahedra, for  $p = 6$ , with Dirichlet boundary conditions on one face of  $\Omega$ , and Neumann on the others. We have used the exact solver on the wire basket. We remark that the global condition number of the additive method coincides with the local one, given in Table 1, at the intersection of the last column and the row that corresponds to  $p = 6$ . We remark that the multiplicative method performs better than the hybrid method, which performs better than the additive one. We can use an inexact solver on the wire basket, which makes the coarse problem cheaper to solve, at some expense in the performance of the full algorithm; see [2, Section 5.2.3].

## References

1. Ivo Babuška, Alan Craig, Jan Mandel, and Juhani Pitkäranta, *Efficient preconditioning for the  $p$ -version finite element method in two dimensions*, SIAM J. Numer. Anal. **28** (1991), no. 3, 624–661.
2. Ion Bica, *Iterative substructuring methods for the  $p$ -version finite element method for elliptic problems*, Ph.D. thesis, New York University, September 1997, Tech. Rep. 1997-743.

3. James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp. **47** (1986), no. 175, 103–134.
4. ———, *The construction of preconditioners for elliptic problems by substructuring, IV*, Math. Comp. **53** (1989), 1–24.
5. Mario A. Casarin, *Diagonal edge preconditioners in  $p$ -version and spectral element methods*, Tech. Report 704, Department of Computer Science, Courant Institute, September 1995, To appear in SIAM J. Sci. Comp.
6. Maksymilian Dryja, Barry F. Smith, and Olof B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal. **31** (1994), no. 6, 1662–1694.
7. Benqi Guo and Weiming Cao, *A preconditioner for the  $h$ - $p$  version of the finite element method in two dimensions*, Numer. Math. **75** (1996), 59–77.
8. ———, *An additive Schwarz method for the  $hp$ -version finite element method in three dimensions*, SIAM J. Sci. Comp. (To appear.).
9. ———, *Preconditioning for the  $h$ - $p$  version of the finite element method in two dimensions*, SIAM J. Numer. Analysis. (To appear).
10. Yvon Maday, *Relèvement de traces polynomiales et interpolations Hilbertiennes entre espaces de polynômes*, C. R. Acad. Sci. Paris **309**, Série I (1989), 463–468.
11. Jan Mandel, *Two-level domain decomposition preconditioning for the  $p$ -version finite element version in three dimensions*, Int. J. Numer. Meth. Eng. **29** (1990), 1095–1108.
12. Rafael Munoz-Sola, *Polynomial liftings on a tetrahedron and applications to the  $h$ - $p$  version of the finite element method in three dimensions*, SIAM J. Numer. Anal. **34** (1997), no. 1, 282–314.
13. Luca F. Pavarino and Olof B. Widlund, *A polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions*, SIAM J. Numer. Anal. **33** (1996), no. 4, 1303–1335.
14. ———, *Iterative substructuring methods for spectral elements: Problems in three dimensions based on numerical quadrature*, Computers Math. Applic. **33** (1997), no. 1/2, 193–209.
15. Barry F. Smith, Petter Bjørstad, and William Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.
16. Barna Szabó and Ivo Babuška, *Finite element analysis*, John Wiley & Sons, New York, 1991.

COURANT INSTITUTE, NEW YORK UNIVERSITY

*Current address:* Schlumberger-Doll Research, Old Quarry Road, Ridgefield, CT 06877

*E-mail address:* bica@ridgefield.sdr.slb.com