Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

38. A preconditioner for the Schur complement domain decomposition method

J.-M. Cros [1]

1. Introduction. This paper presents a preconditioner for the Schur complement domain decomposition method inspired by the dual-primal FETI method [4]. Indeed the proposed method enforces the continuity of the preconditioned gradient at cross-points directly by a reformulation of the classical Neumann-Neumann preconditioner. In the case of elasticity problems discretized by finite elements, the degrees of freedom corresponding to the cross-points coming from domain decomposition, in the stiffness matrix, are separated from the rest. Elimination of the remaining degrees of freedom results in a Schur complement matrix for the cross-points. This assembled matrix represents the coarse problem. The method is not mathematically optimal as shown by numerical results but its use is rather economical. The paper is organized as follows: in sections 2 and 3, the Schur complement method and the formulation of the Neumann-Neumann preconditioner are briefly recalled to introduce the notations. Section 4 is devoted to the reformulation of the Neumann-Neumann preconditioner. In section 5, the proposed method is compared with other domain decomposition methods such as generalized Neumann-Neumann algorithm [7][9], one-level FETI method [5] and dual-primal FETI method. Performances on a parallel machine are also given for structural analysis problems.

2. The Schur complement domain decomposition method. Let Ω denote the computational domain of an elasticity problem. Consider a symmetric and positive definite linear system obtained by finite element discretization of the equations of equilibrium:

$$K\ u = f, \tag{2.1}$$

with the stiffness matrix K, the vector of degrees of freedom u, and the right-hand side f. The original domain Ω is partioned into n_s non-overlapping subdomains Ω^s. Let K^s be the local stiffness matrix and u^s the vector of degrees of freedom corresponding to subdomain Ω^s. Let N^s denote the Boolean matrix mapping the degrees of freedom u^s into global degrees of freedom u:

$$u^s = {N^s}^T\ u. \tag{2.2}$$

Then the stiffness matrix is obtained by the standard assembly process:

$$K = \sum_{s=1}^{n_s} N^s\ K^s\ {N^s}^T. \tag{2.3}$$

The union of all boundaries between subdomains is Γ such that $\Gamma = \cup_{s=1}^{n_s} \Gamma^s$ with $\Gamma^s = \partial\Omega^s \backslash \partial\Omega$. For each subdomain the total set of degrees of freedom is then split into two subsets, the interface degrees of freedom u_b^s associated with Γ^s and the other degrees of freedom u_i^s of the subdomain Ω^s. After this partition, the subdomain stiffness matrix, displacement vector, right-hand side and Boolean matrix take the following form:

$$K^s = \begin{bmatrix} K_{ii}^s & K_{ib}^s \\ {K_{ib}^s}^T & K_{bb}^s \end{bmatrix},\ \{u^s\} = \begin{Bmatrix} u_i^s \\ u_b^s \end{Bmatrix},\ \{f^s\} = \begin{Bmatrix} f_i^s \\ f_b^s \end{Bmatrix},\ \text{and } N^s = [N_i^s\ N_b^s]. \tag{2.4}$$

[1] Université d'ÉVRY-VAL d'ESSONNE, jean-michel.cros@iup.univ-evry.fr

With this notation, the linear system (2.1) takes the form:

$$\begin{bmatrix} K_{ii}^1 & \cdots & 0 & K_{ib}^1 {N_b^1}^T \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & K_{ii}^{n_s} & K_{ib}^{n_s} {N_b^{n_s}}^T \\ N_b^1 {K_{ib}^1}^T & \cdots & N_b^{n_s} {K_{ib}^{n_s}}^T & \sum_{s=1}^{n_s} N_b^s K_{bb}^s {N_b^s}^T \end{bmatrix} \begin{Bmatrix} u_i^1 \\ \vdots \\ u_i^{n_s} \\ u_b \end{Bmatrix} = \begin{Bmatrix} f_i^1 \\ \vdots \\ f_i^{n_s} \\ \sum_{s=1}^{n_s} N_b^s \; f_b^s \end{Bmatrix}. \tag{2.5}$$

After elimination of the interior degrees of freedom, the problem (2.5) reduces to a problem (2.6) posed on the interface Γ:

$$\left(\sum_{s=1}^{n_s} N_b^s \left(K_{bb}^s - {K_{ib}^s}^T {K_{ii}^s}^{-1} K_{ib}^s \right) {N_b^s}^T \right) u_b = \sum_{s=1}^{n_s} N_b^s \left(f_b^s - {K_{ib}^s}^T {K_{ii}^s}^{-1} f_i^s \right). \tag{2.6}$$

Defining the global Schur complement matrix S by:

$$S = \sum_{s=1}^{n_s} N_b^s \; S^s \; {N_b^s}^T \tag{2.7}$$

where the local Schur complement matrix is given by $S^s = K_{bb}^s - {K_{ib}^s}^T {K_{ii}^s}^{-1} K_{ib}^s$. The linear system (2.6) is solved iteratively without assembling S, using a preconditioned conjugate gradient algorithm.

3. Neumann-Neumann preconditioners.

For mechanical problems, the most classical preconditioner used is the Neumann-Neumann method [1][7]. The preconditioner (3.1) is defined by approximating the inverse of the sum of local Schur complement matrices by the weighted sum of the inverses:

$$z = M\, r = \sum_{s=1}^{n_s} N_b^s \; D^s \; {S^s}^{-1} \; D^s \; {N_b^s}^T \; r, \tag{3.1}$$

where r is the conjugate gradient and z is the preconditioned conjugate gradient. For convergence reasons [7], the diagonal weight matrices D^s must verify:

$$\sum_{s=1}^{n_s} N_b^s \; D^s \; {N_b^s}^T = I_\Gamma. \tag{3.2}$$

However, the convergence rate decreases rapidly for a large number of subdomains. Then, the balancing domain decomposition method [8] includes a coarse problem in order to reduce significantly this dependence on the number of subdomains. The balancing method or the generalized Neumann-Neumann preconditioner [7] writes:

$$M = \left(I - G \, [G^T \; S \; G]^{-1} \; G^T S \right) \sum_{s=1}^{n_s} N_b^s \; D^s \; {\tilde{S}^s}{}^{-1} \; D^s \; {N_b^s}^T, \tag{3.3}$$

$$\text{with } G = [N_b^1 D^1 Z^1, ..., N_b^{n_f} D^{n_f} Z^{n_f}], \tag{3.4}$$

where n_f is the number of floating subdomains (subdomain without natural Dirichlet condition), the block matrices Z^s are boundary values of subdomain solutions with restriction of rigid body modes on Γ^s, and ${\tilde{S}^s}{}^{-1}$ is the pseudo inverse of the local Schur complement matrix. The method has been extended [9] for second or fourth order elasticity problems, by using corner modes. By definition, a corner or a cross-point is a node belonging to more than two subdomains and also, for plate and shell problems, a node located at the beginning and the end of each edge of each subdomain. Then, the block matrices Z^s are boundary values of subdomain solutions with successively one degree of freedom fixed to one at one corner, all other corner degrees of freedom fixed to zero. The method is efficient, but the coarse matrix $[G^T\ S\ G]$ is costly to build because it involves a product with S.

4. A new coarse problem. We propose to build a Neumann-Neumann preconditioner by enforcing a continuous field at the cross-points. In the classical Neumann-Neumann preconditioner with or without coarse problem, the field is continuous only by averaging the contributions of each subdomain at the cross-points. Then, we introduce a new partitioning (figure 4.1), by splitting u^s into two sub-vectors (4.1) where u_c is a global solution vector over all defined corner degrees of freedom and u_r^s is the remainder subdomain solution vector.

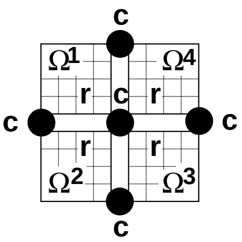

Figure 4.1: Mesh partition : corner (c) and remainder nodes (r)

$$\{u\} = \left\{ \begin{array}{c} u_r \\ u_c \end{array} \right\} = \left\{ \begin{array}{c} u_r^1 \\ \vdots \\ u_r^{n_s} \\ u_c \end{array} \right\}. \tag{4.1}$$

The Boolean matrix N_c^s (4.2) maps the local corner equation to the global corner equation:

$$u_c^s \quad = \quad N_c^{s^T}\ u_c. \tag{4.2}$$

Then, we introduce new Boolean matrices (4.3) which extract from the interface Γ^s, the cross-points and the remainder unknowns:

$$\{u^s\} = \left\{ \begin{array}{c} u_r^s = [R_{ri}^s\ \ R_{rb}^s] \left\{ \begin{array}{c} u_i^s \\ u_b^s \end{array} \right\} \\ u_c^s = R_c^s\ u_b^s \end{array} \right\}. \tag{4.3}$$

According to this new partition of the degrees of freedom, the preconditioned gradient is the restriction on Γ of the solution of problem (4.4) with subdomains connected by the cross-points as shown in figure 4.1:

$$\left[\begin{array}{cccc} K_{rr}^1 & \cdots & 0 & K_{rc}^1 N_c^{1^T} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & K_{rr}^{n_s} & K_{rc}^{n_s} N_c^{n_s^T} \\ N_c^1 K_{rc}^{1^T} & \cdots & N_c^{n_s} K_{rc}^{n_s^T} & \sum_{s=1}^{n_s} N_c^s K_{cc}^s N_c^{s^T} \end{array} \right] \left\{ \begin{array}{c} u_r^1 \\ \vdots \\ u_r^{n_s} \\ u_c \end{array} \right\} = \left\{ \begin{array}{c} f_r^1 \\ \vdots \\ f_r^{n_s} \\ \sum_{s=1}^{n_s} N_c^s\ f_c^s \end{array} \right\}, \tag{4.4}$$

with $f_r^s = R_{rb}^s \, D^s \, N_b^{s^T} r$ (with $f_i^s = 0$) and $f_c^s = R_c^s \, D^s \, N_b^{s^T} r$. It is noted that by definition of D^s, the quantity $\sum_{s=1}^{n_s} N_c^s \, f_c^s = \sum_{s=1}^{n_s} N_c^s \, R_c^s \, D^s \, N_b^{s^T} r$ is the restriction of gradient r at the cross-points (r_c). Thus the solution of (4.4) is written:

$$u_r^s \, = \, K_{rr}^{s^{-1}} \, \big(f_r^s - K_{rc}^s \, N_c^{s^T} \, u_c\big), \tag{4.5}$$

$$u_c \, = \, \Big(\sum_{s=1}^{n_s} \, N_c^s \big(K_{cc}^s - K_{rc}^{s^T} K_{rr}^{s^{-1}} K_{rc}^s\big) \, N_c^{s^T} \Big)^{-1} \, \sum_{s=1}^{n_s} \, N_c^s \, \Big(\, f_c^s - K_{rc}^{s^T} K_{rr}^{s^{-1}} \, f_r^s \, \Big). \tag{4.6}$$

Finally, the preconditioned gradient is given by:

$$z = Mr = \sum_{s=1}^{n_s} N_b^s D^s R_c^{s^T} N_c^{s^T} u_c + \sum_{s=1}^{n_s} N_b^s D^s R_{rb}^{s^T} u_r^s, \tag{4.7}$$

and the proposed preconditioner takes the form:

$$\begin{aligned} M = & \sum_{s=1}^{n_s} N_b^s D^s \Big(N_c^s (R_c^s - K_{rc}^{s^T} K_{rr}^{s^{-1}} R_{rb}^s) \Big)^T \; S_c^{-1} \; \sum_{s=1}^{n_s} \Big(N_c^s \, (R_c^s - K_{rc}^{s^T} K_{rr}^{s^{-1}} R_{rb}^s) \Big) D^s N_b^{s^T} \\ & + \sum_{s=1}^{n_s} N_b^s D^s \, \Big[R_{rb}^{s^T} K_{rr}^{s^{-1}} \, R_{rb}^s \Big] \, D^s N_b^{s^T}. \end{aligned} \tag{4.8}$$

The first term is a coarse problem which couples all subdomains. We suppose that each subdomain owns enough cross-points to have local Neumann problems (4.5) well posed, otherwise artificial cross-points are added. The coarse matrix is built easily by forming the matrices S_c^s in each subdomain and by assembling S_c:

$$S_c = \sum_{s=1}^{n_s} N_c^s \, \Big[K_{cc}^s - K_{rc}^{s^T} K_{rr}^{s^{-1}} K_{rc}^s \Big] \, N_c^{s^T} = \sum_{s=1}^{n_s} N_c^s \; S_c^s \; N_c^{s^T}. \tag{4.9}$$

In comparison with coarse problem of the balancing method, the size of the coarse problem S_c (equals to the number of degrees of freedom per node multiplied by the total number of cross-points) is small, because of the definition of corner modes.

5. Numerical results.

The parallel implementation of the different methods has been developed within message passing programming environment. Each subdomain is allocated to one processor. All coarse problems are assembled and solved by a skyline solver during the iterations of the preconditioned conjugate gradient algorithm.

In all the tables below, GNN denotes the Neumann-Neumann preconditioner with coarse grid solver based on rigid body modes (RBM) [8] or corner modes (CM) [9], NN+C is the proposed Neumann-Neumann preconditioner with coarse grid solver, FETI-DP is a dual-primal Finite Element Tearing and Interconnecting method [4] and FETI-1 is a classical FETI method [5]. In the absence of other specification, the FETI methods are equipped with the Dirichlet preconditioner. The stopping criterion to monitor the convergence is the same in all cases presented and it is related to the global residual:

$$\parallel K \, u - f \parallel \, / \parallel f \parallel \;\; \leq \;\; 10^{-6}. \tag{5.1}$$

We investigate the numerical scalability of the proposed method with respect to the mesh size h and to the number of subdomains n_s. For this purpose, we consider a cylindrical shell roof (figure 5.1) subjected to a loading of its own weight. The roof is supported by walls at each end and is free along the sides. For symmetry reasons only a quarter of the roof is considered and is meshed with 3-node shell elements. We study three discretizations denoted respectively by h (3,750 degrees of freedom), $h/2$ (14,406 degrees of freedom) and $h/4$ (56,454 degrees of freedom). The meshes $h/2$ et $h/4$ are obtained from the first one by global regular refinement.

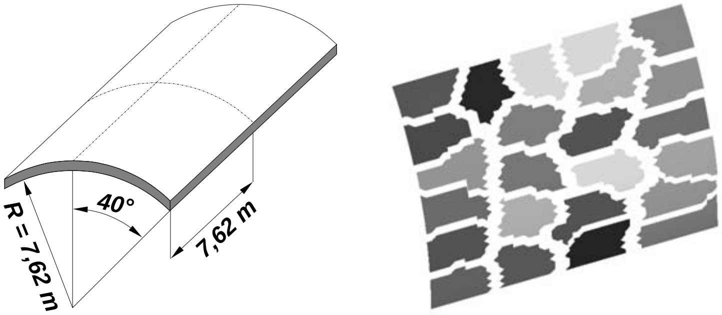

Figure 5.1: A cylindrical shell roof, mesh $h/2$ decomposed into 24 subdomains

First, the three meshes were decomposed into 24 subdomains, we note (figure 5.1) that automatic decomposition by METIS [6] induces rugged interfaces. The results were obtained on a Origin 2000 system (64 processors) of "Pôle Parallélisme Île de France Sud". We report (table 5.1) the number of iterations to converge for the different methods and in brackets the size of the coarse problem. The number of iterations remains roughly constant for the different methods (thickness = 0.1 m and 0.01 m). However all the methods are sensitive to the small thickness of the roof, and especially the FETI-DP method and the NN+C method.

Table 5.1: A cylindrical shell roof, numerical scalability, $n_s = 24$

MESH	THICKNESS (m)	GNN (CM)	NN+C	FETI-DP
h	0.1	30 iter. (726)	42 iter. (270)	44 iter. (270)
	0.01	31 iter.	48 iter.	50 iter.
	0.001	54 iter.	99 iter.	106 iter.
$h/2$	0.1	36 iter. (714)	41 iter. (270)	44 iter. (270)
	0.01	37 iter.	48 iter.	50 iter.
	0.001	45 iter.	74 iter.	80 iter.
$h/4$	0.1	39 iter. (738)	40 iter. (276)	45 iter. (276)
	0.01	39 iter.	44 iter.	47 iter.
	0.001	42 iter.	67 iter.	73 iter.

The second test (table 5.2) consists in fixing the size of the problem ($h/4$, 56,454 degrees of freedom, thickness = 0.1 m) but we change the number of subdomains (12, 24, 48). The CPU times are reported (table 5.2) for both the preparation step (finite element operations, building of coarse problem,...) and for the solution. It appears clearly that the building of coarse problem takes a large part of the cpu time for GNN method. The two other methods have a good speed-up.

We consider now the modal analysis of a plate (1×1 m) embeded on one side. The problem is discretized in 10,086 degrees of freedom with 3-node shell elements. The mesh is partitioned into 20 subdomains. The two lowest eigenmodes are obtained in five iterations of the subspace iteration method. The conjugate gradient method with restart technique [2][5] is used to deal

Table 5.2: A cylindrical shell roof, parallel scalability, mesh $h/4$

n_s	INTERFACE	METHOD	ITERATION	CPU (sec.)
12	3,144	GNN (CM)	38 iter. (426)	$46.22 + 9.81 = 56.03$
		NN+C	40 iter. (216)	$6.42 + 10.38 = 16.80$
		FETI-DP	45 iter. (216)	$6.38 + 12.79 = 19.17$
24	4,818	GNN (CM)	39 iter. (738)	$18.82 + 5.30 = 24.12$
		NN+C	40 iter. (276)	$1.59 + 5.21 = 6.80$
		FETI-DP	45 iter. (276)	$1.52 + 5.45 = 6.97$
48	7,295	GNN (CM)	45 iter. (1602)	$18.67 + 8.18 = 26.85$
		NN+C	51 iter. (630)	$0.81 + 3.21 = 4.02$
		FETI-DP	56 iter. (630)	$0.86 + 3.64 = 4.50$

with successive and multiple right-hand sides. This technique is based on the exploitation of previously computed conjugate directions. Figure 5.2 shows the iteration history with respect to the number of right-hand sides using different methods, and we report also the total number of iterations, the size of the coarse problem (in brackets) and the CPU times. The GNN (CM) method converges quickly but the cost of one iteration is more important than the other methods, because of the large size of the coarse problem. Similar results are obtained for transient analysis. In addition, the solution of time-dependent problems by the implicit Newmark algorithm calls for successive solution of the linear system with the same matrix $[M + \beta\,\Delta t^2\ K]$. In this case, there are no longer floating subdomains due to the inertia term. Then, building a coarse grid based on the rigid body modes of stiffness matrices K^s becomes costly [3]. While the methods using the corner modes are not affected by this shifting of M.

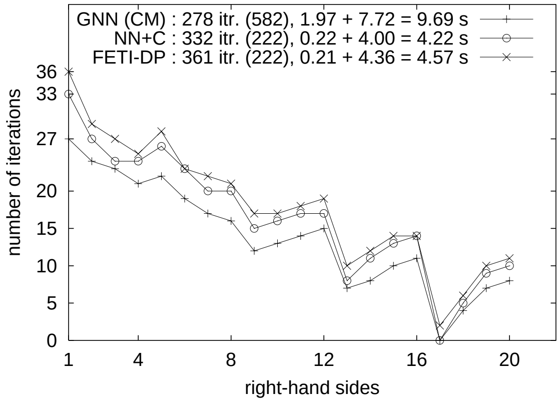

Figure 5.2: A shell problem, modal analysis

Another test example concerns a plane stress problem with a square (1×1 m) embeded on one side and subjected to a distributed load on the opposite side. The problem is discretized in 20,402 degrees of freedom with 3-node elements (101×101 nodes). The mesh is partitioned into 14 and 28 subdomains. The NN+C method is proved (table 5.3) to be efficient for this kind of problems.

Finally, we consider a three-dimensional cantilever beam (4×4×40 m) subjected to a

Table 5.3: 2D elasticity problem, 3-node elements, 101×101 nodes, 20,402 d.o.f.

n_s	GNN (RBM)	GNN (CM)	NN+C	FETI-DP	FETI-1
14	24 iter. (30)	14 iter. (132)	20 iter. (52)	22 iter. (52)	24 iter. (30)
28	26 iter. (72)	14 iter. (286)	23 iter. (108)	24 iter. (108)	26 iter. (72)

bending load. The finite element discretization is done with 8-node brick elements (12×12×76 nodes, 32,832 degrees of freedom). The beam (figure 5.3) is cut into 20 and 40 subdomains.

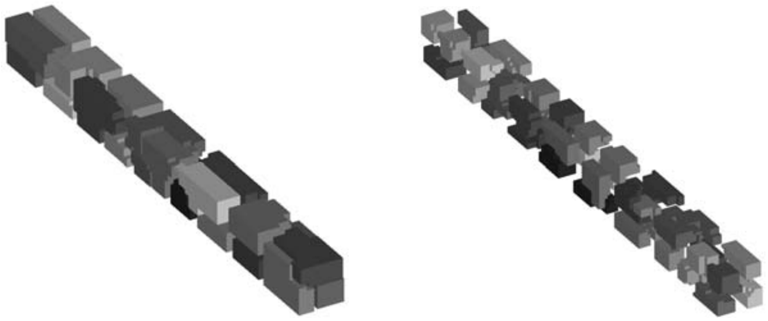

Figure 5.3: A cantilver beam, 20 subdomains (interface 7,614 d.o.f.) and 40 subdomains (interface 10,262 d.o.f.)

Table 5.4: Cantiler beam, 32,832 d.o.f.

			FETI-DP		FETI-1	
n_s	GNN (RBM)	NN+C	DIRICHLET	LUMPED	DIRICHLET	LUMPED
20	44 iter.	30 iter.	31 iter.	60 iter.	43 iter.	67 iter.
	(102)	(852)			(102)	
40	54 iter.	30 iter.	30 iter.	53 iter.	58 iter.	77 iter.
	(216)	(1968)			(216)	

For this analysis, the FETI methods use equally the lumped preconditioner. Table 5.4 summarizes the results with the size of the coarse problem (in brackets). Methods using corner modes have the best convergence rate, but the size of the coarse problem is very large (almost 20% of the interface for 40 subdomains). This size can be reduced easily. In fact, with brick elements, the number of cross-points can be chosen just enough to remove the singularities in subdomains.

6. Conclusion. In this paper, we have presented a modified Neumann-Neumann preconditioner validated by several examples. The results suggest that the proposed method (NN+C method) is numerically scalable with respect to the number of subdomains and to the mesh size. On the representative examples considered the NN+C method has the same performance as the FETI-DP method. Moreover, from the viewpoint of CPU time, the proposed method outperforms the optimal but expensive GNN preconditioner. However, the results depend largely on the implementation of the algorithm for solving the coarse problem.

REFERENCES

[1] J.-F. Bourgat, R. Glowinski, P. Le Tallec, and M. Vidrascu. Variational formulation and algorithm for trace operator in domain decomposition calculations. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Domain Decomposition Methods*, pages 3–16, Philadelphia, PA, 1989. SIAM.

[2] J.-M. Cros and F. Léné. Parallel iterative methods to solve large-scale eigenvalue problems in structural dynamics. In P. E. Bjørstad, M. Espedal, and D. Keyes, editors, *Domain Decomposition Methods in Sciences and Engineering*, pages 318–324. John Wiley & Sons, 1997. Proceedings from the Ninth International Conference, June 1996, Bergen, Norway.

[3] C. Farhat, P.-S. Chen, and J. Mandel. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems. *Int. J. Numer. Meth. Eng.*, 38:3831–3853, 1995.

[4] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Lin. Alg. Appl.*, 7:687–714, 2000.

[5] C. Farhat and F.-X. Roux. Implicit parallel processing in structural mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 2 (1), pages 1–124. North-Holland, 1994.

[6] G. Karypis and V. Kumar. Metis, unstructured graph partitioning and sparse matrix ordering system. version 2.0. Technical report, University of Minnesota, Department of Computer Science, Minneapolis, MN 55455, August 1995.

[7] P. Le Tallec. Domain decomposition methods in computational mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 1 (2), pages 121–220. North-Holland, 1994.

[8] J. Mandel. Balancing domain decomposition. *Comm. Numer. Meth. Engrg.*, 9:233–241, 1993.

[9] P. L. Tallec, J. Mandel, and M. Vidrascu. A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems. *SIAM J. Numer. Math.*, 35:836–867, 1998.