Parallel scalability of a FETI–DP mortar method for problems with discontinuous coefficients

Nina Dokeva and Wlodek Proskurowski

University of Southern California, Department of Mathematics
Los Angeles, CA 90089–1113, USA; {dokeva,proskuro}@usc.edu

Summary. We consider elliptic problems with discontinuous coefficients discretized by FEM on non-matching triangulations across the interface using the mortar technique. The resulting discrete problem is solved by a FETI–DP method using a preconditioner with special scaling described in Dokeva, Dryja and Proskurowski [to appear]. Experiments performed on hundreds of processors show that this FETI–DP mortar method exhibits good parallel scalability.

1 Introduction

Parallelization of finite element algorithms enables one to solve problems with a large amount of degrees of freedom in a reasonable time, which becomes possible if the method is scalable.

We adopt here the definition of scalability following Bhardwaj et al. [2000] and Bhardwaj et al. [2002]: solving n-times larger problem using an n-times larger number of processors in nearly constant cpu time. Domain decomposition algorithms using FETI-DP solvers (Farhat et al. [2001], Farhat et al. [2000], Klawonn, Widlund and Dryja [2002], Mandel and Tezaur [2001]) have been demonstrated to provide scalable performance on massively parallel processors, see Bhardwaj et al. [2002] and the references therein.

The aim of this paper is to experimentally demonstrate that a scalable performance on hundreds of processors can be achieved for a mortar discretization using FETI-DP solvers described in Dokeva, Dryja and Proskurowski [to appear] and Dryja and Widlund [2002].

In view of the page limitation, Section 2 describing the FETI-DP method and preconditioner is abbreviated to a minimum. For a complete presentation refer to Dokeva, Dryja and Proskurowski [to appear]. Section 3 contains the main results.

2 FETI-DP equation and preconditioner

We consider the following differential problem.

Find $u^* \in H_0^1(\Omega)$ such that

$$a(u^*, v) = f(v), \quad v \in H_0^1(\Omega), \tag{1}$$

where

$$a(u, v) = (\rho(x)\nabla u, \nabla u)_{L^2(\Omega)}, \quad f(v) = (f, v)_{L^2(\Omega)}.$$

We assume that Ω is a polygonal region and $\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}_i$, Ω_i are disjoint polygonal subregions, $\rho(x) = \rho_i$ is a positive constant on Ω_i and $f \in L^2(\Omega)$. We solve (1) by the FEM on non–matching triangulation across $\partial\Omega_i$. To describe a discrete problem the mortar technique is used.

We impose on Ω_i a triangulation with triangular elements and parameter h_i. The resulting triangulation in Ω is non-matching across $\partial\Omega_i$. Let $X_i(\Omega_i)$ be a finite element space of piecewise linear continuous functions defined on the introduced triangulation. We assume that functions of $X_i(\Omega_i)$ vanish on $\partial\Omega_i \cap \partial\Omega$.

Let $X^h(\Omega) = X_1(\Omega_1) \times \ldots \times X_N(\Omega_N)$ and $V^h(\Omega)$ be a subspace of $X^h(\Omega)$ of functions which satisfy the mortar condition

$$\int_{\delta_m} (u_i - u_j)\psi ds = 0, \quad \psi \in M(\delta_m). \tag{2}$$

Here, $u_i \in X_i(\Omega_i)$ and $u_j \in X_j(\Omega_j)$ on Γ_{ij}, an edge common to Ω_i and Ω_j and $M(\delta_m)$ is a space of test (mortar) functions.

Let $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ be a common edge of two substructures Ω_i and Ω_j. Let Γ_{ij} as an edge of Ω_i be denoted by $\gamma_{m(i)}$ and called *mortar* (master), and let Γ_{ij} as an edge of Ω_j be denoted by $\delta_{m(j)}$ and called *non-mortar* (slave). Denote by $W_j\left(\delta_{m(j)}\right)$ the restriction of $X_j(\Omega_j)$ to $\delta_{m(j)}$.

Using the nodal basis functions $\varphi^{(l)}_{\delta_{m(i)}} \in W_i\left(\delta_{m(i)}\right)$, $\varphi^{(k)}_{\gamma_{m(j)}} \in W_j\left(\gamma_{m(j)}\right)$ and $\psi^{(p)}_{\delta_{m(i)}} \in M\left(\delta_{m(i)}\right)$, the matrix formulation of (2) is

$$B_{\delta_{m(i)}} u_{i\delta_{m(i)}} - B_{\gamma_{m(j)}} u_{j\gamma_{m(j)}} = 0, \tag{3}$$

where $u_{i\delta_{m(i)}}$ and $u_{j\gamma_{m(j)}}$ are vectors which represent $u_i\big|_{\delta_{m(i)}} \in W_i\left(\delta_{m(i)}\right)$ and $u_j\big|_{\gamma_m(j)} \in W_j\left(\gamma_{m(j)}\right)$, and

$$B_{\delta_{m(i)}} = \left\{(\psi^{(p)}_{\delta_{m(i)}}, \varphi^{(k)}_{\delta_{m(i)}})_{L^2(\delta_{m(i)})}\right\},\ p = 1, \ldots, n_{\delta(i)},\ k = 0, \ldots, n_{\delta(i)} + 1,$$

$$B_{\gamma_{m(j)}} = \left\{(\psi^{(p)}_{\delta_{m(i)}}, \varphi^{(l)}_{\gamma_{m(j)}})_{L^2(\gamma_{m(j)})}\right\},\ p = 1, \ldots, n_{\delta(i)},\ l = 0, \ldots, n_{\gamma_{(j)}} + 1.$$

We rewrite the discrete problem for (1) in V^h as a saddle-point problem using Lagrange multipliers, λ. Its solution is $(u_h^*, \lambda_h^*) \in \widetilde{X}^h(\Omega) \times M(\Gamma)$, where

$\widetilde{X}^h(\Omega)$ denotes a subspace of $X^h(\Omega)$ of functions which are continuous at common vertices of substructures. We partition $u_h^* = \left(u^{(i)}, u^{(c)}, u^{(r)}\right)$ into vectors containing the interior nodal points of Ω_l, vertices of Ω_l, and the remaining nodal points of $\partial\Omega_l \backslash \partial\Omega$, respectively.

Let $K^{(l)}$ be the stiffness matrix of $a_l(\,\cdot\,,\cdot\,)$. It is represented as

$$K^{(l)} = \begin{pmatrix} K_{ii}^{(l)} & K_{ic}^{(l)} & K_{ir}^{(l)} \\ K_{ci}^{(l)} & K_{cc}^{(l)} & K_{cr}^{(l)} \\ K_{ri}^{(l)} & K_{rc}^{(l)} & K_{rr}^{(l)} \end{pmatrix}, \tag{4}$$

where the rows correspond to the interior unknowns, its vertices and its edges.

Using this notation and the assumption of continuity of u_h^* at the vertices of $\partial\Omega_l$, the saddle point problem can be written as

$$\begin{pmatrix} K_{ii} & K_{ic} & K_{ir} & 0 \\ K_{ci} & \widetilde{K}_{cc} & K_{cr} & B_c^T \\ K_{ri} & K_{rc} & K_{rr} & B_r^T \\ 0 & B_c & B_r & 0 \end{pmatrix} \begin{pmatrix} u^{(i)} \\ u^{(c)} \\ u^{(r)} \\ \widetilde{\lambda}^* \end{pmatrix} = \begin{pmatrix} f^{(i)} \\ f^{(c)} \\ f^{(r)} \\ 0 \end{pmatrix}. \tag{5}$$

Here, the matrices K_{ii}, K_{rr} and $\widetilde{K}_{cc}$ are diagonal block-matrices of $K_{ii}^{(l)}$, $K_{rr}^{(l)}$ and $K_{cc}^{(l)}$, where the last one uses the fact that $u^{(c)}$ are the same at the common vertices of substructures. The mortar condition is represented by the global diagonal matrices $B = (B_c, B_r)$.

In the system (5) we eliminate the unknowns $u^{(i)}$ and $u^{(c)}$ to obtain

$$\begin{pmatrix} \widetilde{S} & \widetilde{B}^T \\ \widetilde{B} & \widetilde{S}_{cc} \end{pmatrix} \begin{pmatrix} u^{(r)} \\ \widetilde{\lambda}^* \end{pmatrix} = \begin{pmatrix} \widetilde{f}_r \\ \widetilde{f}_c \end{pmatrix}, \tag{6}$$

where (since $K_{ic} = 0 = K_{ci}$ in the case of triangle elements and a piecewise linear continuous finite element space used in the implementation):

$$\widetilde{S} = K_{rr} - K_{ri}K_{ii}^{-1}K_{ir} - K_{rc}\widetilde{K}_{cc}^{-1}K_{cr}, \quad \widetilde{f}_r = f^{(r)} - K_{ri}K_{ii}^{-1}f^{(i)} - K_{rc}\widetilde{K}_{cc}^{-1}f^{(c)}$$

$$\widetilde{B} = B_r - B_c\widetilde{K}_{cc}^{-1}K_{cr}, \quad \widetilde{S}_{cc} = -B_c\widetilde{K}_{cc}^{-1}B_c^T, \quad \text{and} \quad \widetilde{f}_c = -B_c\widetilde{K}_{cc}^{-1}f_c.$$

We next eliminate the unknown $u^{(r)}$ to get for $\widetilde{\lambda}^* \in M(\Gamma)$

$$F\widetilde{\lambda}^* = d, \tag{7}$$

where

$$F = \widetilde{B}\widetilde{S}^{-1}\widetilde{B}^T - \widetilde{S}_{cc}, \quad \text{and} \quad d = \widetilde{B}\widetilde{S}^{-1}\widetilde{f}_r - \widetilde{f}_c. \tag{8}$$

This is the FETI-DP equation for Lagrange multipliers. Since F is positive definite, the problem has a unique solution. This problem can be solved by conjugate gradient iterations with a preconditioner discussed below.

Let $S^{(l)}$ denote the Schur complement of $K^{(l)}$, see (4), with respect to unknowns at the nodal points of $\partial\Omega_l$. This matrix is represented as

$$S^{(l)} = \begin{pmatrix} S_{rr}^{(l)} & S_{rc}^{(l)} \\ S_{cr}^{(l)} & S_{cc}^{(l)} \end{pmatrix}, \tag{9}$$

where the second row corresponds to unknowns at the vertices of $\partial\Omega_l$ while the first one corresponds to the remaining unknowns of $\partial\Omega_l$. Note that B_r is a matrix obtained from B defined on functions with zero values at the vertices of Ω_l and let

$$S_{rr} = \operatorname{diag}\left\{S_{rr}^{(l)}\right\}_{l=1}^{N},\ S_{cc} = \operatorname{diag}\left\{S_{cc}^{(l)}\right\}_{l=1}^{N},\ S_{cr} = \left(S_{cr}^{(1)}, \ldots, S_{cr}^{(N)}\right). \tag{10}$$

We employ special scaling appropriate for problems with discontinuous coefficients. The preconditioner M for (7) is defined as, see Dokeva, Dryja and Proskurowski [to appear]

$$M^{-1} = \widehat{B}_r \widehat{S}_{rr} \widehat{B}_r^T, \tag{11}$$

where $\widehat{S}_{rr} = \operatorname{diag}\left\{\widehat{S}_{rr}^{(i)}\right\}_{i=1}^{N}$, $\widehat{S}_{rr}^{(i)} = S_{rr}^{(i)}$ for $\rho_i = 1$ and we define $\widehat{B}\big|_{\delta_{m(i)}} = \left(\rho_i^{1/2} I_{\delta_{m(i)}}, -\dfrac{h_{\delta_{m(i)}}}{h_{\gamma_{m(j)}}} \dfrac{\rho_i}{\rho_j} \rho_i^{1/2} B_{\delta_{m(i)}}^{-1} B_{\gamma_{m(j)}}\right)$, for $\delta_{m(i)} \subset \partial\Omega_i$, $i = 1, \ldots, N$; $h_{\delta_{m(i)}}$ and $h_{\gamma_{m(j)}}$ are the step parameters on $\delta_{m(i)}$ and $\gamma_{m(j)}$, respectively.

Following Dokeva, Dryja and Proskurowski [to appear] we have

Theorem 1. *Let the mortar side be chosen where the coefficient ρ_i is larger. Then for $\lambda \in M(\Gamma)$ the following holds*

$$c_0 \left(1 + \log \frac{H}{h}\right)^{-2} \langle M\lambda, \lambda\rangle \leq \langle F\lambda, \lambda\rangle \leq c_1 \left(1 + \log \frac{H}{h}\right)^{2} \langle M\lambda, \lambda\rangle, \tag{12}$$

where c_0 and c_1 are positive constants independent of h_i, H_i, and the jumps of ρ_i; $h = \min_i h_i, H = \max_i H_i$.

This estimate allows us to achieve numerical scalability, an essential ingredient in a successful parallel implementation.

3 Parallel implementation and results

Our parallel implementation problem is divided into three types of tasks: solvers on the subdomains (with different meshes of discretization) which run individually and in parallel, a problem on the interfaces between the subdomains which can be solved in parallel with only a few global communications,

and a "coarse" problem on the vertices between the subdomains which is a global task. A proper implementation of the coarse problem is crucial when the number of processors/subdomains is large.

We discuss some details of the implementation and present experimental results demonstrating that this method is well scalable. The numerical experiments were performed on up to 484 processors provided by the University of Southern California Center for High Performance Computing and Communications (`http://www.usc.edu/hpcc`). All jobs were run on identically configured nodes equipped with dual Intel Pentium 4 Xeon 3.06 GHz processors, 2 GB of RAM and low latency Myrinet networking. Our code was written in C and MPI, using the PETSc toolkit (see Balay et al. [2001]) which interfaces many different solvers.

The test example for our experiments is the weak formulation of

$$-\mathrm{div}(\rho(x)\nabla u) = f(x) \text{ in } \Omega, \tag{13}$$

with the homogenous Dirichlet boundary conditions on $\partial\Omega$, where $\Omega = (0,1)\times(0,1)$ is a union of $N = n^2$ disjoint square subregions Ω_i, $i = 1,\ldots,N$ and $\rho(x) = \rho_i$ is a positive constant in each Ω_i. The coefficients $\rho(x)$ are chosen larger on the mortar sides of the interfaces, see Theorem 1.

The distribution of the coefficients ρ_i and grids h_i in Ω_i, $i = 1,\ldots,4$ with max grid ratio 8 : 1 used in our tests (for larger number of subregions, this pattern of coefficients is repeated) is here with $h = \dfrac{1}{32n}$:

$$\begin{pmatrix} 1e6 & 1e4 \\ 1e2 & 1 \end{pmatrix}, \quad \begin{pmatrix} h/8 & h/4 \\ h/2 & h \end{pmatrix}. \tag{14}$$

Each of the N processors works on a given subdomain and communicates mostly with the processors working on the neighboring subdomains.

For the subdomain solvers, we employ a symmetric block sparse Cholesky solver provided by the SPOOLES library (see Ashcraft and Grimes [1999]). The matrices are decomposed during the first solve and afterwards only a forward and backward substitutions are needed.

At each preconditioned conjugate gradient (PCG) iteration to solve the FETI-DP equation (7) for Lagrange multipliers, there are two main operations:

1. multiplication by the preconditioner $M^{-1} = \widehat{B}_r \widehat{S}_{rr} \widehat{B}_r^T$ which involves solving N Dirichlet problems that are uncoupled, and some operations on the interfaces between the neighboring subdomains.

2. multiplication by $F = \widetilde{B}\widetilde{S}^{-1}\widetilde{B}^T - \widetilde{S}_{cc}$ which involves solving N coupled Neumann problems connected through the vertices.

The latter task involves solving a system with the global stiffness matrix K, see (5), of the form:

$$\begin{pmatrix} K_{ii} & 0 & K_{ir} \\ 0 & \widetilde{K}_{cc} & K_{cr} \\ K_{ri} & K_{rc} & K_{rr} \end{pmatrix} \begin{pmatrix} v_i \\ v_c \\ v_r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ p \end{pmatrix}. \tag{15}$$

Its Schur complement matrix C with respect to the vertices is

$$C = \widetilde{K}_{cc} - (0,\ K_{cr}) \begin{pmatrix} K_{ii} & K_{ir} \\ K_{ri} & K_{rr} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ K_{rc} \end{pmatrix}. \tag{16}$$

C is a sparse, 9-diagonal, $(n-1)^2 \times (n-1)^2$ matrix. Solving a "coarse" problem with C is a global task while the subdomain solvers are local and run in parallel.

Proper implementation of the coarse system solving is important for the scalability especially when the number of processors/subdomains, N is large. Without assembling C, the coarse system could be solved iteratively (for example, with PCG using symmetric Gauss-Seidel preconditioner). Since the cpu cost then depends on N, it is preferable to assemble C.

We implemented two approaches discussed in Bhardwaj et al. [2002]. In the case of relatively small C studied here one can invert C in parallel by duplicating it across a group of processors so that each computes a column of C^{-1} by a direct solver, for which we employed SPOOLES.

When C is larger the above may not be efficient or even possible; in that case one can use distributed storage for C and then a parallel direct solver. In a second implementation we employed the block sparse Cholesky solver from the MUMPS package (see Amestoy et al. [2000] and Amestoy et al. [2001]) interfaced through PETSc. For simplicity, the matrix C was stored on $n-1$ or $(n-1)^2$ processors, with the first choice yielding better performance.

In the tests run on up to (the maximum available to us) $N = 484$ processors the two implementations performed almost identically. In Table 1 and Fig. 1 and 2 we present results from our first implementation when the coarse problem is solved by inverting the matrix C.

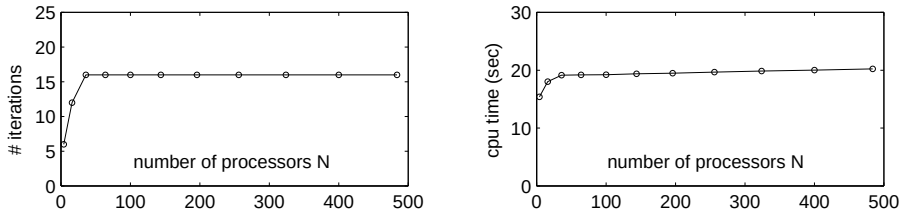

Fig. 1. Iterations and cpu time vs number of processors

Fig. 1 shows that the number of PCG iterations remains constant after $N = 36$ when the number of subdomains/processors is increased. The graph of the execution time (on the right) has a similar pattern. Although the number of degrees of freedom is increasing, the cpu time remains almost constant, see Table 1.

N	# it	d.o.f.	cpu time
4	6	87037	15.41
16	13	350057	18.03
36	16	789061	19.14
64	16	1404049	19.19
100	16	2195021	19.23
144	16	3161977	19.37
196	16	4304917	19.49
256	16	5623841	19.66
324	16	7118749	19.85
400	16	8789641	20.01
484	16	10636517	20.22

Table 1. Cpu time, iterations and d.o.f.

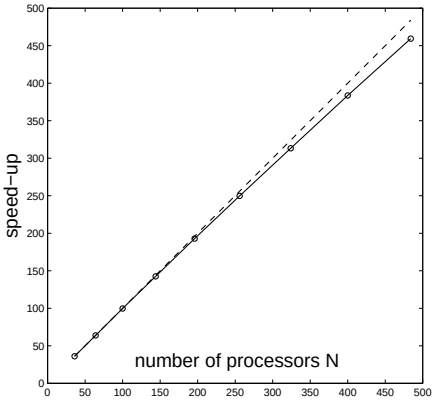

Fig. 2. Speed-up

Fig. 2 shows the speed-up of the algorithm, where the dashed line represents the ideal and the solid line the actual speed-up respectively.

We adopt the definition of the speed-up following Bhardwaj et al. [2000]. Here, it is adjusted to $N_0 = 36$ as a reference point, after which the number of iterations remains constant, see Table 1:

$$Sp = \frac{36 \times T_{36}}{T_{N_p}} \times \frac{N_{dof_{N_s}}}{N_{dof_{36}}},$$

where T_{36} and T_{N_p} denote the CPU time corresponding to 36 and N_p processors, respectively, and $N_{dof_{36}}$ and $N_{dof_{N_s}}$ denote the sizes (in d.o.f.) of the global problems corresponding to 36 and N_s subdomains, respectively.

This definition accounts both for the numerical and parallel scalability.

4 Conclusions

In this paper we study the parallel performance of the FETI–DP mortar preconditioner developed in Dokeva, Dryja and Proskurowski [to appear] for elliptic 2D problems with discontinuous coefficients. Computational evidence presented illustrates a good parallel scalability of the method.

We would like to thank Max Dryja for the collaboration throughout. The first author would like to thank Panayot Vassilevski for the guidance during her summer internship at the Lawrence Livermore National Laboratory. The USC Center for High Performance Computing and Communications (HPCC) is acknowledged for generously providing us with the use of the Linux computer cluster.

References

P. R. Amestoy, I. S. Duff and J-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Methods in Appl. Mech. Eng.*, 184:501–520, 2000.

P. R. Amestoy, I. S. Duff, J. Koster and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. of Matr. Anal. and Appl.*, 23(1):15–41, 2001.

C. Ashcraft and R. G. Grimes. SPOOLES: An object-oriented sparse matrix library. *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*, 1999.

S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith and H. Zhang. PETSc (Portable, Extensible Toolkit for Scientific Computation) web page. `http://www.mcs.anl.gov/petsc`

M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson and D. Rixen. Application of the FETI method to the ASCI problems: scalability results on one thousand processors and discussion of highly heterogenous problems. *Internat. J. Numer. Meths. Engrg.*, 47:513–536, 2000.

M. Bhardwaj, K. Pierson, G. Reese, T. Walsh, D. Day, K. Alvin, J. Peery, C. Farhat, M. Lesoinne. Salinas: A Scalable Software for High-Performance Structural and Solid Mechanics Simulations. *ACM/IEEE Proceedings of SC02: High Performance Networking and Computing Conference (Gordon Bell Award)*, Baltimore, 2002.

N. Dokeva, M. Dryja and W. Proskurowski. A FETI-DP preconditioner with special scaling for mortar discretization of elliptic problems with discontinuous coefficients. *SIAM J. Numer. Anal.*, to appear.

M. Dryja and O. B. Widlund. A FETI-DP method for a mortar discretization of elliptic problems. In L. Pavarino and A. Toselli (eds.), *Recent Developments in Domain Decomposition Methods*, Springer, 2002, pp. 41–52.

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

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. *Internat. J. Numer. Meths. Engrg.*, 50:1523–1544, 2001.

A. Klawonn, O. B. Widlund and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM J. Numer. Anal.*, 40:159–179, 2002.

J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88:543–558, 2001.