# Implementation of Non-overlapping Domain Decomposition Methods on Parallel Computer ADENA

ATSUSHI SUZUKI[1]

This paper discusses parallel efficiency of non-overlapping Domain Decomposition methods(DDM) for elliptic problem with the results of the implementation on parallel computer ADENA, and describes DDM solver for Stokes problem.

## 1 NON-OVERLAPPING METHOD FOR ELLIPTIC PROBLEM

In the Non-overlapping Domain Decomposition method, appropriate boundary condition on inner-boundary need to be found to get a solution satisfying original problem. There are two methods differed in boundary data. One method requires continuity of Dirichlet data on inner-boundary[GDP84]. Neumann data on the interface is obtained by preconditioned Conjugate Gradient solver. We call '*Neumann type*' method in this report. The other method requires continuity of Neumann data on inner-boundary [BW86]. This method is well-known as Schur complement method, we call '*Dirichlet type*' method.

In the following section we discuss details of two algorithms for the elliptic problem

[1] Division of Applied Systems Science, Faculty of Engineering, Kyoto University, Kyoto, 606-01, Japan
e-mail : suzuki@kuamp.kyoto-u.ac.jp

*Domain Decomposition Methods in Sciences and Engineering,* edited by R. Glowinski *et al.*

in two dimensional space:

$$(E)\begin{cases} -\triangle u = f & \text{in} \quad \Omega \\ u = g & \text{on} \quad \Gamma = \partial\Omega \end{cases} . \tag{1}$$

For the simplicity of representation, we discuss the two sub-domains problem. Domain $\Omega$ is divided into two sub-domains $\{\Omega_i\}_{i=1}^2$ and $\gamma_{12}$ is the interface between the two sub-domains : $\gamma_{12} = \partial\Omega_1 \cap \partial\Omega_2$.

### *1.1 Interface Problem with Lagrange Multiplier*

Non-overlapping DDM is formulated as minimization problem with constraint. Let $\{J_i(v_i)\}_{i=1}^2$ be cost function in sub-domain and $b(\{v_1, v_2\}, q)$ be constraint on the interface, the minimization problem is following:

$$\text{Find} \quad \{u_1, u_2\} \in V \quad \text{such that} \quad J_1(u_1) + J_2(u_2) \leq J_1(v_1) + J_2(v_2) \ \forall \{v_1, v_2\} \in V, \tag{2}$$

$$V := \{\{v_1, v_2\} \in X_1 \times X_2\} \mid b(\{v_1, v_2\}, q) = 0 \ \forall q \in M\}.$$

By use of Lagrangean : $\mathcal{L}(\{v_1, v_2\}, q) := J_1(v_1) + J_2(v_2) + b(\{v_1, v_2\}, q)$, the solution of this minimization problem is equal to the solution of the saddle point problem :

$$\begin{aligned} &\text{Find} \quad (\{u_1, u_2\}, p) \in (X_1 \times X_2) \times M \text{ such that} \\ &\mathcal{L}(\{u_1, u_2\}, q) \leq \mathcal{L}(\{u_1, u_2\}, p) \leq \mathcal{L}(\{v_1, v_2\}, p) \quad \forall (\{v_1, v_2\}, q) \in (X_1 \times X_2) \times M. \end{aligned}$$

### *1.2 Neumann type*

In Neumann type, let $X_i := \{v_i \in H^1(\Omega_i) \mid v_i = 0 \text{ on } \partial\Omega_i \setminus \gamma_{12}\}$ and $M := H^{-\frac{1}{2}}(\gamma_{12})$, and cost function and constraint are

$$J_i(v_i) := \frac{1}{2}\int_{\Omega_i} \nabla u_i \nabla v_i dx - \int_{\Omega_i} f_i v_i dx \quad \text{and} \quad b(\{v_1, v_2\}, q) := -\int_{\gamma_{12}} (v_1 - v_2) q d\gamma.$$

The constraint means Dirichlet data on the interface $\gamma_{12}$ is continuous. The dual problem of equation(2) is a variational problem called 'interface problem' which determine Neumann data on the interface.

$$\text{Find} \quad p \in H^{-\frac{1}{2}}(\gamma_{12}) \quad \text{such that} \quad \alpha(p, q) = F(q) \qquad \forall q \in H^{\frac{1}{2}}(\gamma_{12}) \tag{3}$$

$$\alpha(p, q) = \int_{\gamma_{12}} (\mathcal{A}p) q d\gamma \qquad \mathcal{A}p = u_1(p) - u_2(p) \qquad F(q) = \int_{\gamma_{12}} (w_2 - w_1) q d\gamma \tag{4}$$

$$\begin{cases} -\triangle u_i = 0 & \text{in} \quad \Omega_i \\ u_i = 0 & \text{on} \quad \partial\Omega_i \setminus \gamma_{12} \\ \frac{\partial u_i}{\partial n_i} = (-1)^{i-1} p & \text{on} \quad \gamma_{12} \end{cases} \qquad \begin{cases} -\triangle w_i = f_i & \text{in} \quad \Omega_i \\ w_i = g_i & \text{on} \quad \partial\Omega_i \setminus \gamma_{12} \\ \frac{\partial w_i}{\partial n_i} = 0 & \text{on} \quad \gamma_{12} \end{cases} \tag{5}$$

### *1.3 Dirichlet type*

In Dirichlet type, we consider dualization of the Dirichlet problem and divergence space to deal with Dirichlet data in weak sense. let $X_i := \{v_i \in H(\text{div} \ ; \Omega_i) \mid \nabla \cdot v_i + f_i = 0\}$

and $M := H^{\frac{1}{2}}(\gamma_{12})$, cost function and constraint are

$$J_i(v_i) := \frac{1}{2}\int_{\Omega_i}|v_i|^2 dx - \int_{\partial\Omega_i\setminus\gamma_{12}} g_i v_i \cdot n_i d\gamma \quad \text{and} \quad b(\{v_1,v_2\},q) := -\int_{\gamma_{12}}(v_1\cdot n_1 + v_2\cdot n_2)q d\gamma.$$

The constraint means Neumann data is continuous on $\gamma_{12}$. The dual problem of equation(2) is a variational problem called 'interface problem' which determine Dirichlet data on the interface. Usual formulation is used to describe Drichilet problem in sub-domain.

$$\text{Find } p \in H^{\frac{1}{2}}(\gamma_{12}) \text{ such that } \alpha(p,q) = F(q) \quad \forall q \in H^{-\frac{1}{2}}(\gamma_{12}) \tag{6}$$

$$\alpha(p,q) = \int_{\gamma_{12}} (\mathcal{A}p) q d\gamma \quad \mathcal{A}p = \frac{\partial u_1(p)}{\partial n_1} + \frac{\partial u_2(p)}{\partial n_2} \quad F(q) = -\int_{\gamma_{12}} \left(\frac{\partial w_1}{\partial n_1} + \frac{\partial w_2}{\partial n_2}\right) q d\gamma \tag{7}$$

$$\begin{cases} -\triangle u_i = 0 & \text{in} \quad \Omega_i \\ u_i = 0 & \text{on} \quad \partial\Omega_i \setminus \gamma_{12} \\ u_i = p & \text{on} \quad \gamma_{12} \end{cases} \qquad \begin{cases} -\triangle w_i = f_i & \text{in} \quad \Omega_i \\ w_i = g_i & \text{on} \quad \partial\Omega_i \setminus \gamma_{12} \\ w_i = 0 & \text{on} \quad \gamma_{12} \end{cases} \tag{8}$$

Calculation of the gap of Neumann data on $\gamma_{12}$ is done by Gauss-Green's formula exactly,

$$\int_{\gamma_{12}} \left(\frac{\partial u_1}{\partial n_1} + \frac{\partial u_2}{\partial n_2}\right) q d\gamma = \sum_{i=1}^{2} \int_{\Omega_i} \nabla u_i \cdot \nabla \tilde{q}_i dx,$$

where $\tilde{q}$ stands for extension of function $q$ on $\gamma_{12}$, in descrete case FEM basis on $\gamma_{12}$. This $L^2(\gamma_{12})$ inner product is replaced by $H^1(\gamma_{12})$ inner product as the preconditioner for Conjugate Gradient solver of equation(6).

### *1.4 Homogeneous Neumann Boundary Problem*

The Poisson equation with homogeneous Neumann boundary condition must satisfy the compatibility condition, and its solution is unique except ambiguity of a constant.

$$\text{Find } u \in H^1(\Omega)\setminus \mathbb{R} \quad \begin{cases} -\triangle u = f & \text{in } \Omega \\ \frac{\partial u}{\partial n} = 0 & \text{on } \Gamma \end{cases} \qquad \int_\Omega f dx = 0 \tag{9}$$

Neumann type DDM can solve the problem with balancing procedure to satisfy compatibility condition in each sub-domain, and to adjust the constant over sub-domains. This procedure is described as following.

Initial data of search vector $p^0$ of CG solver must satisfy the compatibility condition:

$$\begin{cases} -\triangle u_i^0 = f_i & \text{in} \quad \Omega_i \\ \frac{\partial u_i^0}{\partial n_i} = 0 & \text{on} \quad \partial\Omega_i \setminus \gamma_{12} \\ \frac{\partial u_i^0}{\partial n_i} = (-1)^{i-1} p^0 & \text{on} \quad \gamma_{12} \end{cases} \qquad \int_{\Omega_i} f_i dx + (-1)^{(i-1)} \int_{\gamma_{12}} p^0 d\gamma = 0. \tag{10}$$

On each step of CG iteration, sub-problems with Neumann boundary data $p^n$ which must satisfy the compatibility condition are solved.

$$\begin{cases} -\triangle u_i^n = 0 & \text{in} \quad \Omega_i \\ \frac{\partial u_i^n}{\partial n_i} = 0 & \text{on} \quad \Gamma_i \\ \frac{\partial u_i^n}{\partial n_i} = (-1)^{i-1} p^n & \text{on} \quad \gamma_{12} \end{cases} \qquad \int_{\gamma_{12}} p^n = 0. \tag{11}$$

Because each solution $u_i^n$ has ambiguity of a constant, the constant $\{c_i\}_{i=1}^2$ in each sub-domain should be determined for next step of CG iteration,

$$\int_{\gamma_{12}} (u_1^n - c_1^n) - (u_2^n - c_2^n) d\gamma = 0. \tag{12}$$

If the domain is decomposed into many sub-domains, the matrix with one dimensional kernel have to be solved to determine the constants of sub-domains. Because of the singularity of this matrix, the direct solver is efficient than iterative solvers. This solver must be processed in a single processor to avoid the redundant cost for communication.

## 2 MATRIX REPRESENTATION OF INTERFACE PROBLEM

In this section, we give a matrix representation of 'interface problem'. From finite element base functions $\varphi_i$, matrices $A, B$ are obtained : $(A)_{ij} = a(\varphi_j, \varphi_i)$, $(B)_{ij} = \int_{\gamma_{12}} \varphi_j \varphi_i d\gamma$. Suppose that $u_I^{(i)}$ denotes the values on inner nodes and $u_B^{(i)}$ on the interface $\gamma_{12}$. Also matrix $A^{(i)}$ is decomposed into four parts.

### *2.1 Neumann type*

$$\begin{cases} -\triangle u_i = 0 & \text{in} \quad \Omega_i \\ u_i = 0 & \text{on} \quad \partial\Omega_i \setminus \gamma_{12} \\ \frac{\partial u_i}{\partial n_i} = (-1)^{i-1} p & \text{on} \quad \gamma_{12} \end{cases} \qquad \begin{pmatrix} A_{II}^{(i)} & A_{IB}^{(i)} \\ A_{IB}^{(i)\,T} & A_{BB}^{(i)} \end{pmatrix} \begin{pmatrix} u_I^{(i)} \\ u_B^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ (-1)^{i-1} Bp \end{pmatrix}.$$

By eliminating data in sub-domain $u_I^{(i)}$ matrix representation of 'interface problem' operator is obtained:

$$\mathcal{A}_h = \left( (A_{BB}^{(1)} - A_{IB}^{(1)\,T} A_{II}^{(1)\,-1} A_{IB}^{(1)})^{-1} + (A_{BB}^{(2)} - A_{IB}^{(2)\,T} A_{II}^{(2)\,-1} A_{IB}^{(2)})^{-1} \right) B. \tag{13}$$

We use main part of the operator $\mathcal{A}_h$, $(A_{BB}^{(1)\,-1} + A_{BB}^{(2)\,-1})B$ as a preconditioner for CG solver. We use a direct solver (modified Cholesky decomposition solver) for elliptic problem in sub-domain. The reason is that direct solver is faster than iterative solver in small size problem.

### *2.2 Dirichlet type*

$$\begin{cases} -\triangle u_i = 0 & \text{in} \quad \Omega_i \\ u_i = 0 & \text{on} \quad \partial\Omega_i \setminus \gamma_{12} \\ u_i = p & \text{on} \quad \gamma_{12} \end{cases} \qquad \begin{pmatrix} A_{II}^{(i)} & A_{IB}^{(i)} \\ 0 & E_{BB} \end{pmatrix} \begin{pmatrix} u_I^{(i)} \\ u_B^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ p \end{pmatrix}$$

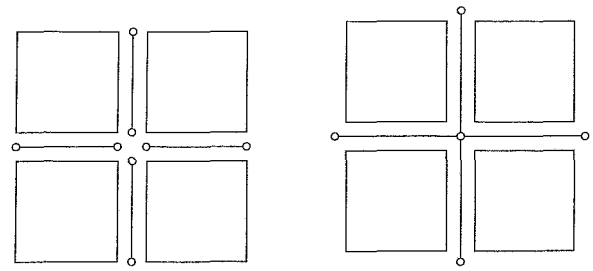
Neumann data on the the interface are calculated by data $u_I^{(i)}$ exactly. A matrix representation of 'interface problem' operator is obtained :

$$\mathcal{A}_h = B^{-1} \left( (A_{BB}^{(1)} - A_{IB}^{(1)\,T} A_{II}^{(1)\,-1} A_{IB}^{(1)}) + (A_{BB}^{(2)} - A_{IB}^{(2)\,T} A_{II}^{(2)\,-1} A_{IB}^{(2)}) \right). \tag{14}$$

The matrix representation of discrete 'interface problem' is equivalent to Schur complement matrix except for matrix $B^{-1}$. As in Neumann type, we use $B^{-1}(A_{BB}^{(1)} + A_{BB}^{(2)})$ as a preconditioner for CG solver.

These preconditioner are easily extended to multi-sub-domains problems. In Neumann type method, pieces of the interface are independent each other(shown in left of figure 1), the residual vector of 'interface problem' are calculated in parallel over all pieces of the interface. While, in Dirichlet type method, because of cross points, the interface has complicated structure(shown in right of figure 1). Therefore the residual vector given in $H^1(\gamma)$ inner product are calculated using the iterative method which requires cooperation of all processors for pieces of the interface.

**Figure 1** The structure of inner-boundary




## 3 IMPLEMENTATION ON PARALLEL COMPUTER

### *3.1 ADENA massively parallel computer*

The ADENA is distributed memory system, which has 256 CPUs with 2M bytes local memory. By using '*Alternating Data Edition*' network, all processors can communicate with two times operations of data-transfer. Moreover it is very easy to gather information of all processors to one processor, and also, to broadcast data in one processor to other all processors. This ability is powerful for inner product calculation, over all edges of sub-domains, which is required in the procedure of updating boundary data on interface in the DDM.

### *3.2 Results of elliptic problem on ADENA*

The test problem is $u - \triangle u = f$ in $[0,1] \times [0,1]$, and the exact solution is $u = \sin(0.5\pi(x+0.1))\sin(0.5\pi(y+0.1))$. Square region contains $480 \times 480$ nodes. Table 1 shows the results in two type algorithms. Time for data transfer is less than 0.7% of total time of solver. Dirichlet type solver is faster than Neumann type solver. However in Dirichlet type, as described in section2.2, due to the cross point of boundary

interface, time for updating boundary data is about 10% at 256 processors, against 1.7% in Neumann type.

**Table 1** Results of elliptic problem on ADENA ($480 \times 480$ nodes, $\varepsilon = 10^{-10}$)

| | Neumann type | | | | Dirichlet type | | | |
|---|---|---|---|---|---|---|---|---|
| # processors | 100 | 144 | 225 | 256 | 100 | 144 | 225 | 256 |
| # iteration | 303 | 353 | 442 | 475 | 165 | 178 | 197 | 203 |
| total time | 395.40 | 277.72 | 189.57 | 171.71 | 199.14 | 129.28 | 78.69 | 68.50 |
| parallel efficiency % | 100.0 | 98.8 | 92.6 | 89.9 | 100.0 | 106.9 | 112.4 | 113.5 |
| boundary data update | 2.83 | 2.60 | 2.48 | 2.92 | 7.11 | 6.60 | 7.02 | 6.96 |
| data transfer | 0.77 | 0.85 | 1.08 | 1.08 | 0.35 | 0.36 | 0.41 | 0.39 |

time (sec.)

Table 2 shows the results of homogeneous Neumann boundary problem with fixed number of CPU and various size of problem. As viewed in section 1.4, balancing procedure is the bottle neck of the algorthim. From this result, we conclude that the algorithm is efficient if only the condition is satisfied : the number of sub-domain is less than the number of nodes in sub-domain.

**Table 2** Homogeneous Neumann boundary problem on ADENA (256CPUs, $\varepsilon = 10^{-10}$)

| problem size | $64 \times 64$ | $96 \times 96$ | $128 \times 128$ | $160 \times 160$ | $192 \times 192$ | $224 \times 224$ | $256 \times 256$ |
|---|---|---|---|---|---|---|---|
| total time | 5.269 | 6.553 | 7.842 | 9.174 | 10.868 | 12.879 | 15.628 |
| direct solver | 0.174 | 0.440 | 0.903 | 1.615 | 2.672 | 4.121 | 6.135 |
| balancing | 4.850 | 5.826 | 6.604 | 7.184 | 7.777 | 8.293 | 8.963 |

time (sec.)

### *3.3 Results on vector parallel computer VPP500*

The Fujitsu VPP500 is vector parallel system which has vector processors with cross-bar network. Vector parallel machine can simulate massively parallel machine by replacing parallel loop over sub-domains with vector loop. However, to use vector parallel machine effectively, vector loop should be long enough. The results of Dirichlet type of elliptic problem is shown in table 3. From this results, we need more fast convergence of 'interface problem' and low-cost algorithm for calculation of norms on the interface .

**Table 3** Results of elliptic problem with Dirichlet type DDM on VPP500 ($480 \times 480$ nodes, $\varepsilon = 10^{-10}$)

| # sub-domains | | $16 \times 16$ | $20 \times 20$ | $30 \times 30$ | $40 \times 40$ | $60 \times 60$ | $80 \times 80$ |
|---|---|---|---|---|---|---|---|
| # iterations | | 223 | 246 | 296 | 337 | 368 | 412 |
| 1 CPU | total time | 6.433 | 6.123 | 6.233 | 7.628 | 11.963 | 20.140 |
| | direct solver | 5.355 | 4.660 | 3.665 | 3.318 | 3.428 | 3.743 |
| | interface | 0.798 | 1.131 | 2.102 | 3.699 | 7.682 | 15.224 |
| 2 CPUs | total time | 5.289 | 4.580 | 4.516 | 5.435 | 7.773 | 12.758 |
| | direct solver | 3.934 | 2.906 | 2.088 | 1.806 | 1.585 | 1.936 |
| | interface | 1.139 | 1.424 | 2.099 | 3.214 | 5.648 | 10.104 |
| 4 CPUs | total time | 4.477 | 3.986 | — | 4.062 | 5.504 | 8.424 |
| | direct solver | 3.396 | 2.372 | — | 1.059 | 0.882 | 0.926 |
| | interface | 1.211 | 1.430 | — | 2.717 | 4.266 | 7.049 |

time (sec.)

## 4 DOMAIN DECOMPOSITION SOLVER FOR STOKES PROBLEM

We consider a generalized Stokes problem in two dimension :

$$(S)\begin{cases} \alpha u - \nu\triangle u + \nabla p = f & \text{in} \quad \Omega \qquad u = g \text{ on } \Gamma = \partial\Omega \\ \nabla \cdot u = 0 & \text{in} \quad \Omega \qquad (\text{with } \int_\Gamma g \cdot n d\gamma = 0 \end{cases}, \tag{15}$$

where $\alpha$ and $\nu$ are positive constants. Generalized Stokes problem is solved by saddle point algorithm which contains two kinds of elliptic problem. One is problem for velocity with Dirichlet boundary condition, the other is problem for pressure with Neumann boundary condition for preconditioner : $(\nu I_h^{-1} + \alpha(-\triangle)_h^{-1})$ to improve convergence of saddle point solver[CC88].

We consider two alternatives of DDM approaches. One approach is to consider generalized Stokes problems in each sub-domain. This approaches needs CG solver for the saddle point problem of each sub-problem, in each iterative procedure of DDM. Therefore this method is not better because of high cost of iterative solver for sub-problems.

The other is to apply DDM only to the elliptic solver in generalized Stokes saddle point solver. In this method we can use the direct solver as elliptic solver, which is effective for small size sub-problems.

We use Dirichlet type DDM for elliptic problem of velocity, and Neumann type DDM for pressure.

The test problem is the kernel of implicit scheme for time dependent cavity flow problem. $\alpha(= 68.28)(\nu(= 0.001715))$ is reciprocally proportional to the time step(Reynolds number, respectively). We use $P1$ iso P2/P1 mesh for velocity/pressure. Table4 shows computation time by second in various numbers of pressure nodes. In small size problems, time for preconditioner of pressure consumes almost time of the solver.

**Table 4** Result of generalized Stokes problem on ADENA (256 CPUs, $\varepsilon = 10^{-10}$)

| # pressure node | 64×64 | 96×96 | 128×128 | 160×160 | 192×192 | 224×224 | 256×256 |
|---|---|---|---|---|---|---|---|
| solver total | 174.80 | 215.51 | 278.44 | 361.73 | 468.03 | 598.71 | 797.24 |
| velocity | 16.72 | 23.45 | 43.78 | 78.58 | 127.48 | 201.67 | 307.33 |
| Neumann prob. | 145.99 | 174.82 | 209.27 | 246.93 | 289.72 | 328.94 | 398.06 |
| mass matrix | 4.71 | 7.18 | 11.67 | 17.84 | 26.28 | 35.69 | 49.00 |

time (sec.)

## 5 CONCLUSION

We implemented the DDM for elliptic problem on massively parallel computer ADENA. Numerical results show that Dirichlet type is more fast and stable than Neumann type. Another implementation on vector parallel computer shows that we need much faster algorithm which can deal huge number of sub-domains. Application of elliptic DDM to the generalized Stokes problem is suitable for parallel processing. To get more high performance, we need to develop better preconditioner for Stokes solver suitable for DDM.

*Acknowledgment.* The author wihes to thank Professor T. Nogi for his help in this study.

# References

[BF91] Brezzi F. and Fortin M. (1991) *Mixed and Hybrid Finite Element Methods.* Springer-Verlag.

[BW86] Bjørstad P. E. and Widlund O. B. (December 1986) Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.* 23(6): 1097–1120.

[CC88] Cahouet J. and Chabard J. P. (1988) Some fast 3d finite element solvers for the generalized stokes problem. *Int. J. Numer. Methods in Fluids.* 8: 869–895.

[GDP84] Glowinski R., Dinh Q. V., and Périaux J. (1984) *Domain Decomposition for Elliptic Problems*, volume 5 of *Finite Elements in Fluids*, chapter 3, pages 45–106. John Wiley & Sons Limited.

[Su94] Suzuki A. (1994(to appear)) Implementation of domain decomposition methods on parallel computer ADENART. In Ecer A., Périaux J., and Satofuka N. (eds) *Proceedings of Parallel CFD '94.* Elsevier Science.