数理解析研究所講究録
1061巻 1998年 164-176

A Domain-Decomposition / Upwind Finite Element Approximation for the Navier-Stokes Equations

SHOICHI FUJIMA (藤間 昌一)
Department of Mechanical Science and Engineering, Kyushu University,
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan

1 Introduction

In recent years, parallel computers have changed techniques to solve problems in various kinds of fields. In parallel computers of distributed memory type, data can be shared by communication procedures called message-passing, whose speed is slower than that of computations in a processor. From a practical point of view, it is important to reduce the amount of message-passings. Domain-decomposition is an efficient technique to parallelize partial differential equation solvers on such parallel computers.

In one type of the domain decomposition method, a Lagrange multiplier for the weak continuity between subdomains is used. This type has the potential to decrease message-passings since (i) independency of computations in each subdomain is high and (ii) two subdomains which share only one nodal point do not need to execute message-passings each other. For the Navier-Stokes equations, domain decomposition methods using a Lagrange multipliers have been proposed. Achdou et al.[1, 2] has applied the mortar element method to the Navier-Stokes equations of stream function-vorticity formulation. Glowinski et al.[3] has shown the fictitious domain method in which they use the constant element for the Lagrange multiplier. Suzuki[4] has shown a method using the iso-P2 P1 element. But the choice of the base function for the Lagrange multiplier has not been well compared in one domain decomposition algorithm.

In this paper we propose a domain-decomposition/finite-element method for the Navier-Stokes equations of the velocity-pressure formulation. In the method, subdomain-wise finite element spaces by the iso-P2 P1/P1 elements[5] are used for the velocity and the pressure, respectively. For the upwinding, the upwind finite element approximation based on the choice of up- and downwind points[6] is used. For the discretization of the Lagrange multiplier, three cases are compared numerically. As a result, iso-P2 P1/P1/P1 element is the best choice.

2 Domain decomposition/finite-element method for the Navier-Stokes equations

Let Ω be a bounded domain in R^2. Let $\Gamma_D(\neq \emptyset)$ and Γ_N be two parts of the boundary $\partial\Omega$. We consider the incompressible Navier-Stokes equations,

$$\begin{aligned} \partial u/\partial t + (u \cdot \mathrm{grad})u + \mathrm{grad}p &= (1/Re)\Delta u + f \qquad \text{in } \Omega, && (1)\\ \mathrm{div} u &= 0 \qquad \text{in } \Omega, && (2)\\ u &= g_D \qquad \text{on } \Gamma_D, && (3)\\ \sigma n &= g_N \qquad \text{on } \Gamma_N, && (4) \end{aligned}$$

where u is the velocity, p is the pressure, Re is the Reynolds number, f is the external force, g_D and g_N are given boundary data, σ is the stress tensor and n is the unit outward normal to Γ_N.

We decompose a domain into K non-overlapping subdomains,

$$\overline{\Omega} = \overline{\Omega_1} \cup \cdots \cup \overline{\Omega_K}, \quad \Omega_k \cap \Omega_l = \emptyset \quad (k \neq l). \tag{5}$$

We denote by n_k the unit outward normal on $\partial\Omega_k$. If $\overline{\Omega_k} \cap \overline{\Omega_l}$ $(k \neq l)$ includes an edge of an element, we say an interface of the subdomains appears. We denote all interfaces by $\Gamma_m, m = 1, \ldots, M$. We assume they are straight segments. Let us define integers $\kappa_-(m)$ and $\kappa_+(m)$ by

$$\Gamma_m = \overline{\Omega_{\kappa-(m)}} \cap \overline{\Omega_{\kappa+(m)}} \qquad (\kappa_-(m) < \kappa_+(m)). \tag{6}$$

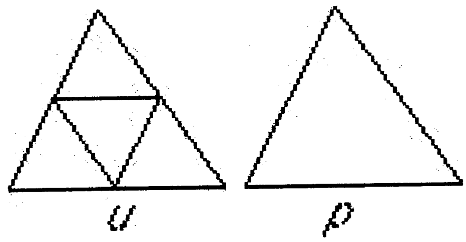


Figure 1: Iso-P2 P1/P1 elements

Let $\mathcal{T}_{k,h}$ be a triangular subdivision of Ω_k. We further divide each triangle into four congruent triangles, and generate a finer triangular subdivision $\mathcal{T}_{k,h/2}$. We assume that the positions of the nodal points in $\Omega_{\kappa+(m)}$ and those in $\Omega_{\kappa-(m)}$ coincides on Γ_m. We use iso-P2 P1/P1 finite elements[5] for the velocity and the pressure subdomainwise by

$$V_{k,h} = \{v \in (C(\overline{\Omega_k}))^2; \quad v_{|e} \in (P^1(e))^2, e \in \mathcal{T}_{k,h/2}, v = 0 \text{ on } \partial\Omega_k \cap \Gamma_D\}, \tag{7}$$

$$Q_{k,h} = \{q \in C(\overline{\Omega_k}); \quad q_{|e} \in P^1(e), e \in \mathcal{T}_{k,h}\}, \tag{8}$$

respectively, we construct the finite element spaces by

$$V_h = \prod_{k=1}^{K} V_{k,h}, \qquad Q_h = \prod_{k=1}^{K} Q_{k,h}. \tag{9}$$

Concerning weak continuity of the velocity between subdomains, we employ the Lagrange multiplier on the interfaces. For the discretization of the spaces of the Lagrange multiplier defined on Γ_m $(1 \le m \le M)$, we compare three cases (see Figure 2):

Case 1. The conventional iso-P2 P1 element, that is defined by

$$W_{m,h} = (X_{\kappa+(m),h}|_{\Gamma_m})^2, \tag{10}$$

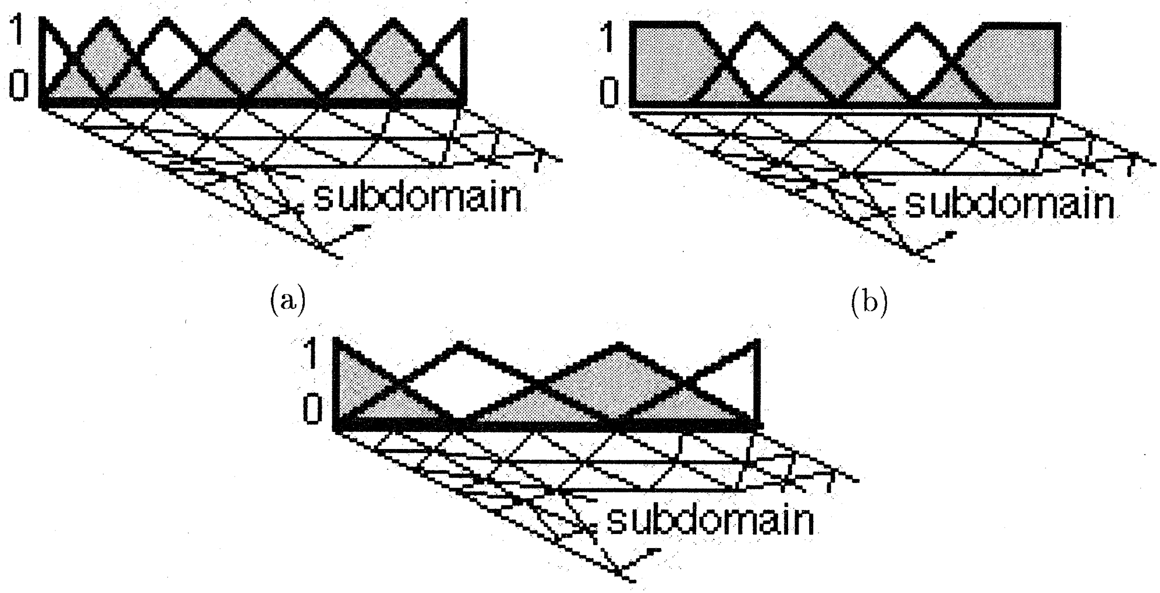
where

$$X_{k,h} = \{v \in C(\overline{\Omega_k}); \quad v_{|e} \in P^1(e), e \in \mathcal{T}_{k,h/2}\}. \tag{11}$$

Case 2. A modified iso-P2 P1 element having no freedoms at both edges of interfaces[7].

Case 3. The conventional P1 element, that is defined by

$$W_{m,h} = (Y_{\kappa+(m),h}|_{\Gamma_m})^2, \tag{12}$$



(a) (b) (c)

Figure 2: Shapes of (a)iso-P2, (b)modified iso-P2 and (c)P1 base functions for the Lagrange multiplier and a subdivision $\mathcal{T}_{k,h/2}$

where

$$Y_{k,h} = \{v \in C(\overline{\Omega_k}); \quad v_{|e} \in P^1(e), e \in \mathcal{T}_{k,h}\}. \tag{13}$$

The finite element space W_h is defined by

$$W_h = \prod_{m=1}^{M} W_{m,h}. \tag{14}$$

We consider the time-discretized finite element equations derived from (1)-(4),

Problem 1. Find $(u_h^{n+1}, p_h^n, \lambda_h^n) \in V_h \times Q_h \times W_h$ such that

$$\begin{aligned} \forall v_h \in V_h, \quad (\frac{u_h^{n+1} - u_h^n}{\tau}, v_h)_h + b(v_h, p_h^n) + j(v_h, \lambda_h^n) &= \langle \hat{f}, v_h \rangle \\ &\quad -a_1^h(u_h^n, u_h^n, v_h) \\ &\quad -a_0(u_h^n, v_h), \end{aligned} \tag{15}$$

$$\forall q_h \in Q_h, \quad b(u_h^{n+1}, q_h) = 0, \tag{16}$$

$$\forall \mu_h \in W_h, \quad j(u_h^{n+1}, \mu_h) = 0, \tag{17}$$

where

$$
\begin{aligned}
(u, v) &= \sum_{k=1}^{K} \int_{\Omega_k} u_k \cdot v_k dx, && (18)\\
a_1(w, u, v) &= \sum_{k=1}^{K} \int_{\Omega_k} (w_k \cdot \mathrm{grad} u_k) v_k dx, && (19)\\
a_0(u, v) &= \frac{2}{Re} \sum_{k=1}^{K} \int_{\Omega_k} D(u_k) \otimes D(v_k) dx, && (20)\\
b(v, q) &= -\sum_{k=1}^{K} \int_{\Omega_k} q_k \mathrm{div} v_k dx, && (21)\\
j(v, \mu) &= -\sum_{m=1}^{M} \int_{\Gamma_m} (v_{\kappa_+(m)} - v_{\kappa_-(m)}) \mu_m ds, && (22)\\
\langle \hat{f}, v \rangle &= \sum_{k=1}^{K} \Big(\int_{\Omega_k} f \cdot v_k dx + \int_{\partial\Omega_k \cap \Gamma_N} g_N \cdot v_k ds \Big), && (23)
\end{aligned}
$$

$(,)_h$ denotes the mass-lumping corresponding to $(,)$, a_1^h is the upwind finite element approximation based on the choice of up- and downwind points[6] to a_1, and D is the strain rate tensor.

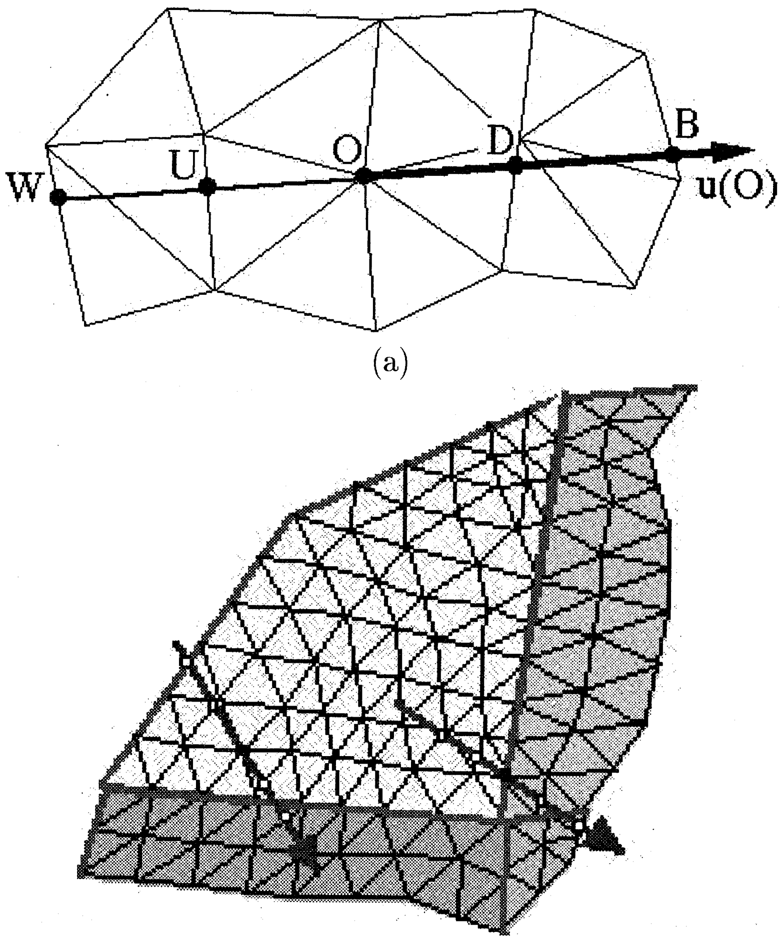
We rewrite Problem 1 by a matrix form as,

$$
\begin{pmatrix} \bar{M} & B^T & J^T \\ B & O & O \\ J & O & O \end{pmatrix} \begin{pmatrix} U^{n+1} \\ P^n \\ \Lambda^n \end{pmatrix} = \begin{pmatrix} F^n \\ 0 \\ 0 \end{pmatrix}, \tag{24}
$$

where $\bar{M}$ is the lumped-mass matrix, B is the divergence matrix, J is the jump matrix, F^n is a known vector, and U^{n+1},P^n and Λ^n are unknown vectors. Eliminating U^{n+1} from (24), we get a domain-decomposition version of the consistent discretized pressure equation[8]. Further eliminating P^n, we obtain a system of linear equations with respect to Λ^n. Applying CG method to this equation, a domain decomposition algorithm is obtained[9].

Remark 1. The quantity $\lambda_{m,h}$ corresponds to $\sigma \cdot n_{\kappa_+(m)}|_{\gamma_m}$.

Remark 2. In implementation, an idea of two data types[10] is applied to the Lagrange multipliers and the jump matrix. The idea simplifies the implementation and reduce the amount of message-passings. For the velocity and the pressure, we do not need to execute message-passings.



(a)

(b)

Figure 3: (a)Two upwind points(W,U) and two downwind points(D,B) in the upwind finite element approximation based on the choice of up- and downwind points and (b)a domain-decomposition situation

Remark 3. In order to evaluate $a_1^h(u_h^n, u_h^n, v_h)$, we need to find two upwind points and two downwind points for each nodal points. In the domain-decomposition situation, some of these up- and downwind points for nodal points near interfaces may be included in the neighbor subdomains. In order to treat it, each processor corresponding to a subdomain has geometry information of all elements which share at least a point with neighbor subdomains (see Figure 3(right)). The processors exchange each other the values of u_h^n before the evaluation. Hence the evaluation itself is parallelized without any message-passings.

3 Numerical experiments

3.1 Test problem

Let $\Omega = (0,1) \times (0,1)$ and $\Gamma_D = \partial\Omega$ ($\Gamma_N = \emptyset$). The exact stational solution is

$$u(x,y) = (x^2y + y^3, -x^3 - xy^2)^T, \tag{25}$$

$$p(x,y) = x^3 + y^3 - 1/2, \tag{26}$$

and the Reynolds number is set to 400. The boundary condition and the external force are calculated from the stational Navier-Stokes equations.

We have divided Ω into a union of uniform $N \times N \times 2$ triangular elements, where $N = 4$, 8, 16 or 32. We have computed in two domain-decomposed ways, where the number of subdomains in each direction is 2 or 4. Figure 4 shows the domain-decomposition and the triangulation in the case $N = 32$ and 4×4 subdomains. Starting from an initial condition for the velocity, the numerical solution is expected to converge to the stational solution in time-marching. If $\max_{k,i} |u_{k,i}^n - u_{k,i}^{n-1}|/\tau < 10^{-5}$ is satisfied, we judge that the numerical solution has converged and stop the computation. Computation parameters are set as $\tau = 0.24/N$ and $\alpha = 1.0$ (the latter is the stabilizing parameter of the upwind approximation).

Figure 5 shows errors between the obtained numerical solutions and the exact solu-

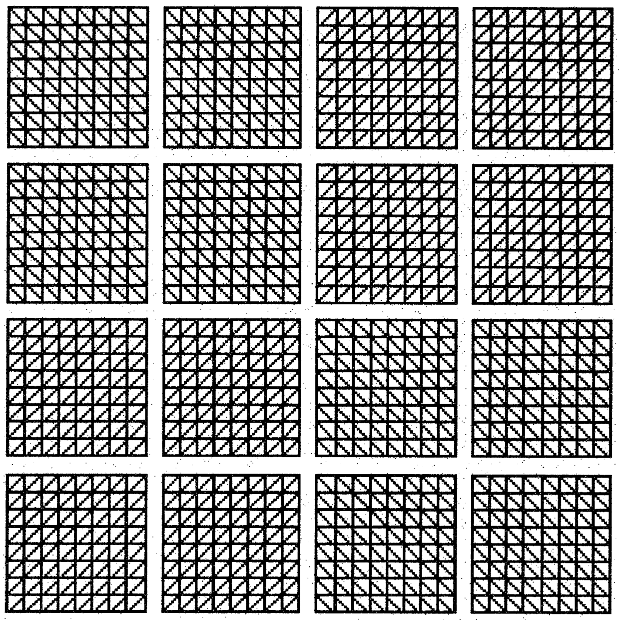


Figure 4: Domain-decomposition(4×4) and the triangulation$(N = 32)$

tions. They are measured by

$$|v|_{V_h} = \left\{ \sum_{k=1}^{K} |v|^2_{(H^1(\Omega_k))^2} \right\}^{1/2}, \quad ||q||_{Q_h} = \left\{ \sum_{k=1}^{K} ||q||^2_{L^2(\Omega_k)} \right\}^{1/2}, \quad ||\lambda||_{W_h} = \max_m \max_{\Gamma_m} |\lambda|.$$

Results of the non-domain-decomposition case are also plotted in the figure. We can observe that the errors of the velocity and the pressure realize the optimal convergence rate of the iso-P2 P1/P1 elements ,that is $O(h)$, regardless of choice of $W_{m,h}$. In the first case (iso-P2 P1 element for $W_{m,h}$), the maximum error of the Lagrange multiplier does not converge to 0 when h tends to 0. It may indicate the appearance of some spurious Lagrange multiplier modes, since the degree of freedom of the Lagrange multiplier is larger than that of jump of the velocity in the choice. In the latter two cases the convergence of the Lagrange multiplier has also observed. The third case (P1 element for $W_{m,h}$) shows the best property with respect to the convergence of the Lagrange multiplier.

Since the conventional P1 element has the smallest degree of freedom of the Lagrange multiplier, it can decrease the amount of computation steps in a iteration time in the conjugate gradient solver. Hence we adopt P1 element for $W_{m,h}$ in the following.

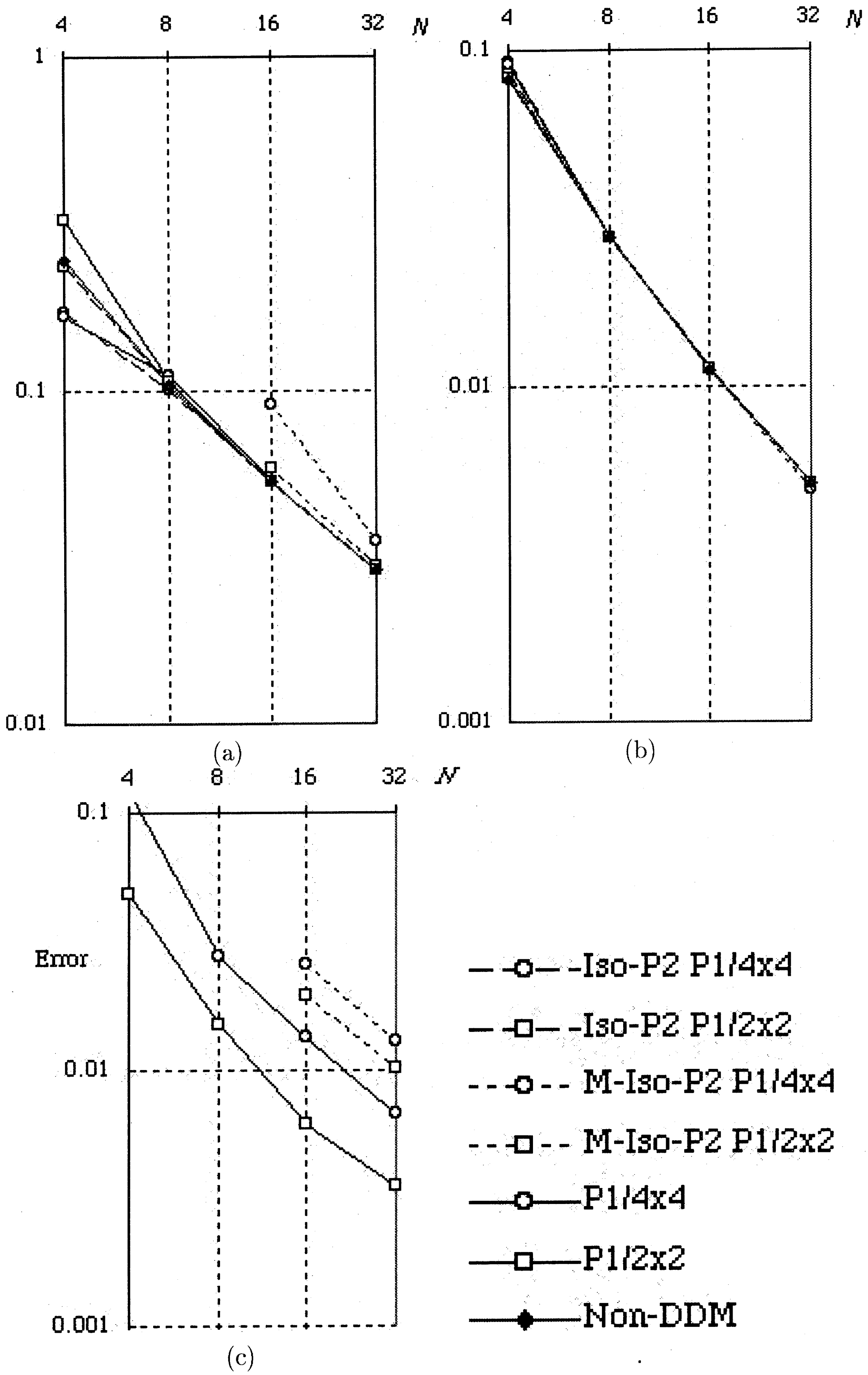


Figure 5: Errors of (a)velocity, (b)pressure and (c)Lagrange multiplier in the test problem

Table 1: Domain-decomposition vs. computation times

Domain-decomposition	Computation times(sec.)
4×4	415.4
4×2	510.9
2×2	720.1
2×1	1231.
1×1	1090.

3.2 Lid-driven cavity flow problem

We next computed the two-dimensional lid-driven cavity flow problem. The Reynolds number is 400. The domain $\Omega = (0,1) \times (0,1)$ is divided into a uniform $24 \times 24 \times 2$ triangular subdivision. We chose $\tau = 0.01$ and $\alpha = 1$. We computed in the cases of 4×4, 4×2, 2×2, 2×1 and 1×1 domain-decompositions. The computation time are listed in Table 1. We see that the computation time becomes shorter as the number of subdomains (i.e. processors) increases, except for the case of 2×1 subdomains. The velocity vectors and the pressure contours of the computed stationary flow in 4×4 subdomains are shown in Figure 6. We can observe that the flow is captured well in the domain decomposition algorithm.

4 Conclusion

We have considered a domain decomposition algorithm of the finite element scheme for the Navier-Stokes equations. In the scheme, subdomain-wise finite element spaces by iso-P2 P1/P1 elements are constructed and weak continuity of the velocity between subdomains are treated by a Lagrange multiplier method. This domain decomposition algorithm has advantages such as: (i) each subdomain-wise problem is a consistent discretized pressure Poisson equation so that it is regular, (ii) the size of a system of linear equations to be solved by the conjugate gradient method is smaller than that of the original consistent discretized pressure Poisson equation. For the discretization of the Lagrange multiplier,

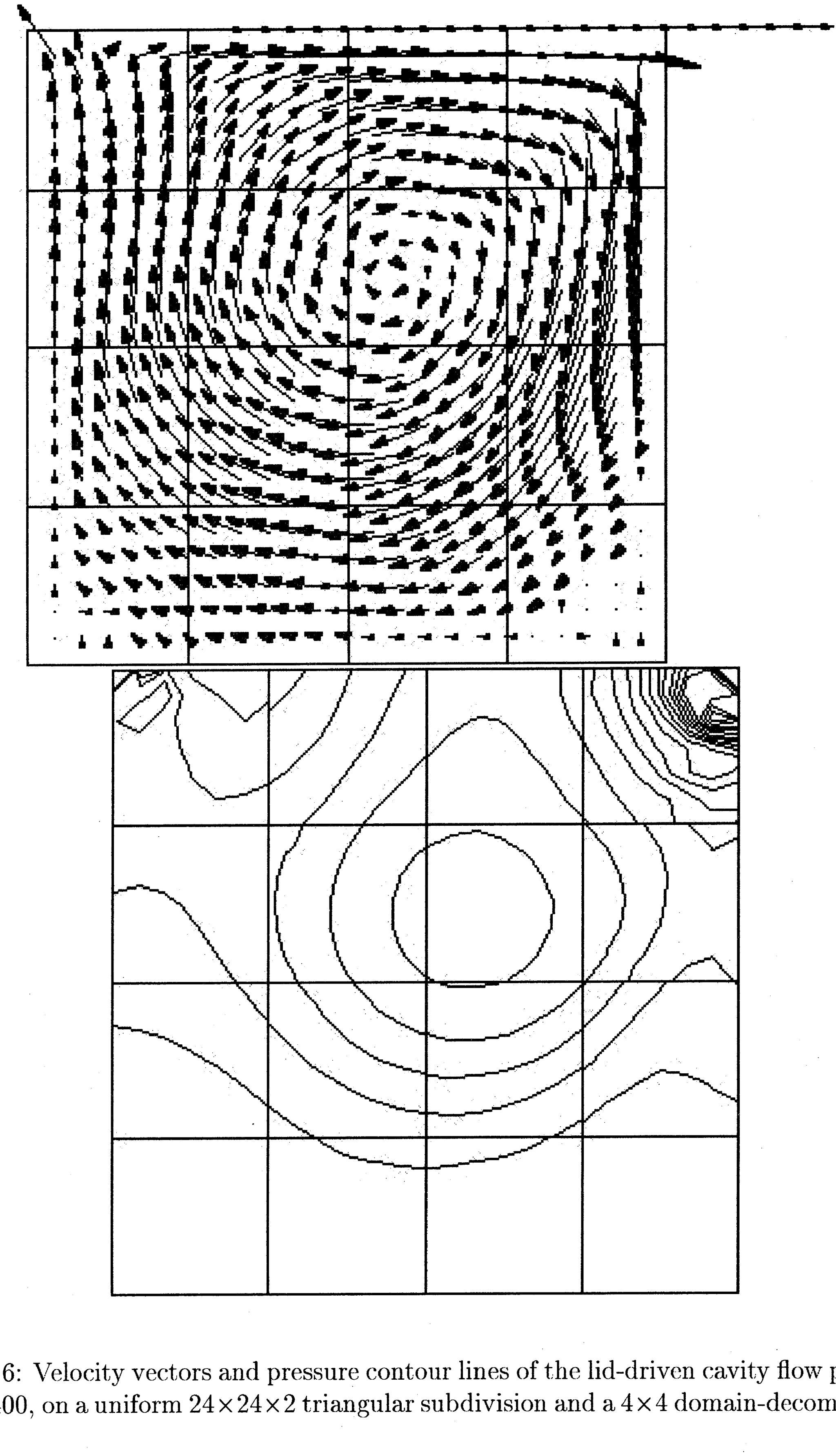


Figure 6: Velocity vectors and pressure contour lines of the lid-driven cavity flow problem, $Re = 400$, on a uniform $24\times24\times2$ triangular subdivision and a 4×4 domain-decomposition

we compared three cases: the conventional iso-P2 P1 element, a modified iso-P2 P1 element having no freedoms at both edges of interfaces, and the conventional P1 element. In every case, we checked numerically in a sample problem that the scheme could produce solutions which converged to the exact solution at the optimal rates for the velocity and the pressure. In the latter two cases we have also observed the convergence of the Lagrange multiplier. Employing the conventional P1 element, we have computed the lid-driven cavity flow problem. The computation time becomes shorter when the number of processor increases. We therefore recommend iso-P2 P1(u)/P1(p)/P1(λ) element in this method.

Acknowledgements

The author wish to thank Professor Masahisa Tabata (Kyushu University) for many valuable discussions and suggestions. The author was supported by the Ministry of Education, Science and Culture of Japan under Grant-in-Aid for Encouragement of Young Scientists, No.09740148. The computations have been done on Intel Paragon XP/S in INSAM(Institute for Numerical Simulations and Applied Mathematics), Hiroshima University.

References

[1] Y. Achdou and Y. A. Kuznetsov, Algorithm for a non conforming domain decomposition method, *Tech. Rep.* **296**, Ecole Polytechnique, 1994.

[2] Y. Achdou and O. Pironneau, A fast solver for Navier-Stokes equations in the laminar regime using mortar finite element and boundary element methods, *SIAM. J. Numer. Anal.*, **32**, 985-1016, 1995.

[3] R. Glowinski, T.-W. Pan and J. Périaux, A one shot domain decomposition/fictitious domain method for the Navier-Stokes equations, *Domain Decomposition Methods in*

Scientific and Engineering Computing, Proc. 7th Int. Conf. on Domain Decomposition, D. E. Keyes and J. Xu (eds.), *Contemporary Mathematics* **180**, A. M. S., Providence, Rhode Island, 211-220, 1994.

[4] A. Suzuki, Implementation of domain decomposition methods on parallel computer ADENART, *Parallel Computational Fluid Dynamics: New Algorithms and Applications*, N. Satofuka, J. Periaux and A. Ecer (eds.), Elsevier, 1995.

[5] M. Bercovier and O. Pironneau, Error estimates for finite element method solution of the Stokes problem in the primitive variable, *Numer. Math.*, **33**, 211-224, 1979.

[6] M. Tabata and S. Fujima, An upwind finite element scheme for high-Reynolds-number flows, *Int. J. Num. Meth. Fluids*, **12**, 305-322, 1991.

[7] C. Bernardi, Y. Maday and A. Patera, A new nonconforming approach to domain decomposition: the mortar element method, *Nonlinear Partial Differential Equations and their Applications* **XI**, H. Brezis and J. L. Lions (eds.), Longman Scientific & Technical, Essex, UK, 13-51, 1994.

[8] P. M. Gresho, S. T. Chan, R. L. Lee and C. D. Upson, A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations, Part 1: Theory, *Int. J. Num. Meth. Fluids*, **4**, 557-598, 1984.

[9] S. Fujima, An upwind finite element scheme for the Navier-Stokes equations and its domain decomposition algorithm, *Thesis*, Hiroshima University, 1997.

[10] S. Fujima, Implementation of mortar element method for flow problems in the primitive variables, to appear in *Int. J. Comp. Fluid Dyn.*

[11] S. Fujima, Iso-P2 P1/P1/P1 Domain-decomposition/Finite-element Method for the Navier-Stokes equations, to appear.