An Uzawa Domain Decomposition Method for Stokes Problem

Jonas Koko[1] and Taoufik Sassi[2]

[1] LIMOS, Université Blaise-Pascal – CNRS UMR 6158
Campus des Cézeaux, 63173 Aubière cedex, France, koko@isima.fr
[2] LMNO, Université de Caen – CNRS UMR 6139
BP 5186, 14032 Caen Cedex, France, Taoufik.Sassi@math.unicaen.fr

1 Introduction

The Stokes problem plays an important role in computational fluid dynamics since it is encountered in the time discretization of (incompressible) Navier-Stokes equations by operator-splitting methods [2, 3]. Space discretization of the Stokes problem leads to large scale ill-conditioned systems. The Uzawa (preconditioned) conjugate gradient method is an efficient method for solving the Stokes problem. The Uzawa conjugate gradient method is a decomposition coordination method with coordination by a Lagrange multiplier.

The paper is organized as follows. In the next section we recall the Stokes problem in its strong and constrained minimization formulations. Then we introduce an additional (interface) continuity condition in the resulting constrained minimization problem and we derive a decomposition coordination method with two multiplier: the pressure (for the divergence free condition) and the interface multiplier (for the continuity condition). A domain decomposition algorithm which solves on each step an uncoupled scalar Poisson sub-problem is defined in § 3.3 and the paper concludes by several numerical realizations.

2 Model Problem

Let Ω be a bounded domain in $\mathbb{R}^d$ ($d = 2, 3$) with Lipschitz-continuous boundary Γ. We consider in Ω the Stokes problem

$$\alpha u - \nu \Delta u + \nabla p = f, \quad \text{in } \Omega, \tag{1}$$

$$\nabla \cdot u = 0, \quad \text{in } \Omega, \tag{2}$$

$$u = 0, \quad \text{on } \Gamma, \tag{3}$$

where $u = u(x)$ is the velocity vector, $p = p(x)$ is the pressure and $f = f(x)$ is the field of external forces.

In (1), $\alpha \geq 0$ is an arbitrary constant. If $\alpha = 0$, (1)–(2) is the classical Stokes problem. The constant $\nu \geq 0$ is the kinematic viscosity and if $\nu = 0$, (1)–(2) turns out to be L^2 projection encountered in time discretization of Navier-Stokes equations (see e.g. [2, 3]).

We need the following functional spaces

$$V = \left\{ v \in (H^1(\Omega))^d \ : \ v = 0 \text{ on } \Gamma \right\}, \quad L_0^2(\Omega) = \left\{ q \in L^2(\Omega) \ : \ \int_\Omega q \, \mathrm{d}x = 0 \right\},$$

and the bilinear form

$$a(u, v) = \int_\Omega (\alpha u \cdot v + \nu \nabla u : \nabla v) \, \mathrm{d}x.$$

Let us introduce the quadratic functional $J : V \to \mathbb{R}$ defined by

$$J(v) = \frac{1}{2} a(v, v) - (f, v)_\Omega,$$

where $(\cdot, \cdot)_\Omega$ denotes the standard L^2 scalar product over Ω. Assuming $\mathrm{mes}(\Gamma) > 0$, the functional J is convex, G-differentiable and coercive on V.

With the above preparations, the Stokes problem (1)–(3) can be formulated as the following constrained minimization problem

Find $u \in V_\sigma = \{ v \in V \mid \nabla \cdot v = 0 \text{ on } \Omega \}$ such that

$$J(u) \leq J(v), \quad \forall v \in V_\sigma. \tag{4}$$

Since V_σ is closed and convex subset of V and the functional J is strongly convex and coercive, the constrained minimization problem (4) has a unique solution. The pressure p is recovered as the Lagrange multiplier associated with the divergence constraint (2).

3 Uzawa Domain Decomposition for Stokes Problem

We now study the domain decomposition of (4). We first rewrite (4) in the following more useful form.

Find $u \in V$ such that:

$$J(u) \leq J(v) \quad \forall v \in V, \tag{5}$$

$$\text{subject to } \nabla \cdot u = 0 \text{ in } \Omega. \tag{6}$$

Let $\{\Omega_1, \ \Omega_2\}$ be a partition of Ω, as shown in Fig. 1, and let

$$\Gamma_{12} = \Gamma_{21} = \partial\Omega_1 \cap \partial\Omega_2, \quad \Gamma_i = \Gamma \cap \partial\Omega_i,$$
$$v_i = v|_{\Omega_i}, \quad p_i = p|_{\Omega_i}, \quad V_i = \left\{ v \in H^1(\Omega_i), \ \ v|_{\Gamma_i} = 0 \right\}.$$

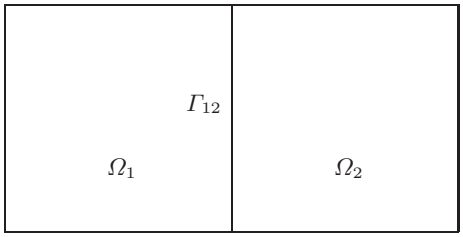

Fig. 1. Decomposition of Ω into two subdomains.

Inner products over Ω_i and Γ_{12} are defined by

$$(u,v)_{\Omega_i} = \int_{\Omega_i} uv\mathrm{d}x, \quad \text{and} \quad (u,v)_{\Gamma_{12}} = \int_{\Gamma_{12}} uv\mathrm{d}\Gamma.$$

Restrictions of the functionals a and J over Ω_i are denoted by a_i and J_i, respectively. To simplify, let us denote vector-valued functions and spaces by bold-face letters, i.e. $\boldsymbol{u} = (u_1, u_2)$, $\mathbf{V} = V_1 \times V_2$, etc.

3.1 Lagrangian Formulation and Dual Problem

Consider the following constrained minimization problem

Find $(u_1, u_2) \in V_1 \times V_2$ such that:

$$J_1(u_1) + J_2(u_2) \leq J_1(v_1) + J_2(v_2) \quad \forall (v_1, v_2) \in V \tag{7}$$

$$\nabla \cdot u_i = 0 \quad \text{in } \Omega_i \tag{8}$$

$$u_1 - u_2 = 0 \quad \text{on } \Gamma_{12}. \tag{9}$$

It is obvious that (7)–(9) is equivalent to (5)–(6). The Lagrangian functional associated with (7)–(9) is

$$\mathcal{L}(\boldsymbol{v}; \boldsymbol{q}, \mu) = \sum_{i=1}^{2} [J_i(v_i) - (q_i, \nabla \cdot v_i)_{\Omega_i}] - (\mu, [\boldsymbol{u}])_{\Gamma_{12}}. \tag{10}$$

where we have set $[\boldsymbol{u}] = (u_1 - u_2)_{|\Gamma_{12}}$. Let us introduce the set

$$\boldsymbol{P} = \left\{ (q_1, q_2) \in L^2(\Omega_1) \times L^2(\Omega_2) \ : \ \int_{\Omega_1} q_1(x)\mathrm{d}x + \int_{\Omega_2} q_2(x)\mathrm{d}x = 0 \right\}.$$

Then the solution of (7)–(9) is characterized by the following saddle-point problem

Find $(\boldsymbol{u}; \boldsymbol{p}, \lambda) \in \boldsymbol{V} \times \boldsymbol{P} \times L^2(\Gamma_{12})$ such that:

$$\mathcal{L}(\boldsymbol{u}; \boldsymbol{q}, \mu) \leq \mathcal{L}(\boldsymbol{v}; \boldsymbol{q}, \mu) \leq \mathcal{L}(\boldsymbol{v}; \boldsymbol{p}, \lambda) \quad \forall (\boldsymbol{v}; \boldsymbol{p}, \lambda) \in \boldsymbol{V} \times \boldsymbol{P} \times L^2(\Gamma_{12}). \tag{11}$$

Solving (11) is equivalent to solving the saddle point equations

$$a_i(u_i, v_i) = (f_i, v_i)_{\Omega_i} + (p_i, \nabla \cdot v_i)_{\Omega_i} + (-1)^{i+1}(\lambda, v_i) \quad \forall v_i \in V^i, \tag{12}$$

$$-(q_i, \nabla \cdot u_i)_{\Omega_i} = 0 \quad \forall q_i \in L^2(\Omega_i) \tag{13}$$

$$(\mu, [\boldsymbol{u}])_{\Gamma_{12}} = 0 \quad \forall \mu \in L^2(\Gamma_{12}). \tag{14}$$

The main advantage of the saddle point formulation is that (12) reduces to $2d$ uncoupled scalar Poisson problems if p_i and λ are known.

Suppose that $u_i = u_i(p_i, \lambda)$ is the solution of the Poisson equation (12). For convenience, in the sequel, we suppress the dependence of u_i on (p_i, λ). setting $v_i = u_i$ in (12) and substituting in (10) we get

$$J^*(\boldsymbol{p}, \lambda) := \mathcal{L}(\boldsymbol{u}(\boldsymbol{p}); \boldsymbol{p}, \lambda) = -\frac{1}{2}\sum_{i=1}^{2} a_i(u_i, u_i). \tag{15}$$

Since the mapping $(\boldsymbol{p}, \lambda) \mapsto \boldsymbol{u}(\boldsymbol{p}, \lambda)$ is linear and the bilinear forms a_i are strongly convex, we deduce that J^* is a strictly concave functional. The dual problem of (7)-(9) is the maximization problem

Find $(\boldsymbol{p}, \lambda) \in \boldsymbol{P} \times L^2(\Gamma_{12})$ such that:

$$J^*(\boldsymbol{p}, \lambda) \geq J^*(\boldsymbol{q}, \mu) \quad \forall (\boldsymbol{q}, \mu) \in \boldsymbol{P} \times L^2(\Gamma_{12}). \tag{16}$$

To derive a maximization method for (16), we need some differential informations on J^*.

3.2 Sensitivity Analysis

The sensitivity problem (which measures the variation of u_i vs. (p_i, λ)) is

$$\tilde{u}_i \in V_i;\ a_i(\tilde{u}_i, v_i) = (\tilde{p}_i, \nabla \cdot v_i)_{\Omega_i} + (-1)^{i+1}(\tilde{\lambda}, v_i)_{\Gamma_{12}} \quad \forall v_i \in V_i \tag{17}$$

so that $u_i(p_i + t\tilde{p}_i, \lambda + t\tilde{\lambda}) = u_i(p_i, \lambda) + t\tilde{u}_i$. The directional derivative of J^* is given by

$$\frac{\partial J^*(\boldsymbol{p}, \lambda)}{\partial(\boldsymbol{p}, \lambda)} \cdot (\tilde{\boldsymbol{p}}, \tilde{\lambda}) = -\sum_{i=1}^{2} a_i(u_i, \tilde{u}_i), \quad \forall (\tilde{\boldsymbol{p}}, \tilde{\lambda}), \tag{18}$$

where $\tilde{u}_i$ is the solution of the sensitivity problem (17). If we set $v_i = u_i$ in (17) and substitute the result into (18), we get

$$\frac{\partial J^*(\boldsymbol{p}, \lambda)}{\partial(\boldsymbol{p}, \lambda)} \cdot (\tilde{\boldsymbol{p}}, \tilde{\lambda}) = -\sum_{i=1}^{2} (\tilde{p}_i, \nabla \cdot u_i)_{\Omega_i} + (\tilde{\lambda}, [\boldsymbol{u}])_{\Gamma_{12}}. \tag{19}$$

Setting $g_i = \nabla_{p_i} J^*$ and $\gamma = \nabla_{\lambda} J^*$, the gradient of J^* with respect to p_i and λ, respectively, we deduce from (19) that

$$g_i = -\nabla \cdot u_i, \quad i = 1, 2, \tag{20}$$

$$\gamma = [\boldsymbol{u}], \tag{21}$$

for the standard L^2 scalar product.

Let $(\tilde{\boldsymbol{p}}, \tilde{\lambda})$ be a search direction for J^*. Since J^* is quadratic and concave, the best search direction is a conjugate gradient direction. At each iteration k, the conjugate gradient direction $(\tilde{\boldsymbol{p}}^k, \tilde{\lambda}^k)$ is given by

$$\beta_k = \left[\|\boldsymbol{g}^k\|^2_{L^2(\Omega)} + \|\gamma^k\|^2_{L^2(\Gamma_{12})}\right]\left[\|\boldsymbol{g}^{k-1}\|^2_{L^2(\Omega)} + \|\gamma^{k-1}\|^2_{L^2(\Gamma_{12})}\right]^{-1}, \quad (22)$$

$$\tilde{\boldsymbol{p}}^k = \boldsymbol{g}^k + \beta_k \tilde{\boldsymbol{p}}^{k-1}, \quad \tilde{\lambda}^k = \gamma^k + \beta_k \tilde{\lambda}^{k-1}. \quad (23)$$

We need to determine the optimal step size to complete the iteration. The optimal step size is computed as the maximizer of the real-valued function $\rho(t) = J^*(\boldsymbol{p} + t\tilde{\boldsymbol{p}}, \lambda + t\tilde{\lambda})$. Since J^* is quadratic and strictly concave, the maximizer of ρ is the unique solution of the linear equation

$$\rho'(t) = \frac{\partial J^*(\boldsymbol{p} + t\tilde{\boldsymbol{p}}, \lambda + t\lambda)}{\partial(\boldsymbol{p}, \lambda)} \cdot (\tilde{\boldsymbol{p}}, \tilde{\lambda}) = 0.$$

We deduce, from (19), that the optimal step size is

$$t = \frac{(\tilde{\lambda}, [\boldsymbol{u}])_{\Gamma_{12}} - \sum_i (\tilde{p}_i, \nabla \cdot u_i)_{\Omega_i}}{(\tilde{\lambda}, [\tilde{\boldsymbol{u}}])_{\Gamma_{12}} - \sum_i (\tilde{p}_i, \nabla \cdot \tilde{u}_i)_{\Omega_i}}. \quad (24)$$

3.3 Uzawa Conjugate Gradient Domain Decomposition Algorithm

With the preparations described in the previous subsection, we can now present our Uzawa/conjugate gradient domain decomposition algorithm for the Stokes problem.

Algorithm DDM/P

Iteration $k = 0$. $\boldsymbol{p}^0 \in \boldsymbol{P}$ and $\lambda^0 \in L^2(\Gamma_{12})$ given

Compute $u_i^0 \in V_D^i$ via

$$a_i(u_i^0, v_i) = (f_i, v_i)_{\Omega_i} + (p_i^0, \nabla \cdot v_i)_{\Omega_i} + (-1)^{i+1}(\lambda^0, v_i)_{\Gamma_{12}}, \quad \forall v_i \in V^i \quad (25)$$

Compute $\boldsymbol{g}^0 \in \boldsymbol{P}$ via

$$(g_i^0, q_i)_{\Omega_i} = -(\nabla \cdot u_i^0, q_i)_{\Omega_i} \quad \forall q_i \in L^2(\Omega_i),\ i = 1, 2.$$

Compute $\gamma^0 \in L^2(\Gamma_{12})$ via

$$(\gamma^0, \mu)_{\Gamma_{12}} = ([\boldsymbol{u}^0], \mu)_{\Gamma_{12}} \quad \forall \mu \in L^2(\Gamma_{12})$$

Initial direction: $\tilde{\boldsymbol{p}}^0 = \boldsymbol{g}^0$, $\tilde{\lambda}^0 = \gamma^0$

Iteration $k \geq 0$. Assuming $\boldsymbol{p}^k$, λ^k, $\tilde{\boldsymbol{p}}^k$, $\tilde{\lambda}^k$, $\boldsymbol{g}^k$, γ^k and $\boldsymbol{u}^k$ are known

Sensitivity problems: compute $\tilde{u}_i \in V^i$ via

$$a_i(\tilde{u}_i^k, v_i) = (\tilde{p}_i^k, \nabla \cdot v_i)_{\Omega_i} + (-1)^{i+1}(\tilde{\lambda}^k, v_i)_{\Gamma_{12}} \quad \forall v_i$$

Step size: Compute t_k using (24)

Update:

$$\lambda^{k+1} = \lambda^k + t_k\tilde{\lambda}^k,$$
$$p_i^{k+1} = p_i^k + t_k\tilde{p}_i^k, \quad u_i^{k+1} = u_i^k + t_k\tilde{u}_i^k,\ i = 1, 2.$$

Gradient: Solve the gradient systems

$$(g_i^{k+1}, q_i)_{\Omega_i} = -(\nabla \cdot u_i^{k+1}, q_i)_{\Omega_i} \quad \forall q_i \in L^2(\Omega_i),\ i = 1, 2. \tag{26}$$
$$(\gamma^{k+1}, \mu)_{\Gamma_{12}} = ([\boldsymbol{u}^{k+1}], \mu)_{\Gamma_{12}} \quad \forall \mu \in L^2(\Gamma_{12}). \tag{27}$$

Conjugate gradient direction: Compute $(\tilde{\boldsymbol{p}}^{k+1}, \tilde{\lambda}^{k+1})$ using (22)–(23)

We iterate until the gradient is "sufficiently" small, i.e.

$$\left\|\boldsymbol{g}^k\right\|^2_{L^2(\Omega)} + \left\|\gamma^k\right\|^2_{L^2(\Gamma_{12})} < \varepsilon\left(\left\|\boldsymbol{g}^0\right\|^2_{L^2(\Omega)} + \left\|\gamma^0\right\|^2_{L^2(\Gamma_{12})}\right) \tag{28}$$

where $\varepsilon > 0$ is a given tolerance. Each iteration, we solve in parallel $2d$ scalar Poisson problems. The parallelizability of the algorithm is therefore obvious.

Note that if $\boldsymbol{p}^0 = (p_1^0, p_2^0) \in \boldsymbol{P}$ and $\boldsymbol{g}^k = (g_1^k, g_2^k) \in \boldsymbol{P}$ for all $k \geq 0$, then $\boldsymbol{p}^k = (p_1^k, p_2^k) \in \boldsymbol{P}$ for all $k \geq 1$. To compute $\boldsymbol{g}^k$ in $\boldsymbol{P}$, we first solve the gradient systems (26) separately to obtain $\bar{g}_i^k$. The gradient $\boldsymbol{g}^k = (g_1^k, g_1^k)$ is then obtained by the projection of $(\bar{g}_1^k, \bar{g}_2^k)$ onto $\boldsymbol{P}$.

4 Numerical Experiments

The domain decomposition algorithms presented in the previous sections were implemented in Fortran 90, on a 172-core Linux cluster, using an MPI library. MPI subroutines are used only for solving in parallel uncoupled Poisson problems. All linear systems involved are solved by a preconditioned conjugate gradient algorithm. The preconditioning is obtained by an incomplete Choleski factorization with drop tolerance varying from 10^{-5} to 10^{-3}. For discrete velocity-pressure spaces, we use the P^1-iso-P^2/P^1 element. These spaces are well known to satisfy the discrete Babuska-Brezzi inf-sup condition.

We consider the domain $\Omega = (0,\ a) \times (0,\ 1)$ and we take $\alpha = 0$ and $\nu = 1$ in (1). The right-hand side in (1) is adjusted such that the exact solution is

$$u_1(x, y) = (1 - \cos(2\pi x/a))\sin(2\pi y),\ \ u_2(x, y) = \frac{1}{a}(\cos(2\pi y) - 1)\sin(2\pi x/a)$$
$$p(x, y) = \frac{2\pi}{a}(cos(2\pi y) - \cos(2\pi x/a))$$

In our numerical experiments, $a = 10$.

We first consider the classical Stokes problem (i.e. $\alpha = 0$ and $\nu = 1$). Table 1 shows the L^2 and H^1 errors for u_h and p_h ($h = 1/256$) obtained using Algorithm DDM/P with decomposition into 2, 4, 8, 16, 32 and 64 subdomains. We notice that

the errors are comparable with those obtained with the standard Uzawa/conjugate gradient for Stokes problem ($N_{SD} = 1$). Note that for $h = 1/256$, we have $2561 \times 257 = 658,177$ nodes in the fine (velocity) mesh. Figure 2 shows that the speed-up is significant for $N_{SD} \geq 8$.

Table 1. H^1 and L^2 errors for various number of subdomains N_{SD} for Algorithm DDM/P with $h = 1/256$.

N_{SD}	Iterations	$\|u - u_h\|_{L^2(\Omega)}$	$\|u - u_h\|_{H^1(\Omega)}$	$\|p - p_h\|_{L^2(\Omega)}$
1	20	3.7602×10^{-4}	1.2293×10^{-1}	7.1937×10^{-4}
2	110	3.7611×10^{-4}	1.2295×10^{-1}	7.2209×10^{-4}
4	120	3.7608×10^{-4}	1.2295×10^{-1}	7.5186×10^{-4}
8	131	3.7616×10^{-4}	1.2295×10^{-1}	7.8013×10^{-4}
16	158	3.7610×10^{-4}	1.2295×10^{-1}	8.2989×10^{-4}
32	205	3.7611×10^{-4}	1.2295×10^{-1}	9.4120×10^{-4}
64	258	3.7592×10^{-4}	1.2295×10^{-1}	1.1080×10^{-3}

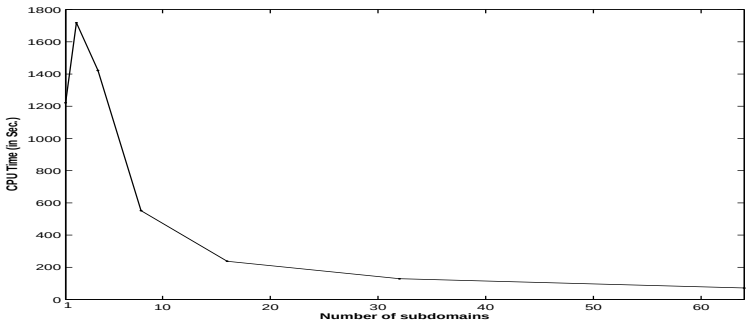

Fig. 2. CPU times vs. number of subdomains for Algorithm DDM/P with $\alpha = 0$, $\nu = 1$ and $h = 1/256$.

We now consider the case $\alpha = 1$ and $\nu = 10^{-3}$. For the one-subdomain case, we use the Cahouet-Chabard preconditioner [1], see also [2, 3]. This preconditioner is the best for the Uzawa/conjugate gradient algorithm for the generalized Stokes problem with $\alpha/\nu \gg 1$. Figure 3 shows the CPU times obtained with our domain

decomposition algorithm. We notice that the speed-up is significant when $N_{SD} \geq 16$.

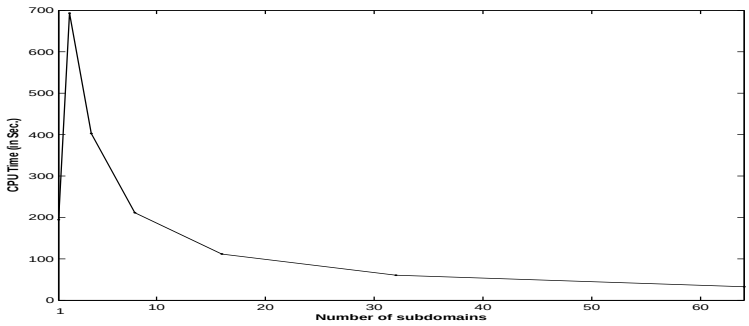

Fig. 3. CPU times vs. number of subdomains for Algorithm DDM/P with $\alpha = 1$, $\nu = 10^{-3}$ and $h = 1/256$.

5 Conclusion

A Uzawa domain decomposition algorithm for Stokes problem has been introduced. The standard L^2-scalar product is used for computing the gradient (26)–(27). This approach is easy to implement and has a significant speed-up when the number of subdomains is larger then 10. Nevertheless, it leads to a h-dependent algorithm. To improve this algorithm, different preconditioners will be investigated :

- the Steklov-Poincaré operator on the interface (see e.g. [4, 5]),
- the Cahouet-Chabard preconditioner [1] for the pressure multiplier.

A coarse component of the preconditioner will be studied. Indeed, the iteration counts for large N_{SD} could indicate the lack of a coarse component in the preconditioner.

References

1. J. Cahouet and J.P. Chabard Some fast 3-D solvers for the generalized Stokes problem. *Int. J. Numer. Methods Fluids*, 8:269–295, 1988.
2. R. Glowinski Numerical methods for fluids (part 3). In Ciarlet P.G. and Lions J.L., editors, *Numerical Methods for Fluids (Part 3)*, volume IX of *Handbook of Numerical Analysis*, pp. 3–1074. North-Holland, Amsterdam, 2003.
3. R. Glowinski and P. Le Tallec *Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics*. SIAM, Philadelphia, PA, 1989.
4. J. Koko Convergence analysis of optimization-based domain decomposition methods for a bonded structure. *Appl. Numer. Math.*, 58:69–87, 2008.
5. P. Le Tallec and T. Sassi Domain decomposition with nonmatching grids: augmented Lagrangian approach. *Math. Comput.*, 64:1367–1396, 1995.