Boundary Integral-Based Domain Decomposition Technique for Solution of Navier Stokes Equations

N. Mai-Duy[1] **and T. Tran-Cong**[2]

Abstract: This paper presents a new domain decomposition technique based on the use of Boundary Integral Equations (BIEs) for the analysis of viscous flow problems. The domain of interest is divided into a number of non-overlapping subdomains and an iterative procedure is then employed to update the boundary conditions at interfaces. The new feature in the present work is that at each iteration, the relevant two subdomains, together containing a particular interface, are assumed to satisfy the governing BI equations which they do at the end of a convergent iterative process. Hence the boundary conditions on such an interface can be updated using the interior point formulas. Updating formulas based on standard and hypersingular BIEs are derived and the final forms obtained are simple. Furthermore, the internal point formula can be used as a means to estimate the initial interface solution. The proposed method is verified in conjunction with the BEM through the simulation of Poiseuille, driven cavity and backward facing step viscous flows.

keyword: domain decomposition, viscous flow, standard boundary integral equation, hypersingular boundary integral equation.

1 Introduction

Domain Decomposition (DD) methods have become necessary to deal with large industrial applications. A physical DD method is based on the assumption that the given analysis domain is partitioned into subdomains. The original problem can then be reformulated for each subdomain, yielding a family of smaller subproblems that are coupled one to another through the unknown, but common, solution at subdomain interfaces [Quarteroni and Valli (1999)].

Based on the concept of a spatial decomposition, the DD methods can be classified into two categories: overlapping and non-overlapping. The present work is concerned with the case of non-overlapping subdomains and the use of an iterative procedure to update the boundary conditions (from here on referred to as "updating mechanism") at interfaces for the analysis of viscous flows, which extends the earlier work on potential problems [Mai-Duy and Tran-Cong (2002)]. The new feature in the present updating mechanism (with associated "updating formulas") is that at each iteration, the relevant two subdomains, together containing a particular interface, are assumed to satisfy the governing BI equations, which they do at the end of a convergent iterative process, and hence the boundary conditions on such an interface can be updated using the interior point formulas. Updating formulas based on standard and hypersingular BIEs are derived and the final forms obtained are quite simple, where the need for the evaluation of volume integrals is eliminated and the boundary integrals are confined to subdomain interfaces only. Furthermore, the BIE formula for interior points can also be utilised as a means to estimate the initial interface solution.

The present method is different from the subregioning technique [Brebbia and Dominguez (1992)] in some aspects. The latter involves the formation and solution of a system of equations, where the total number of unknowns is the sum of nodal degrees of freedom associated with the real boundaries and twice the number of nodal degrees of freedom associated with subdomain interfaces. In the present method, no such a large system of equations is required. However, it is necessary to employ an iterative process. Furthermore, subdomain problems obtained here are relatively independent (in the sense that they are separate problems once the interface conditions are specified) and hence can be solved in parallel using any suitable numerical method such as BEM

[1] Faculty of Engineering and Surveying, University of Southern Queensland, Toowoomba Qld 4350, Australia.

[2] Corresponding author, Faculty of Engineering and Surveying, University of Southern Queensland, Toowoomba Qld 4350, Australia, email: trancong@usq.edu.au, Fax: +61 7 46312526.

or FEM. The BEM works very well for linear and mildly nonlinear problems, for some of which the discretization needs be performed only on the boundaries [Han and Atluri(2002,2003); Tsai, Young, and Cheng (2002); Agnantiaris and Polyzos (2003); Zhang and Savaidis (2003)]. The parallelisation of the most time-consuming part can therefore lead to speed-up. Although the proposed method is well-suited to parallel distributed computing environments, such an investigation is beyond the scope of the present paper. In contrast to some other DD methods that use the exchange of data obtained at subdomain interfaces to update the interface conditions (which are referred to as the “conventional methods” from here on), the present method is based on the requirement of satisfying the governing BI equations on relevant subdomains. The remainder of the paper is organised as follows. In section 2, the standard and hypersingular BIEs governing viscous flow problems are briefly reviewed and then in section 3 they are utilised to develop formulas for updating the interface boundary conditions. In section 4, the proposed method is verified in conjunction with the BEM through the simulation of Poiseuille, driven cavity and backward facing step viscous flows. Section 5 gives some concluding remarks.

2 Boundary Integral Equations (BIEs)

Consider a two dimensional and steady flow of an incompressible viscous fluid of density ρ and viscosity μ. The equations for the conservation of momentum and mass take the forms

$$\begin{aligned} \mu(u_{1,11}+u_{1,22})-p_{,1} &= \rho(u_1u_{1,1}+u_2u_{1,2}), && (1)\\ \mu(u_{2,11}+u_{2,22})-p_{,2} &= \rho(u_1u_{2,1}+u_2u_{2,2}), && (2)\\ u_{1,1}+u_{2,2} &= 0, && (3)\end{aligned}$$

where u_j is the velocity vector and p the pressure. In the indicial notation used here, the subscripted comma denotes a partial derivative. The governing equations Eq. 1-Eq. 3 can be transformed into integral equations (IEs). For creeping flows, IEs contain only boundary integrals and can be written in the following two forms for interior points and smooth boundary points. The case of non-creeping flows is discussed in section 3.3.

2.1 Standard Boundary Integral Equation (SBIE) for creeping flows

For a point $\mathbf{y} \in \Gamma$, the SBIE is given by

$$c_{ij}(\mathbf{y})u_j(\mathbf{y}) = \int_\Gamma U_{ij}(\mathbf{y},\mathbf{x})t_j(\mathbf{x})d\Gamma(\mathbf{x}) - \mathrm{CPV}\int_\Gamma T_{ij}(\mathbf{y},\mathbf{x})u_j(\mathbf{x})d\Gamma(\mathbf{x}), \quad (4)$$

$$U_{ij}(\mathbf{y},\mathbf{x}) = \frac{1}{4\pi\mu}\left[\frac{r_i}{r}\frac{r_j}{r} - \delta_{ij}\ln(r)\right], \quad (5)$$

$$T_{ij}(\mathbf{y},\mathbf{x}) = -\frac{1}{\pi r}\left[\frac{r_i}{r}\frac{r_j}{r}\frac{\partial r}{\partial n}\right], \quad (6)$$

where CPV stands for Cauchy Principal Value, U_{ij} and T_{ij} are the Stokeslet fundamental solution and its associated traction respectively, t_j the traction vector, c_{ij} the free term coefficient which is $0.5\delta_{ij}$ if Γ is a smooth boundary, $r_i = x_i - y_i$, $r = \|\mathbf{x}-\mathbf{y}\|$ and n the direction of the outwardly unit vector normal to the boundary. The SBIE for boundary points contains weakly singular and CPV singular integrals.

For an interior point $\mathbf{y}$, $c_{ij} = \delta_{ij}$ and the second integral in Eq. 4 is a normal integral (i.e. not a CPV one).

2.2 Hypersingular Boundary Integral Equation (HBIE) for creeping flows

For a point $\mathbf{y} \in \Gamma$, the HBIE is given by

$$c_{ijkh}(\mathbf{y})\sigma_{kh}(\mathbf{y}) = \mathrm{CPV}\int_\Gamma Q_{ijk}(\mathbf{y},\mathbf{x})t_k(\mathbf{x})d\Gamma - \mathrm{HPV}\int_\Gamma P_{ijk}(\mathbf{y},\mathbf{x})u_k(\mathbf{x})d\Gamma, \quad (7)$$

$$Q_{ijk}(\mathbf{y},\mathbf{x}) = \frac{1}{\pi r}\frac{r_i}{r}\frac{r_j}{r}\frac{r_k}{r}, \quad (8)$$

$$P_{ijk}(\mathbf{y},\mathbf{x}) = \frac{\mu}{\pi r^2}\left\{2\frac{\partial r}{\partial n}\left[\frac{1}{2}\left(\delta_{ik}\frac{r_j}{r}+\delta_{jk}\frac{r_i}{r}\right) - 4\frac{r_i}{r}\frac{r_j}{r}\frac{r_k}{r}\right] + n_i\frac{r_j}{r}\frac{r_k}{r} + n_j\frac{r_i}{r}\frac{r_k}{r} + n_k\delta_{ij}\right\}, \quad (9)$$

where HPV stands for Hadamard Principal Value and the free term $c_{ijkh}\sigma_{kh}$ is $0.5\sigma_{ij}$ if Γ is a smooth boundary. The fundamental solutions Q_{ijk} and P_{ijk} are obtained by differentiating the kernels U_{ij} and T_{ij} respectively. The kernel singularities of HBIE are one order higher than the counterparts of SBIE, e.g. in 2D problems, U_{ij} and

T_{ij} exhibit singularities of order $(\ln r)$ and $(1/r)$ respectively, while the singularities of Q_{ijk} and P_{ijk} are of order $(1/r)$ and $(1/r^2)$ as r approaches zero. Eq. 7 is an integral equation of the second kind for the boundary traction involving a CPV singular and a hypersingular integral.

For an interior point $\mathbf{y}$, $c_{ijkh}\sigma_{kh} = \sigma_{ij}$ and both integrals in Eq. 7 are normal integrals (i.e. not a CPV or HPV one).

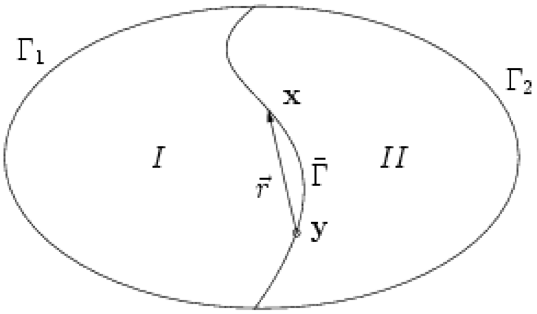

Figure 1 : BIE-based estimation of the interface boundary conditions.

3 BIE-Based Domain Decomposition Method

A mechanism for the iterative update of interface boundary u_i, t_i-values is developed in this section. Without loss of generality, the case of two subdomains and one interface (Fig. 1) is considered. The relevant volume and boundaries are $\Omega = I + II$, $\partial I = \Gamma_1 + \overline{\Gamma}$, $\partial II = \Gamma_2 + \overline{\Gamma}$ and $\partial\Omega = \Gamma_1 + \Gamma_2$. Let (u_i, t_i) be the solution to the problem and assume that the pressure is continuous across the interface $\overline{\Gamma}$.

3.1 Estimate of the interface boundary condition for u_i

A point $\mathbf{y}$ on the interface $\overline{\Gamma}$ can be considered as (a) an interior point relative to the domain Ω, (b) a boundary point relative to the domain I, and (c) a boundary point relative to the domain II, resulting in the corresponding SBIEs as follows

$$u_i(\mathbf{y}) = \int_{\Gamma_1} U_{ij}t_j d\Gamma + \int_{\Gamma_2} U_{ij}t_j d\Gamma - \int_{\Gamma_1} T_{ij}u_j d\Gamma - \int_{\Gamma_2} T_{ij}u_j d\Gamma, \quad (10)$$

$$\frac{1}{2}u_i(\mathbf{y}) = \int_{\Gamma_1} U_{ij}t_j d\Gamma + \int_{\overline{\Gamma}} U^I_{ij}t^I_j d\Gamma - \int_{\Gamma_1} T_{ij}u_j d\Gamma - \text{CPV}\int_{\overline{\Gamma}} T^I_{ij}u_j d\Gamma, \quad (11)$$

$$\frac{1}{2}u_i(\mathbf{y}) = \int_{\Gamma_2} U_{ij}t_j d\Gamma + \int_{\overline{\Gamma}} U^{II}_{ij}t^{II}_j d\Gamma - \int_{\Gamma_2} T_{ij}u_j d\Gamma - \text{CPV}\int_{\overline{\Gamma}} T^{II}_{ij}u_j d\Gamma. \quad (12)$$

where t^I_j and t^{II}_j are the tractions acting on I and II at point $\mathbf{y}$, respectively, and therefore formally $t^I_j = -t^{II}_j$; also the kernels $U^I_{ij} = U^{II}_{ij}$ and $T^I_{ij} = -T^{II}_{ij}$. At the iteration k where convergence has not been achieved yet, the actual solutions u_i and t_i at points along $\partial\Omega$ can be expressed as

$$u_i = \overline{u}_i + \Delta\overline{u}_i, \qquad \mathbf{x} \in \partial\Omega, \quad (13)$$

$$t_i = \overline{t}_i + \Delta\overline{t}_i, \qquad \mathbf{x} \in \partial\Omega, \quad (14)$$

where $\overline{u}_i$ and $\overline{t}_i$ are current approximations obtained at the iteration $(k-1)$ by solving subdomain problems while $\Delta\overline{u}_i$ and $\Delta\overline{t}_i$ are errors of $\overline{u}_i$ and $\overline{t}_i$ respectively. However, at points along the interface $\overline{\Gamma}$, there are two current approximations corresponding to the two adjacent subdomains I and II for each variable u_i and t_i, which can be written as

$$u_i = \overline{u}^I_i + \Delta\overline{u}^I_i, \qquad \text{or} \quad (15)$$

$$u_i = \overline{u}^{II}_i + \Delta\overline{u}^{II}_i, \quad (16)$$

$$t^I_i = \overline{t}^I_i + \Delta\overline{t}^I_i, \qquad \text{or} \quad (17)$$

$$t^{II}_i = \overline{t}^{II}_i + \Delta\overline{t}^{II}_i = -t^I_i, \qquad \mathbf{x} \in \overline{\Gamma}. \quad (18)$$

The substitution of Eq. 13 and Eq. 14 into Eq. 10; of Eq. 13, Eq. 14, Eq. 15 and Eq. 17 into Eq. 11; of Eq. 13, Eq. 14, Eq. 16 and Eq. 18 into Eq. 12 yields

$$u_i = \int_{\Gamma_1} U_{ij}(\overline{t}_j + \Delta\overline{t}_j)d\Gamma + \int_{\Gamma_2} U_{ij}(\overline{t}_j + \Delta\overline{t}_j)d\Gamma - \int_{\Gamma_1} T_{ij}(\overline{u}_j + \Delta\overline{u}_j)d\Gamma - \int_{\Gamma_2} T_{ij}(\overline{u}_j + \Delta\overline{u}_j)d\Gamma, \quad (19)$$

$$\frac{1}{2}(\overline{u}^I_i + \Delta\overline{u}^I_i) = \int_{\Gamma_1} U_{ij}(\overline{t}_j + \Delta\overline{t}_j)d\Gamma + \int_{\overline{\Gamma}} U^I_{ij}(\overline{t}^I_j + \Delta\overline{t}^I_j)d\Gamma - \int_{\Gamma_1} T_{ij}(\overline{u}_j + \Delta\overline{u}_j)d\Gamma - \text{CPV}\int_{\overline{\Gamma}} T^I_{ij}(\overline{u}^I_j + \Delta\overline{u}^I_j)d\Gamma, \quad (20)$$

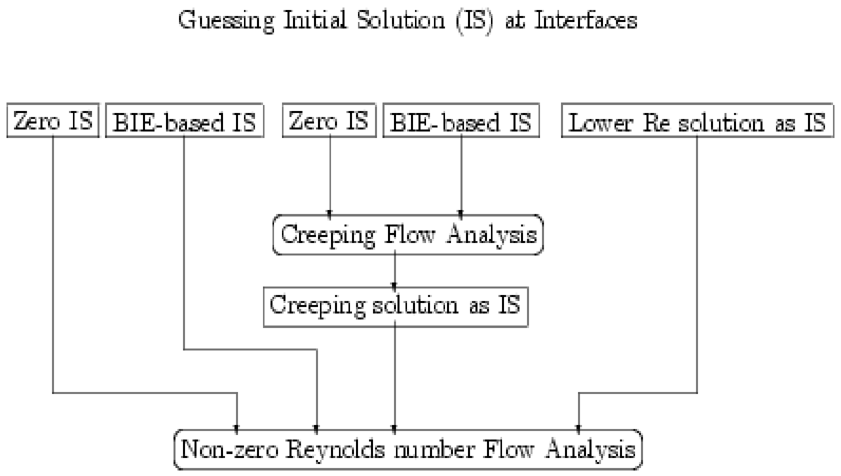

Figure 2 : Methods of guessing an initial solution at interfaces.

$$\frac{1}{2}(\overline{u}_i^{II}+\Delta\overline{u}_i^{II}) = \int_{\Gamma_2} U_{ij}(\overline{t}_j+\Delta\overline{t}_j)d\Gamma + \int_{\overline{\Gamma}} U_{ij}^{II}(\overline{t}_j^{II}+\Delta\overline{t}_j^{II})d\Gamma - \int_{\Gamma_2} T_{ij}(\overline{u}_j+\Delta\overline{u}_j)d\Gamma - \text{CPV}\int_{\overline{\Gamma}} T_{ij}^{II}(\overline{u}_j^{II}+\Delta\overline{u}_j^{II})d\Gamma. \quad (21)$$

The subtraction of Eq. 20 and Eq. 21 from Eq. 19 yields

$$u_i = \frac{1}{2}\left(\overline{u}_i^{I}+\Delta\overline{u}_i^{I}\right) - \int_{\overline{\Gamma}} U_{ij}^{I}(\overline{t}_j^{I}+\Delta\overline{t}_j^{I})d\Gamma + \text{CPV}\int_{\overline{\Gamma}} T_{ij}^{I}(\overline{u}_j^{I}+\Delta\overline{u}_j^{I})d\Gamma + \frac{1}{2}\left(\overline{u}_i^{II}+\Delta\overline{u}_i^{II}\right) - \int_{\overline{\Gamma}} U_{ij}^{II}(\overline{t}_j^{II}+\Delta\overline{t}_j^{II})d\Gamma + \text{CPV}\int_{\overline{\Gamma}} T_{ij}^{II}(\overline{u}_j^{II}+\Delta\overline{u}_j^{II})d\Gamma. \quad (22)$$

Since $U_{ij}^{I} = U_{ij}^{II}$ and $T_{ij}^{I} = -T_{ij}^{II}$, Eq. 22 can be rewritten as

$$u_i = \frac{1}{2}\left(\overline{u}_i^{I}+\overline{u}_i^{II}\right) - \int_{\overline{\Gamma}} U_{ij}^{I}(\overline{t}_j^{I}+\overline{t}_j^{II})d\Gamma + \text{CPV}\int_{\overline{\Gamma}} T_{ij}^{I}(\overline{u}_j^{I}-\overline{u}_j^{II})d\Gamma + \Delta e_{u_i}, \quad (23)$$

where

$$\Delta e_{u_i} = \frac{1}{2}\left(\Delta\overline{u}_i^{I}+\Delta\overline{u}_i^{II}\right) - \int_{\overline{\Gamma}} U_{ij}^{I}(\Delta\overline{t}_j^{I}+\Delta\overline{t}_j^{II})d\Gamma + \text{CPV}\int_{\overline{\Gamma}} T_{ij}^{I}(\Delta\overline{u}_j^{I}-\Delta\overline{u}_j^{II})d\Gamma$$

can be regarded as the error at the iteration $(k-1)$. Assuming a priori that the error will decrease as the iteration goes on, the updated estimate of the interface boundary condition for u_i, for the current iteration k, is calculated using Eq. 23 as

$$u_i^{update}(\mathbf{y}) = u_i(\mathbf{y}) - \Delta e_{u_i}(\mathbf{y}) = \frac{1}{2}\left(\overline{u}_i^{I}(\mathbf{y})+\overline{u}_i^{II}(\mathbf{y})\right) - \int_{\overline{\Gamma}} U_{ij}^{I}(\mathbf{y},\mathbf{x})(\overline{t}_j^{I}(\mathbf{x})+\overline{t}_j^{II}(\mathbf{x}))d\Gamma(\mathbf{x}) + \text{CPV}\int_{\overline{\Gamma}} T_{ij}^{I}(\mathbf{y},\mathbf{x})(\overline{u}_j^{I}(\mathbf{x})-\overline{u}_j^{II}(\mathbf{x}))d\Gamma(\mathbf{x}). \quad (24)$$

3.2 *Estimate of the interface boundary condition for t_i*

The updating formula for t_i can be obtained in the same way as for u_i, except that the SBIEs are replaced by HBIEs. A point $\mathbf{y}$ on the interface $\overline{\Gamma}$ can be considered as (a) an interior point relative to the domain Ω, (b) a boundary point relative to the domain I, and (c) a boundary point relative to the domain II, resulting in the corresponding HBIEs as follows

$$\sigma_{ij}(\mathbf{y}) = \int_{\Gamma_1} Q_{ijk}t_k d\Gamma + \int_{\Gamma_2} Q_{ijk}t_k d\Gamma - \int_{\Gamma_1} P_{ijk}u_k d\Gamma - \int_{\Gamma_2} P_{ijk}u_k d\Gamma, \quad (25)$$

$$\frac{1}{2}\sigma_{ij}(\mathbf{y}) = \int_{\Gamma_1} Q_{ijk}t_k d\Gamma + \text{CPV}\int_{\overline{\Gamma}} Q_{ijk}^{I}t_k^{I} d\Gamma - \int_{\Gamma_1} P_{ijk}u_k d\Gamma - \text{HPV}\int_{\overline{\Gamma}} P_{ijk}^{I}u_k d\Gamma, \quad (26)$$

$$\frac{1}{2}\sigma_{ij}(\mathbf{y}) = \int_{\Gamma_2} Q_{ijk}t_k d\Gamma + \text{CPV}\int_{\overline{\Gamma}} Q^{II}_{ijk}t^{II}_k d\Gamma - \int_{\Gamma_2} P_{ijk}u_k d\Gamma - \text{HPV}\int_{\overline{\Gamma}} P^{II}_{ijk}u_k d\Gamma. \tag{27}$$

As in the case of velocity, the appropriate substitution of some of Eq. 13-Eq. 18 together with $\sigma_{ij} = \overline{\sigma}^I_{ij} + \Delta\overline{\sigma}^I_{ij}$ or $\sigma_{ij} = \overline{\sigma}^{II}_{ij} + \Delta\overline{\sigma}^{II}_{ij}$ into Eq. 25-Eq. 27, followed by manipulation, yields

$$\sigma_{ij} = \frac{1}{2}\left(\overline{\sigma}^I_{ij} + \Delta\overline{\sigma}^I_{ij}\right) - \text{CPV}\int_{\overline{\Gamma}} Q^I_{ijk}(\overline{t}^I_k + \Delta\overline{t}^I_k)d\Gamma + \text{HPV}\int_{\overline{\Gamma}} P^I_{ijk}(\overline{u}^I_j + \Delta\overline{u}^I_j)d\Gamma + \frac{1}{2}\left(\overline{\sigma}^{II}_{ij} + \Delta\overline{\sigma}^{II}_{ij}\right) - \text{CPV}\int_{\overline{\Gamma}} Q^{II}_{ijk}(\overline{t}^{II}_k + \Delta\overline{t}^{II}_k)d\Gamma + \text{HPV}\int_{\overline{\Gamma}} P^{II}_{ijk}(\overline{u}^{II}_k + \Delta\overline{u}^{II}_k)d\Gamma. \tag{28}$$

Since $Q^I_{ijk} = Q^{II}_{ijk}$ and $P^I_{ijk} = -P^{II}_{ijk}$, equation Eq. 28 can be rewritten as

$$\sigma_{ij} = \frac{1}{2}\left(\overline{\sigma}^I_{ij} + \overline{\sigma}^{II}_{ij}\right) - \text{CPV}\int_{\overline{\Gamma}} Q^I_{ijk}(\overline{t}^I_k + \overline{t}^{II}_k)d\Gamma + \text{HPV}\int_{\overline{\Gamma}} P^I_{ijk}(\overline{u}^I_k - \overline{u}^{II}_k)d\Gamma + \Delta e_{\sigma_{ij}}, \tag{29}$$

where

$$\Delta e_{\sigma_{ij}} = \frac{1}{2}\left(\Delta\overline{\sigma}^I_{ij} + \Delta\overline{\sigma}^{II}_{ij}\right) - \text{CPV}\int_{\overline{\Gamma}} Q^I_{ijk}(\Delta\overline{t}^I_k + \Delta\overline{t}^{II}_k)d\Gamma + \text{HPV}\int_{\overline{\Gamma}} P^I_{ijk}(\Delta\overline{u}^I_k - \Delta\overline{u}^{II}_k)d\Gamma$$

can be regarded as the error at the iteration $(k-1)$. Assuming a priori that the error will decrease as the iteration goes on, the updated estimate of the interface boundary condition for t_i, for the current iteration k, is calculated using Eq. 29 as

$$\sigma^{update}_{ij}(\mathbf{y}) = \sigma_{ij}(\mathbf{y}) - \Delta e_{\sigma_{ij}}(\mathbf{y}) = \frac{1}{2}\left(\overline{\sigma}^I_{ij}(\mathbf{y}) + \overline{\sigma}^{II}_{ij}(\mathbf{y})\right) - \text{CPV}\int_{\overline{\Gamma}} Q^I_{ijk}(\mathbf{y},\mathbf{x})(\overline{t}^I_k(\mathbf{x}) + \overline{t}^{II}_k(\mathbf{x}))d\Gamma + \text{HPV}\int_{\overline{\Gamma}} P^I_{ijk}(\mathbf{y},\mathbf{x})(\overline{u}^I_k(\mathbf{x}) - \overline{u}^{II}_k(\mathbf{x}))d\Gamma, \tag{30}$$

$$t^{update}_i(\mathbf{y}) = \sigma^{update}_{ij}(\mathbf{y})n_j(\mathbf{y}). \tag{31}$$

It can be seen that the first term on the RHS of Eq. 24 and Eq. 30 appears to be similar to those found in conventional methods. The new feature here lies in the remaining terms on the RHS, where the pre-convergence "imbalanced" tractions and velocities along the common interface are utilised in the interface integrals to update the boundary conditions.

3.3 Non-creeping flows

In the case of viscous flows with non-zero Reynolds number, the nonlinear convective terms introduce volume integrals into the IEs. However, in the process of deriving Eq. 10-Eq. 31, the volume integrals associated with the original domain and subdomains cancel each other and hence the boundary-only nature of the integral equations is preserved. Thus the formulas Eq. 24 and Eq. 31 can be used for both linear and nonlinear viscous flow problems.

3.4 Continuity of interface pressures

In practice, subdomain problems are solved separately and hence the pressure obtained could be arbitrarily discontinuous across an interface due to incompressibility. In general, the arbitrary pressure levels must be adjusted so that the pressure is continuous across interfaces. Although the pressure will be at the correct level and continuous when the iterative process converges, the adjustment is not an easy task during the iterative process because of either (a) the errors involved (i.e. it is not easy to separate the error from the arbitrary constant) or (b) the fact that pressure is not calculated explicitly (e.g. in the present velocity-traction formulation). Thus it is more convenient to enforce the pressure continuity condition by saying that the term $(\overline{t}^I_j + \overline{t}^{II}_j)$ in Eq. 24 and Eq. 31 is the sum of the tractions due to the extra stress $\mu(\partial u_i/\partial x_j + \partial u_j/\partial x_i)$ only, which can be calculated in a straightforward manner from the velocity field.

3.5 Numerical evaluation of integrals

The updating formulae Eq. 24 and Eq. 31 involve three kinds of singularity: (a) weakly singular integral (integrable in the ordinary Riemann sense); (b) strongly singular integral (Cauchy principal value sense) and (c) hyper-singular integral (Hadamard finite part sense).

The weakly singular integral can be treated numerically using the special log-weighted Gaussian integration or Telles's polynomial coordinate transformation [Telles (1987)], while the computation of strongly- and hyper-singular integrals can be accomplished by using finite parts of the integrals involved [Brebbia, Telles, and Wrobel (1984)]. Considerable attention has been devoted to the treatment of finite-part (f.p.) integrals in recent decades [Kutt (1975); Paget (1981); Ioakimidis (1995)]. A comprehensive review can be found in an article by Tanaka, Sladek, and Sladek (1994). By disregarding the divergent part, Kutt (1975) has derived some Gaussian quadrature rules of the form

$$\text{f.p.}\int_0^1 \frac{f(x)}{x^k}dx = \sum_{i=1}^{m} w_i f(x_i),$$

where $k \geq 1$, $f(0) \neq 0$, w_i the set of weights and x_i the set of abscissae. Note that these quadrature rules also permit the evaluation of one-dimensional Cauchy-type principal value integrals. This method, which is very convenient for computational purposes since mere scalar products of certain weights and function values have to be calculated, is applied to compute singular integrals in the present study.

The updating formulae Eq. 24 and Eq. 31 are applicable to a curved or straight interface. The latter is recommended here for practical uses because of the ease of implementation and a reduction of the computational cost associated with the updating process. When an interface is flat, the derivative of r with respect to a normal direction vanishes (i.e. $\partial r/\partial n = 0$) and hence the kernels T_{ij} and P_{ijk} reduce to zero and $\frac{\mu}{\pi r^2}\left(n_i\frac{r_j}{r}\frac{r_k}{r} + n_j\frac{r_i}{r}\frac{r_k}{r} + n_k\delta_{ij}\right)$ respectively. Furthermore, if such an interface is parallel to one of the coordinate axes, the updating formulas for u_i and t_i on that interface simply are

$$u_i^{update}(\mathbf{y}) = \frac{1}{2}\left(\overline{u}_i^I(\mathbf{y}) + \overline{u}_i^{II}(\mathbf{y})\right) - \frac{1}{4\pi\mu}\int_{\Gamma}\left(\frac{(r_i)}{r}\frac{(r_i)}{r} - \ln(r)\right)(\overline{t}_i^I(\mathbf{x}) + \overline{t}_i^{II}(\mathbf{x}))d\Gamma, \quad (32)$$

$$t_i^{update}(\mathbf{y}) = \frac{1}{2}\left(\overline{t}_i^I(\mathbf{y}) - \overline{t}_i^{II}(\mathbf{y})\right) + \text{HPV}\int_{\Gamma}\frac{\mu}{\pi r^2}(\overline{u}_i^I(\mathbf{x}) - \overline{u}_i^{II}(\mathbf{x}))d\Gamma, \quad (33)$$

where the strongly singular integrals disappear and the kernels in the remaining integrals are simplified. In contrast to conventional methods, in the present method, the pre-convergence imbalance of tractions on the interface is now taken into account for the update of velocity boundary conditions, and similarly, the pre-convergence imbalance of velocities are utilised in updating tractions. Furthermore, the present method exploits the use of the integral error (surface integrals) of velocities and tractions, rather than the simple point-wise error, to compute the update for each point under consideration.

The case of many subdomains can be treated straightforwardly by applying the above formulation for the case of two subdomains, i.e. Eq. 24 and Eq. 31, to each interface and the two relevant subdomains. The process of updating the boundary conditions at subdomain interfaces can also be proceeded in parallel owing to the independency of the associated input data.

3.6 Initial guess boundary conditions on interfaces

Due to the fact that the solving procedure is iterative, the initial interface solution has an influence on the convergence behaviour. For creeping flow problems, the initial solution is usually chosen to be zero. However, it is proposed here that the BIE formula for interior points can be used as a means to estimate the initial solution. By letting the nonlinear terms and the unknown boundary values of the original domain be zero (i.e. all boundary values are thus specified), the BIEs allow an estimate of the interior solution which is to be used as the initial interface solution. It is hypothesised that such an estimate is closer to the final solution than the usual assumption stated above. In the case of non-zero Reynolds number viscous flows, the initial boundary conditions at subdomain interfaces could be chosen among (a) zero-solution, (b) BIE-based values, (c) creeping flow solution and (d) lower Reynolds number flow solution. It is expected that convergence would be increasingly better for cases from (a) to (d), which will be tested in numerical examples. Fig. 2 summarises alternative methods of guessing an initial solution for the analysis of viscous flows.

3.7 Criterion for convergence

The iterative procedure converges when the solutions obtained from all subdomain problems match at the interfaces, i.e.

$$|\overline{u}_i^I - \overline{u}_i^{II}| < tol,$$

$$|\overline{t}_i^I + \overline{t}_i^{II}| < tol,$$

where tol is a preset tolerance. When the above criterion is met, the solution obtained is continuous everywhere and the governing equations are satisfied everywhere with the given actual boundary conditions. Thus the solution obtained is the required solution. When this happens, the solution, within the specified tolerance, is

$$u_i \approx \overline{u}_i^I \approx \overline{u}_i^{II},$$

$$t_i \approx \overline{t}_i^I \approx \overline{t}_i^{II},$$

$$\Delta e_{u_i}, \Delta e_{\sigma_{ij}} \to 0$$

It can be seen that on convergence, equations Eq. 24 and Eq. 31 are self-consistent.

3.8 Flow chart

The flow chart for the present procedure can be summarized as follows

1. Divide the domain of interest into a number of non-overlapping subdomains;
2. Initialize the boundary conditions at interfaces;
3. Form a boundary value problem for each subdomain by taking either the velocity boundary conditions or the traction boundary conditions from the set of artificial boundary conditions at its relevant interfaces;
4. Solve subdomain problems. Since they are boundary value problems, any numerical method such as FDM, FEM, BEM or FVM can be applied to solve subdomain problems for a numerical solution;
5. Check for convergence;
6. If not yet converged, update the artificial boundary conditions on all interfaces using BIE-based formulae Eq. 24 and Eq. 31 and then repeat from step 3;
7. If converged, stop.

4 Numerical examples

The present DD method is verified through the simulation of Poiseuille, driven cavity and backward facing step viscous flows, where subdomain problems are solved by the BEM using continuous elements in general and discontinuous elements adjacent to corners. Nodes at a corner are shifted into the associated elements by an amount of one quarter of the length of the corner elements [Tran-Cong and Phan-Thien (1988)]. This shifting allows the correct description of multi-valued tractions at corners and also the updating process to be confined to smooth interface points only, i.e. the formulae Eq. 24 and Eq. 31 above are sufficient here for the updating mechanism. The integral formulations are applied to relatively small subdomains instead of to the whole domain, resulting in a great reduction of the cost associated with the BEM, especially for the computation of volume integrals arising from the pseudo body forces term in the analysis of non-linear viscous flow problems. The convergence measure (*CM*) employed here is the norm of relative error of velocity between two adjacent subdomains at the interfaces as follows

$$CM = \sqrt{\frac{\sum_{interface}\sum_{i=1}^{N}\left[(u_1^{(i)I}-u_1^{(i)II})^2+(u_2^{(i)I}-u_2^{(i)II})^2\right]}{\sum_{interface}\sum_{i=1}^{N}\left[(u_1^{(i)I})^2+(u_2^{(i)I})^2\right]}}, \tag{34}$$

where $interface$ denotes subdomain interfaces, (i) the nodes on the $interface$ and N the number of nodes on the $interface$. Generally, it is necessary to use a relaxation scheme within the iterative process to achieve convergence. The resultant interface velocity and traction conditions at each iteration are calculated using the ones obtained in the previous iteration and the currently computed interface conditions from Eq. 24 and Eq. 31 as follows

$$u_i^k = (1-\alpha)u_i^{k-1} + \alpha u_i^k, \tag{35}$$

$$t_i^k = (1-\alpha)t_i^{k-1} + \alpha t_i^k, \tag{36}$$

where k denotes the current iteration and $0 < \alpha \leq 1$ is the relaxation parameter. The case $\alpha = 1$ means that no relaxation is applied.

4.1 Poiseuille flow

Consider the Poiseuille creeping flow problem on a 1×1 domain. Owing to the symmetry, only one half of the domain (1×0.5) needs to be modelled. The domain of analysis is divided into two subdomains with the same size of 0.5×0.5 (Fig. 3). A 6×6 mesh is employed for each subdomain. In order to assess the effect of the use of interface BIs, a simplified version of Eq. 24 and Eq. 31 (i.e. without the BIs) are also used for comparison. In that case, the updating mechanism is somewhat similar to that of conventional methods. The results obtained for the two cases using zero initial solution are displayed in Fig. 4 which shows that the interface BIs make the convergence more stable and faster. Furthermore, the effect of initial solution on convergence is also

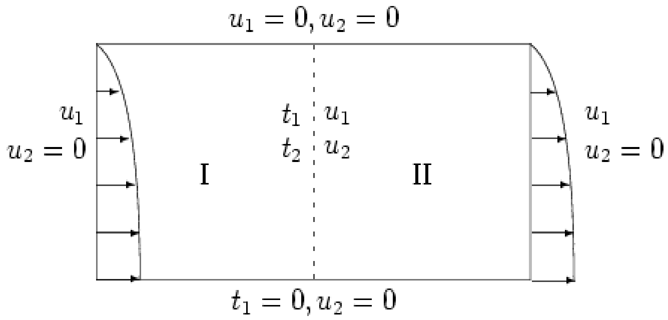

Figure 3 : Poiseuille flow, 2 subdomains: geometry and boundary conditions.

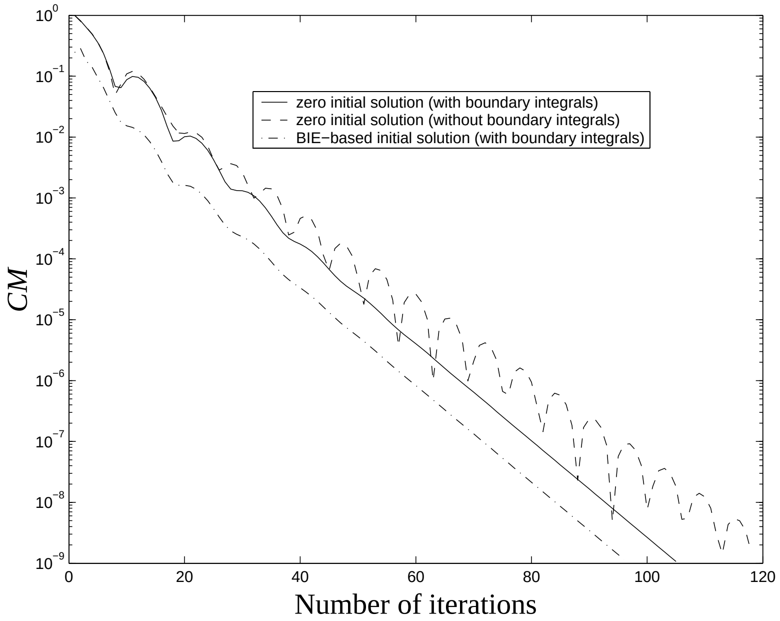

Figure 4 : Creeping Poiseuille flow, 2 subdomains, density of 6×6 per subdomain, relaxation factor $\alpha = 0.3$: comparison of convergence behaviour. The initial interface boundary conditions are assumed either to be zero or BIE-based estimates. Subsequent estimates of the interface boundary conditions are calculated using the updating formulas with BIs in one case and without BIs in the other case.

examined. As shown in Fig. 4, the use of BIE-based initial solution yields a better result than that obtained with zero initial solution. In all cases, convergence rates obtained are high and the value of *CM* achieved is very small ($\mathrm{O}(10^{-9})$). However, the procedure can usually be considered as convergent when the *CM* is of the order of $\mathrm{O}(10^{-4})$ which corresponds to about 40 iterations for the present method and about 50 iterations for the conventional method. It can be seen that the EFFICIENCY factor defined as the ratio of the CPU time of the CONVENTIONAL method to that of the PRESENT method is about 1.25. The effect of the relaxation factor on the convergence rate is displayed in Fig. 5. As expected, a low relaxation factor makes the convergence smooth but also slow. The linear interpolation used to compute interface integrals gives a slightly better convergence than the constant interpolation as shown in Fig. 6.

4.2 *Driven cavity viscous flow*

The driven cavity viscous flow is recognised as a benchmark problem for checking and validating numerical methods in the field of CFD and is utilised here for the same reason.

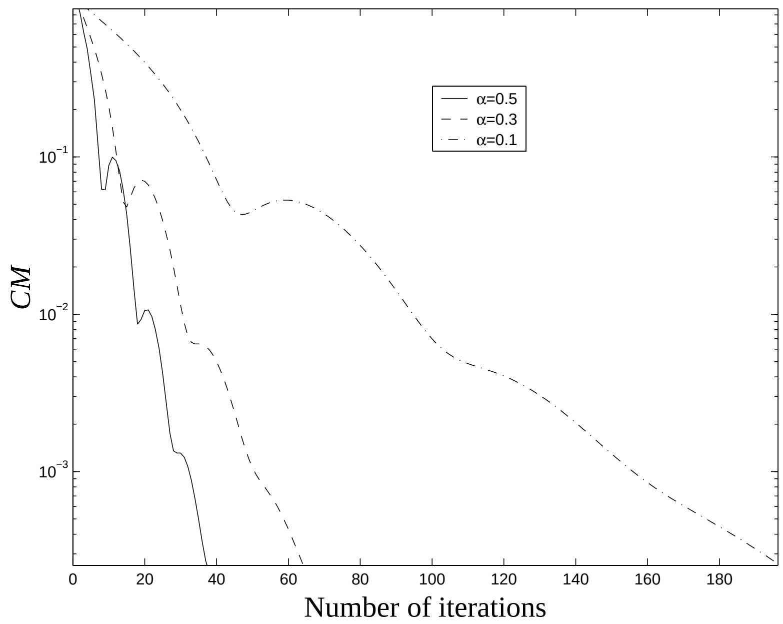

Figure 5 : Creeping Poiseuille flow, 2 subdomains, density of 6 × 6 per subdomain: effect of the relaxation factor α on the convergence behaviour. The convergence is observed to be smooth but slow with a reduction in α.

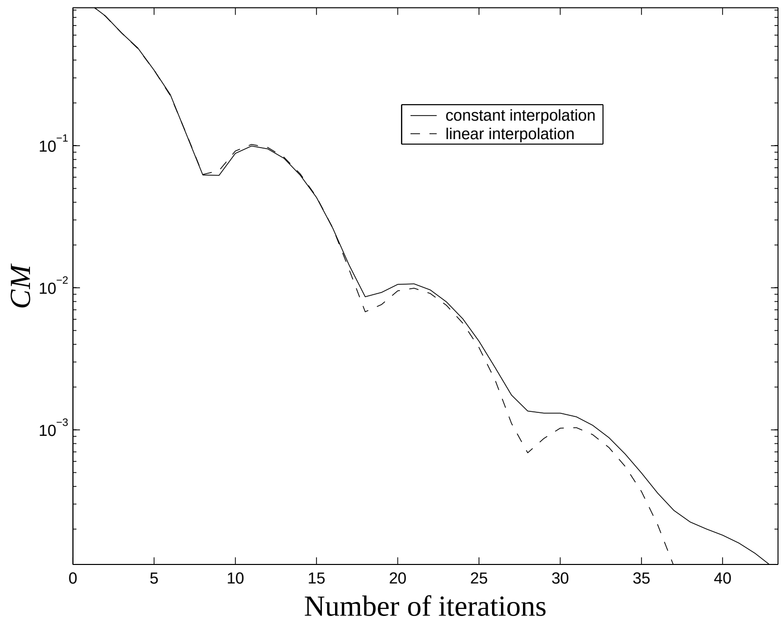

Figure 6 : Creeping Poiseuille flow, 2 subdomains, density of 6 × 6 per subdomain: effect of the interpolation type used in the computation of interface integrals on the convergence behaviour. The linear interpolation yields a slightly better convergence behaviour than the constant one.

4.2.1 Two subdomains - creeping viscous flow

The domain of dimension 1 × 1 is partitioned into two subdomains with the same size 0.5 × 1 (Fig. 7). A 6 × 11 mesh is employed for each subdomain. The effect of interface integrals on convergence rate is further verified in this problem. Fig. 8 shows that the interface integrals

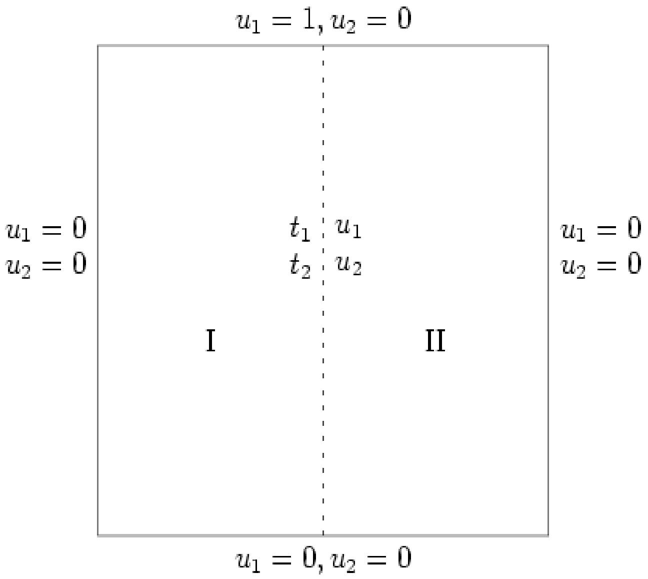

Figure 7 : Cavity driven flow, 2 subdomains: geometry and boundary conditions.

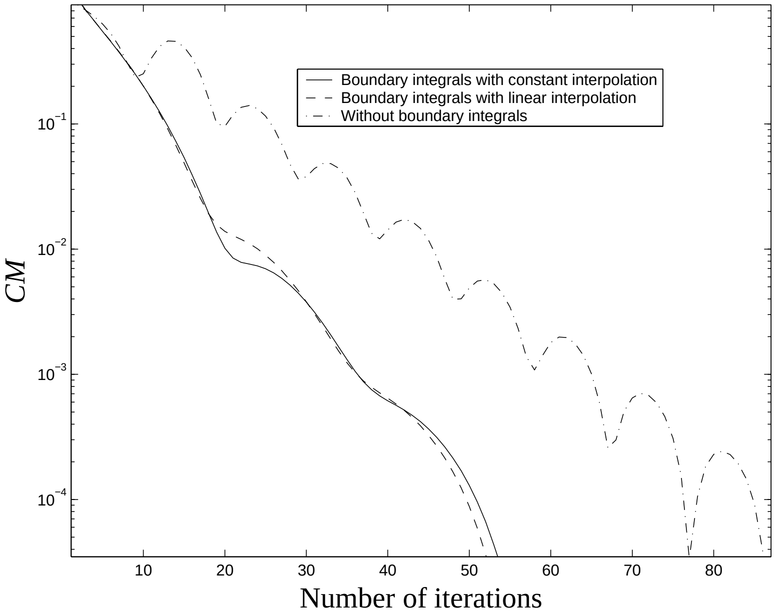

Figure 8 : Creeping driven cavity flow, 2 subdomains, density of 6×11 per subdomain, relaxation factor $\alpha = 0.3$: convergence behaviour. This figure further demonstrates the effectiveness of interface integrals in the updating process. Constant and linear interpolations yield similar performances.

play an important role in the updating process as they make the convergence much smoother and faster. With the preset tolerance of $1.e-4$, it takes about 50 and 75 iterations to OBTAIN CONVERGENCE for the present and conventional methods respectively. In this case, the EFFICIENCY factor is about 1.5. Interface integrals computed using constant and linear interpolations yield similar convergence behaviours.

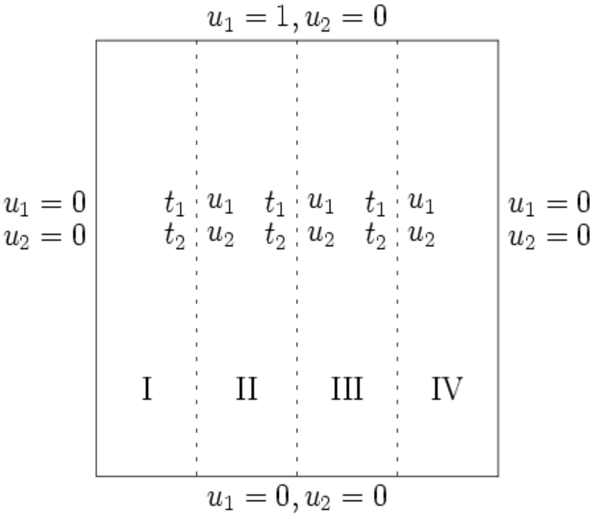

Figure 9 : Cavity driven flow, 4 subdomains: geometry and boundary conditions.

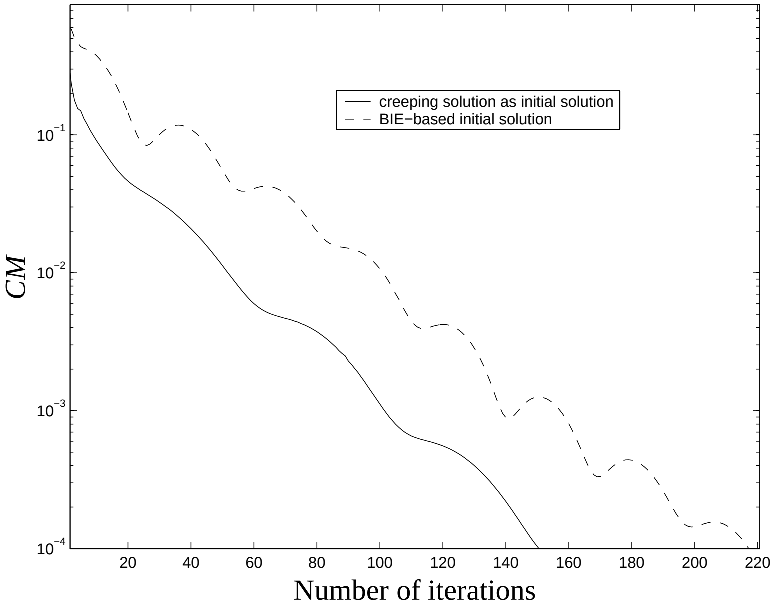

Figure 10 : Driven cavity flow, 4 subdomains, *Re* $= 100$, density of 6×21 per subdomain, relaxation factor $\alpha = 0.3$: convergence behaviour. The use of the creeping flow solution as an initial solution yields a better convergence than the use of BIE-based initial solution, which confirms the expectation that an initial solution closer to the final solution to be found can yield a faster convergence.

4.2.2 Four subdomains - viscous flow

The computational domain here is partitioned into 4 subdomains with the same size 0.25×1 (Fig. 9), where the two middle subdomains each involve two interfaces. A mesh of 6×21 is employed for each subdomain. The flow with *Re* $= 100$ is simulated with different initial solutions used. In the case of non-zero Reynolds number, the use of the creeping flow solution (*Re* $= 0$) as the initial interface solution leads to better convergence characteristics than the use of BIE-based initial solution directly at that Reynolds number (Fig. 10). This result is not surprising because the creeping flow solution is closer to the

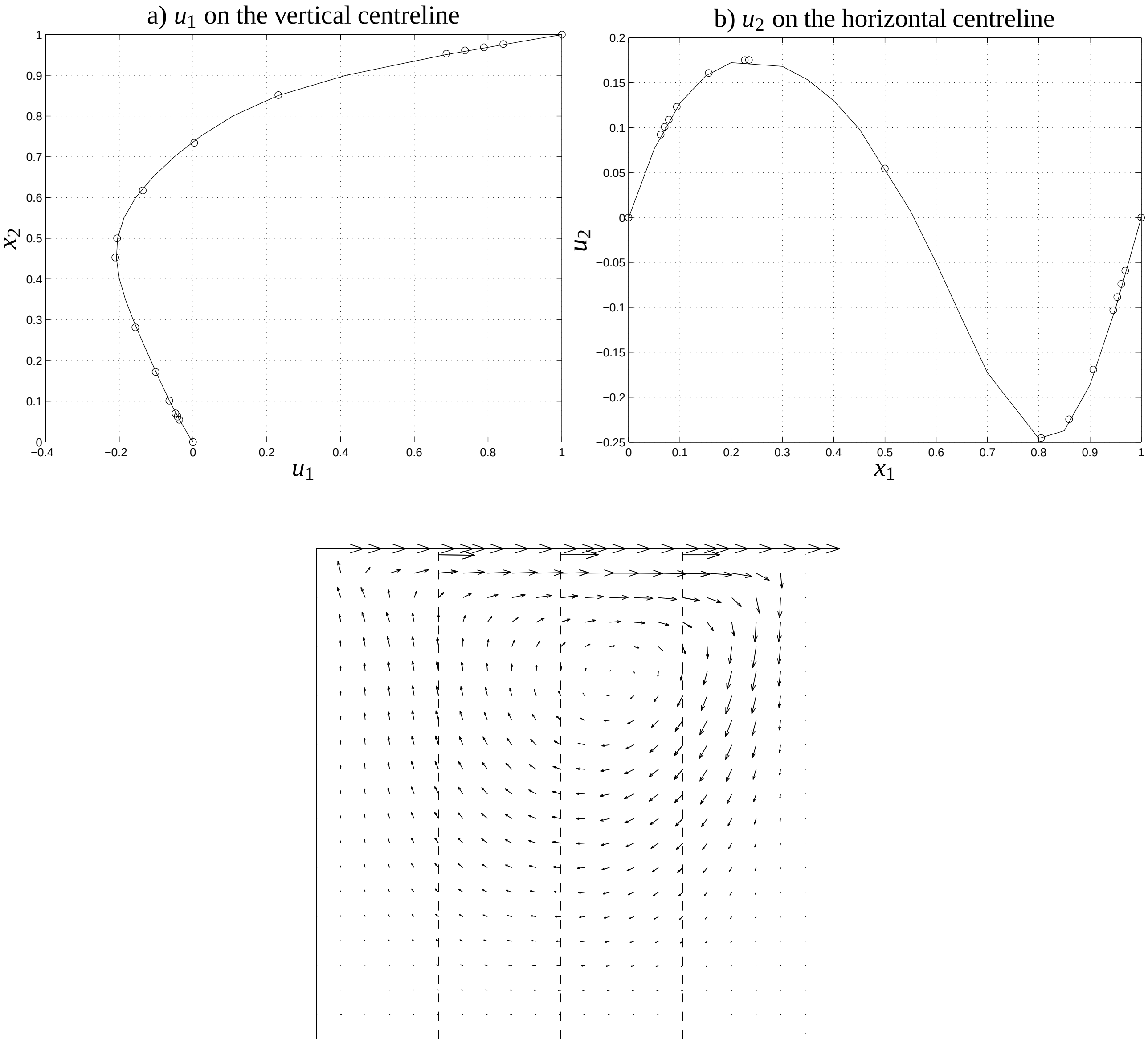

Figure 11 : Driven cavity flow, 4 subdomains, $Re = 100$: comparison of velocity profiles on the vertical and horizontal centrelines denoted by solid lines with the benchmark solution obtained by Ghia, Ghia, and Shin (1982) denoted by $\circ$. The velocity vector field obtained is also displayed.

final solution to be found than the BIE-based initial solution. The results obtained, including the velocity profiles along centrelines and the velocity vector field, are displayed in Fig. 11 which are in very good agreement with the benchmark solution of Ghia, Ghia, and Shin (1982).

4.3 Backward facing step viscous flow

A steady viscous incompressible flow over a backward facing step provides an excellent test case and has been addressed by numerous authors using a wide variety of numerical methods [e.g. Kim and Moin (1985); Thompson and Ferziger (1989); Gartling (1990); Verma and Eswaran (1999)]. The experimental results have been given by Armaly, Durst, Pereira, and Schonung (1983). The geometry and boundary conditions are shown in Fig. 12 where the size of the rectangular computational domain is chosen to be the same as that proposed by Verma and Eswaran (1999). The step height is one half of the channel height H. No-slip condition is imposed on all solid surfaces. At the inlet, the velocity field is

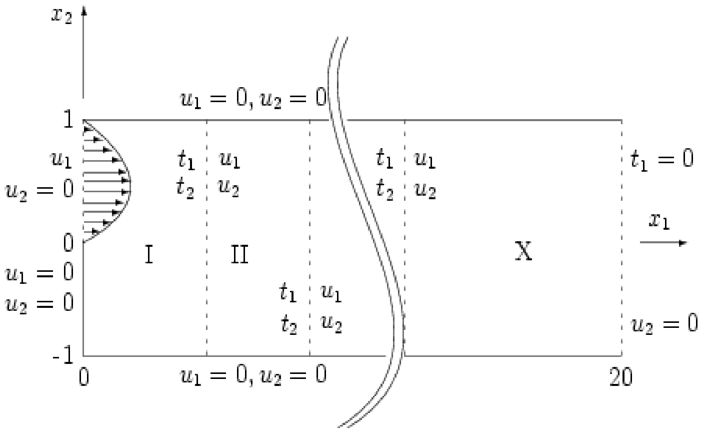

Figure 12 : Backward facing step flow, 10 subdomains: geometry and boundary conditions.

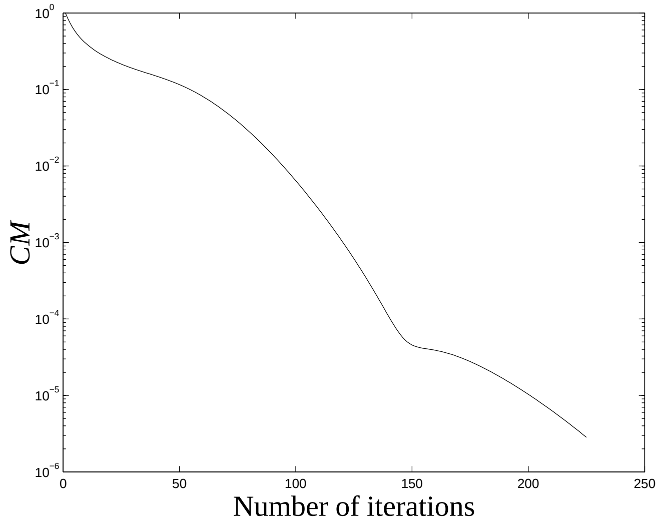

Figure 13 : Creeping flow over backward facing step, 10 subdomains, density of 11×11 per subdomain, relaxation factor $\alpha = 0.3$: convergence behaviour.

specified as a parallel flow with a parabolic horizontal component given by $u_1 = 6x_2(1-x_2)$ for $0 \leq x_2 \leq 1$ and a vertical component $u_2 = 0$. This produces a maximum inflow velocity of $u_{1max} = 1.5$ and an average inflow velocity of $\overline{u}_1 = 1.0$. At the outlet, the flow is assumed to be a parallel flow ($u_2 = 0$) and a zero-traction normal to the boundary. It can be seen that the computational domain is slender and hence a subdivision along the flow direction is used to increase computational efficiency. The domain here is divided into 10 subdomains (Fig. 12) that have the same dimension of 2×2 and the same uniform mesh of 11×11. The use of multiple subdomains (10 in this case) would provide a strong test for the present DD method, which is verified with both creeping flow and non-zero Reynolds number flows in the following sections.

4.3.1 Creeping flow

For creeping flows, there is no nonlinear term in the governing equations, and hence the solution for each subdomain can be obtained without the use of iteration. The iterative process is employed here to update the boundary conditions at subdomain interfaces and a high convergence rate is obtained as shown in Fig. 13. The *CM* achieved is very small (at least $O(10^{-6})$). The final result is displayed through a plot of the velocity profile at the outlet (Fig. 14), which shows that the computed velocity profile agrees very well with the exact one.

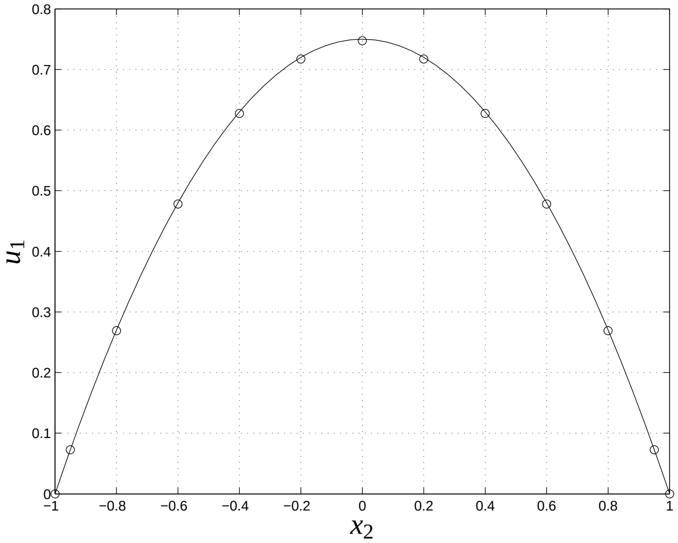

Figure 14 : Creeping flow over backward facing step, 10 subdomains, density of 11×11 per subdomain, relaxation factor $\alpha = 0.3$: Comparison of the computed x_1 component of the velocity profile at the outlet denoted by $\circ$ and the exact solution denoted by solid line, which shows a very good agreement.

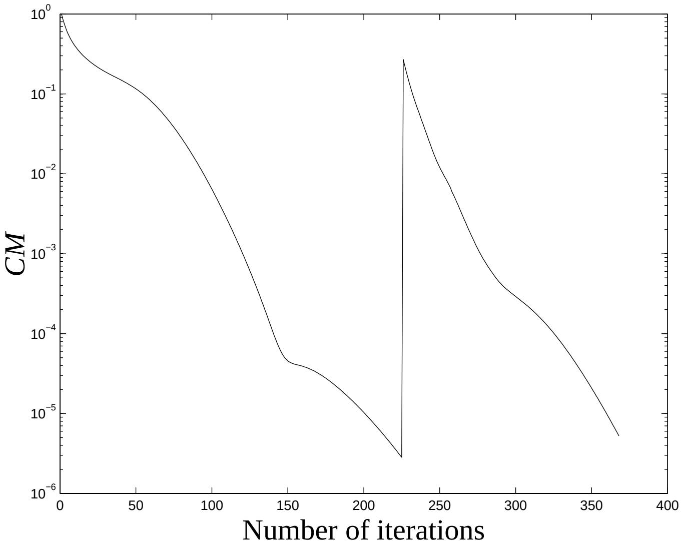

Figure 15 : Flow over backward facing step, $Re = 133$, 10 subdomains, density of 11×11 per subdomain, relaxation factor $\alpha = 0.3$, creeping flow initial solution: convergence behaviour. The first stage of the curve corresponds to the creeping flow, while the second stage to the flow with $Re = 133$. Although the creeping flow stage took approximately 225 iterations, each iteration is extremely fast in comparison with an iteration during the second stage. Furthermore, the first stage can easily be stopped at $CM = 10^{-3}$ (say), or about 125 iterations, without affecting the performance of the second stage.

4.3.2 Viscous flow with non-zero Reynolds number

The Reynolds number is defined as $R_e = \frac{\overline{u}_1 H}{\nu}$. The numerical results obtained for the flows at $Re = 60$ and $Re = 133$ are displayed in Fig. 15-Fig. 17, where the vortex appearance is observed. The predicted reattachment length of 4 at $Re = 133$ is in good agreement with the values reported by Thompson and Ferziger (1989) and

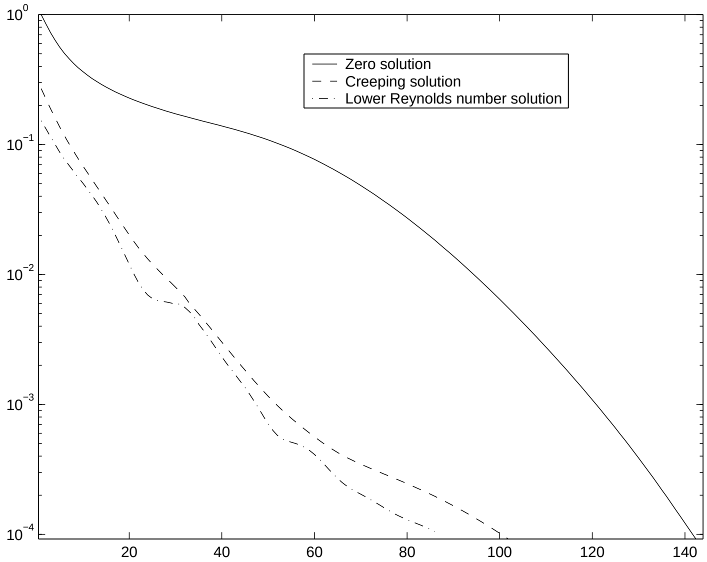

Figure 16 : Flow over backward facing step, $Re = 133$, 10 subdomains, density of 11×11 per subdomain, relaxation factor $\alpha = 0.3$: comparison of the convergence behaviour between different cases of initial solution. The use of a lower Reynolds number flow solution (e.g. Re=60 here) as an initial solution clearly results in the best convergence here.

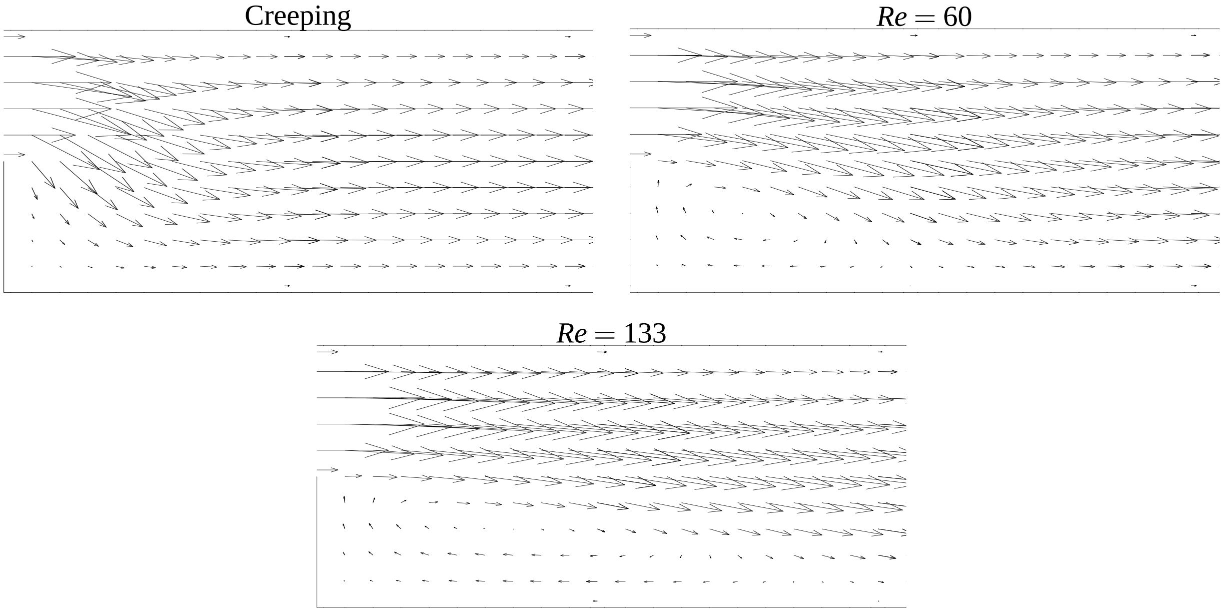

Figure 17 : Flow over backward facing step, 10 subdomains, density of 11×11 per subdomain, relaxation factor $\alpha = 0.3$: Enlargement at the upstream section shows the growing of a recirculation behind the step with increasing Reynolds number.

Verma and Eswaran (1999). Fig. 16 shows that the use of a lower Reynold number flow solution as the initial solution yields a convergence rate much higher than the use of the zero initial solution and slightly better than the use of the creeping flow initial solution. A comparison of the flow patterns for zero and non-zero Reynolds numbers is displayed in Fig. 17 which reveals more clearly the recirculation flow behind the step in the case of non-zero

Reynolds number flows.

5 Concluding Remarks

This paper reports a new BIE-based domain decomposition method for the analysis of viscous flow problems. The idea of the proposed iterative method is to utilise the governing integral equations as a means to update the interface boundary conditions. Formulas for updating interface boundary conditions are derived, where the computations are confined to subdomain interfaces only. Numerical results showed that the proposed method performs well for both linear and non-linear viscous flow problems. The initial interface solution based on BIEs is able to give a faster convergence rate than the zero-initial solution in the analysis of creeping viscous flows, while the use of the creeping flow or lower Reynolds number flow as an initial interface solution provides better convergence characteristics in the case of non-zero Reynolds number flows.

Acknowledgement: The authors would like to thank the referees for their helpful comments.

References

Agnantiaris, J.; Polyzos, D. (2003): A boundary element method for acoustic scattering from non-axisymmetric and axisymmetric elastic shells. *CMES: Computer Modeling in Engineering & Sciences*, vol. 4, no. 1 pp. 197–212.

Armaly, B.; Durst, F.; Pereira, J.; Schonung, B. (1983): Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics*, vol. 127, pp. 473–496.

Brebbia, C.; Dominguez, J. (1992): *Boundary Element An Introductory Course*. Computational Mechanics Publications, Southampton.

Brebbia, C.; Telles, J.; Wrobel, L. (1984): *Boundary Element Techniques Theory and Applications in Engineering*. Springer-Velag, Berlin.

Gartling, D. (1990): A test problem for outflow boundary conditions-flow over a backward-facing step. *International Journal for Numerical Methods in Fluids*, vol. 11, pp. 953–967.

Ghia, U.; Ghia, K.; Shin, C. (1982): High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, vol. 48, pp. 387–411.

Han, Z.; Atluri, S. (2002): SGBEM (for Cracked Local Subdomain) – FEM (for uncracked global Structure) Alternating Method for Analyzing 3D Surface Cracks and Their Fatigue-Growth. *CMES: Computer Modeling in Engineering & Sciences*, vol. 3, pp. 699–716.

Han, Z.; Atluri, S. (2003): On Simple Formulations of Weakly-Singular Traction & Displacement BIE, and Their Solutions through Petrov-Galerkin Approaches,. *CMES: Computer Modeling in Engineering & Sciences*, vol. 4, no. 1 pp. 5–20.

Ioakimidis, N. (1995): Remarks on the Gauss quadrature rule for a particular class of finite part integrals. *International Journal for Numerical Methods in Engineering*, vol. 38, pp. 2433–2448.

Kim, J.; Moin, P. (1985): Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, vol. 59, pp. 308–323.

Kutt, H. (1975): Quadrature Formulae for Finite Part Integrals. Technical Report CSIR Special Report WISK 178, National Research Institute for Mathematical Sciences, Pretoria, 1975.

Mai-Duy, N.; Tran-Cong, T. (2002): Mesh-free radial basis function network methods with domain decomposition for approximation of functions and numerical solution of Poisson's equations. *Engineering Analysis with Boundary Elements*, vol. 14, pp. 185–199.

Paget, D. (1981): A quadrature rule for finite part integrals. *BIT*, vol. 21, pp. 212–220.

Quarteroni, A.; Valli, A. (1999): *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, New York.

Tanaka, M.; Sladek, V.; Sladek, J. (1994): Regularization techniques applied to boundary element methods. *Applied Mechanics Reviews*, vol. 47, pp. 457–499.

Telles, J. (1987): A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 959–973.

Thompson, M.; Ferziger, J. (1989): An adaptive multigrid technique for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, vol. 82, pp. 94–121.

Tran-Cong, T.; Phan-Thien, N. (1988): Three-dimensional study of extrusion processes by Boundary Element Method. Part 1: an implementation of high order elements and some Newtonian results. *Rheol. Acta*, vol. 27, pp. 21–30.

Tsai, C.; Young, D.; Cheng, A.-D. (2002): Meshless BEM for three-dimensional Stokes flows. *CMES: Computer Modeling in Engineering & Sciences*, vol. 3, pp. 117–128.

Verma, A.; Eswaran, V. (1999): An overlapping control volume method for the Navier-Stokes equations on non-staggered grids. *International Journal for Numerical Methods in Fluids*, vol. 30, pp. 279–308.

Zhang, C.; Savaidis, A. (2003): 3-D transient dynamic crack analysis by a novel time-domain BEM. *CMES: Computer Modeling in Engineering & Sciences*, vol. 4, pp. 603–618.