Transactions on Modelling and Simulation vol 9, © 1995 WIT Press, www.witpress.com, ISSN 1743-355X

Domain decomposition methods for vorticity transport equation in boundary domain integral method

M. Hriberšek, L. Škerget
Faculty of Mechanical Engineering, University of Maribor, Smetanova 17, 62000 Maribor, Slovenia

Abstract

The paper deals with the use of domain decomposition methods in Boundary Domain Integral Method (BDIM) for Navier-Stokes equations of laminar viscous incompressible fluid flow. Since the vorticity transport equation presents the major problem in solving BDIM set of equations, domain decomposition methods are used for its solution. Multiplicative and additive Schwarz iterations are implemented. Partial subdomain problems are computed by the use of Krylov subspace type iterative solvers. Overlapping and non-overlapping subdomain divisions are presented and their effects on convergence of domain decomposition iterative procedures are reported. The new iterative schemes are tested on Poiseuille's flow and Backward facing step flow for various Re number values.

1 Basic theory of Domain Decomposition Method

Domain Decomposition Method (DDM) is a relatively old method (Schwarz 1870) and is becoming increasingly popular in modern approximation methods due to high parallelisation capabilities. Similarly as Subdomain technique in BEM it divides the original computational domain into several overlapping or non-overlapping subdomains, which now present partial problems, from which solution over the original domain can be found. The main feature of Domain Decomposition Method is a fact, that solving partial problems is only one step towards the solution of the original system. This follows from a fact, that values of functions and its derivatives on the subdomain interface are not known at the beginning of computation. Through an iterative procedure it is then possible to find the final solution with some combination of calculated values from subdomains.

Let us look at the basic steps of the method (Hackbusch[5]). Let the original domain Ω be divided into several subdomains Ω_κ: $\Omega = \sum_{\kappa=1}^{n_s} \Omega_\kappa$ with n_s number of subdomains, x soultion of original system $Ax = b$, n number of nodes in original domain and x_κ solution of partial problem in Ω_κ. Then, solution x can be obtained by some combination of x_κ. Partial problems are defined as

$$A_\kappa x_\kappa = b_\kappa, \tag{1}$$

where A_κ represents a quadratic matrix with dimensions $n_\kappa \times n_\kappa$, with n_κ being number of nodes in subdomain κ. In BDIM this matrix is different from BDIM with Subdomain technique, where a subdomain matrix is rectangular and therefore impossible to solve.

To construct an algorithm, which connects partial solutions, we need to define a mapping p_κ, which maps a group of vectors of partial solutions into a group of vectors in the original system:

$$p_\kappa : X_\kappa \rightarrow X, \tag{2}$$

where X also incorporates the original solution x. The latter can be obtained by a combination of partial solutions

$$x = \sum_\kappa p_\kappa x^\kappa. \tag{3}$$

Since one solution of partial problems is not enough to obtain original solution, we have to map the intermediate solution x back into the subdomains. This can be accomplished by a restriction r_κ:

$$r_\kappa = A_\kappa^{-1} p_\kappa^T \quad ; X \rightarrow X_\kappa \tag{4}$$

what gives us projection P_κ for one iteration of DDM:

$$P_\kappa = p_\kappa r_\kappa A \quad ; X \rightarrow X. \tag{5}$$

Together with projection P we have to define the iteration step Φ_κ

$$\Phi_\kappa(x, b) = x - p_\kappa r_\kappa (Ax - b). \tag{6}$$

A natural choice for mappings p_κ and r_κ is the Dirac function, what excludes any averaging or interpolation in data transfer between global and local meshes. This approach is used in this paper for BDIM.

Solution of the original problem with help of partial solutions from subdomains $\kappa = 1, 2, ...n_s$ is possible in two different ways:

a) Multiplicative Schwarz Iteration (MSI):

$$\Phi^{MSI} = \Phi_\kappa \circ \Phi_{\kappa-1} ... \circ \Phi_2 \circ \Phi_1 \tag{7}$$

Feature of MSI is its sequential nature since a part of the solution of the first partial problem is immediately transferred to the neighboring subdomain, the partial solution in this subdomain then to its neighboring subdomain etc.. MSI is therefore more appropriate for implementing on scalar and vector computers, while it is only partially paralelizable.

b) Additive Schwarz Iteration (ASI):

$$\Phi^{ASI} = x - \Theta_{ASI} \sum_{\kappa-1} p_\kappa r_\kappa (Ax - b) \tag{8}$$

Feature of ASI is its parallel nature as all the partial problems can be computed in parallel. When partial solutions are obtained, a communication step follows, where values of functions and its derivatives are interchanged between subdomains. In order to control the iterative procedure, underrelaxation parameter Θ_{ASI} is introduced.

2 Boundary -Domain Integral method

Boundary-Domain Integral method (BDIM) (Škerget[1], Alujevič et al.[2]) is another member of Boundary Element Methods. It was developed for numerical solution of Navier-Stokes equations. In this paper, planar laminar time dependent flow of viscous incompressible fluid will be considered. BDIM uses velocity-vorticity formulation of Navier-Stokes equations, where flow kinematics is numerically separated from computation of flow kinetics (vorticity and energy transport). The final system of discretised equations reads as (for derivation see Škerget[1]):

- flow kinematics (normal form):

$$\left([H_1^E] + [H_{t1}^E]\right)\{v_n\} = \left([H_{t2}^E] - [H_2^E]\right)\{v_t\} + [D_t^E]\{\omega\}, \tag{9}$$

- flow kinematics (tangential form):

$$\left([H_1^E] + [H_{t1}^E]\right)\{v_t\} = \left([H_2^E] - [H_{t2}^E]\right)\{v_n\} + [D_n^E]\{\omega\}, \tag{10}$$

- vorticity transport:

$$\begin{aligned}[H^P]\{\omega\}_F &= [G^P]\left\{\frac{\partial\omega}{\partial n}\right\}_F - \frac{1}{\nu}[G^P]\{v_n\omega\}_F + \frac{1}{\nu}[D_x^P]\{v_x\omega\}_F + \\ &\quad \frac{1}{\nu}[D_y^P]\{v_y\omega\}_F + \frac{1}{\nu}[B]\{\omega\}_{F-1}, \end{aligned} \tag{11}$$

Transactions on Modelling and Simulation vol 9, © 1995 WIT Press, www.witpress.com, ISSN 1743-355X

- energy transport:

$$[H^P]\{T\}_F = [G^P]\left\{\frac{\partial T}{\partial n}\right\}_F - \frac{1}{\alpha}[G^P]\{v_n T\}_F + \frac{1}{\alpha}[D_x^P]\{v_x T\}_F + \frac{1}{\alpha}[D_y^P]\{v_y T\}_F + \frac{1}{\alpha}[B]\{T\}_{F-1}, \tag{12}$$

where $F-1$ denotes previous time step values. By applying appropriate boundary conditions equations (9) + (10), (11) and (12) are transferred into well known forms of systems of linear equations:

$$A_{km}\ \ x_{km} = b_{km}, \tag{13}$$

$$A_w\ \ x_w = b_w, \tag{14}$$

$$A_T\ \ x_T = b_T, \tag{15}$$

where subscript km denotes flow kinematics, w flow kinetic and T energy equation. They are coupled througt vorticity and velocity values into strong nonlinear system of equations, which can be solved by under-relaxation technique. Subdomain division can successfully be applied to equations (11) and (12), whereas for flow kinematics segmentation technique must be used (Hriberšek[3]). From this point of view DDM can be applied to equations (11) and (12), leaving flow kinematics out. Since $[A_{km}]$ is much smaller than other two system matrices this does not seriously affect the performance of BDIM. Due to similarity of equations (11) and (12) only domain decomposition for vorticity transport will be discussed.

3 Domain decomposition

The original domain can be decomposed in two different ways:

1. $\sum_\kappa n_k = n$ simple (non-overlapping) division, where subdomains are linked through the common interface boundaries; only one way of expressing x is possible (DDM).

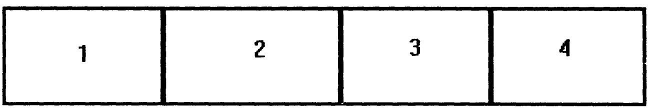

Figure 1: Non-overlapping division of the original domain

2. $\sum_\kappa n_k > n$ overlapping division, several ways of composing x are possible (DDM-O).

In further text DDM-O will stand for overlapping subdomains and DDM for non-overlapping subdomains.

Transactions on Modelling and Simulation vol 9, © 1995 WIT Press, www.witpress.com, ISSN 1743-355X

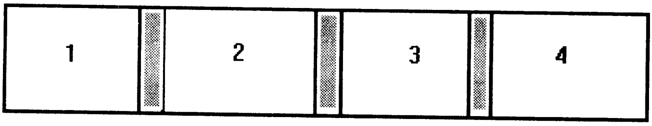

Figure 2: Overlapping division of the original domain

3.1 Decomposition without subdomain overlapping

Decomposition without subdomain overlapping is very similar to Subdomain technique in BEM, as subdomains are connected through the common (interface) boundaries. In the iterative procedure of solving one outer iteration of BDIM partial problems are solved separately, their mutual connection being compatibility conditions at the interface boundaries (Table 1). Contrary to Subdomain technique here exists only one unknown at each interface node (for example function), while the other value (derivative of funcion, which is also unknown in Subdomain technique) is known from the neighboring subdomain. In our case the function can be vorticity or temperature (here vorticity has been chosen).

Table 1: DDM and unknowns in neighboring subdomains

	Subdomain i	Subdomain j
unknown	$\omega_{\Gamma_{ij}}$	$\frac{\partial \omega}{\partial n}_{\Gamma_{ji}}$
known value	$\frac{\partial \omega}{\partial n}_{\Gamma_{ij}} = -\frac{\partial \omega}{\partial n}_{\Gamma_{ji}}$	$\omega_{\Gamma_{ji}} = \omega_{\Gamma_{ij}}$

3.2 Decomposition with subdomain overlapping

Subdomain division by overlapping of subdomains is a classical approach to DDM. The extent of overlapping is a parameter, which influences the speed of DDM-O convergence. Larger overlapping areas usually produce more accurate values of function derivatives in other approximation methods (FEM, FDM), but such DDM iterations are much more computationally expensive.

In Boundary Element Methods function values and normal derivatives on the boundary are known after the solution at the boundary. Values of functions and its derivatives in the interior of the domain can then be explicitly computed from the known boundary values. In BDIM (vorticity equation) we already have values of vorticity in the domain (implicit system of equations!) after the system solution, but values of vorticity derivatives in the interior of the domain are still unknown. To compute these values two possibilities exist:

Transactions on Modelling and Simulation vol 9, © 1995 WIT Press, www.witpress.com, ISSN 1743-355X

1. Computation by means of interpolation functions: simple implementation, does not use too much computation time, quality of results can be questionable.

2. Computation by derivation of integral equations of vorticity transport: excellent accuracy of derivatives with an appropriate integration scheme (due to high order of singularity), very time consuming procedure, additional computer memory is needed.

Due to testing purposes only the first approach was selected and used in this computation, especially because of lower computation time and memory demands.

In table 2 distribution of unknowns in subdomains is presented. Comparing with DDM the difference is now in the value of normal derivative of vorticity on the boundary of subdomain i, which is needed to construct the right-hand side of subdomain i. This derivative can be computed with the known vorticity gradient in subdomain j (computed by derivation of interpolation functions) and normal vector on the boundary of subdomain i.

Table 2: DDM-O and unknowns in neighboring subdomains

	Subdomain i	Subdomain j
unknown	$\omega_{\Gamma_{ij}}$	$\frac{\partial \omega}{\partial n}_{\Gamma_{ji}}$
known value	$\frac{\partial \omega}{\partial n}_{\Gamma_{ij}} = - \boldsymbol{\nabla} \omega_{\Omega_j} \, \boldsymbol{n}_{ij}$	$\omega_{\Gamma_{ji}} = \omega_{\Omega_i}$

4 Test examples

Two test examples were computed to validate the applicability of DDM for vorticity transport equation. The first test presents viscous flow in a narrow channel with parabolic inlet velocity profile and fully developed outflow. Computational domain was discretised with 5 subdomains, 48 boundary and 77 internal nodes. The second test was the well known backward facing step flow (BFS). Geometry, which is the same as with test 1, and boundary conditions are presented in Figure 3. Computational domain was discretised with two meshes, both with 15 subdomains, 256 (512) boundary nodes and 959 (3823) internal nodes (dense mesh data in brackets). At the outlet fully developed flow was considered. Computations were made for *Re* number values 100, 400 and 800. In both test cases quadratic isoparametric continuous boundary elements and internal cells were used. In present work, all numerical tests were performed by using conjugate gradients squared with Jacobi or ILU preconditioning for the solution of partial problems. *Re* number was based on full chanel height and average inlet velocity.

Transactions on Modelling and Simulation vol 9, © 1995 WIT Press, www.witpress.com, ISSN 1743-355X

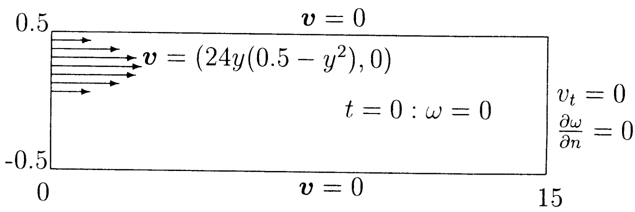

Figure 3: Backward facing step flow: geometry and boundary conditions

Table 3: Number of DDM iteration for Poiseuille flow

Method	*Re* =200	*Re* =2000	*Re* =20000
DDM-ASI	10	10	10
DDM-MSI	6	6	6
DDM-O-ASI	8	8	8
DDM-O-MSI	3	3	2

Tables 3 and 4 (coarse mesh) show results of DDM applied to one outer iteration of BDIM for vorticity transport equation. Table 3 show that number of iterations almost did't change with rising *Re* number value, while for BFS flow the change in iterations was obvious. This is a result of influence of starting guess (taken a vector of zeros) on speed of convergence. For Poiseuille's flow (for given boundary conditions) the flow field does not change with rising *Re* number values, but it changes significantly in BFS flow. When the initial guess from the previous outer iteration is taken, the number of iterations further decreases. The slight rise in DDM-O versions for BFS was probably caused by deteriorating exactness of internal derivatives, which are currently the weak point of DDM-O methods. Time development of velocity field for *Re* =400 (dense mesh) shows Figure 4.

Table 4: Number of DDM iteration for BFS flow

Method	DDM-ASI	DDM-MSI	DDM-O-ASI	DDM-O-MSI
Re =100	24	10	3	2
Re =800	5	5	5	3

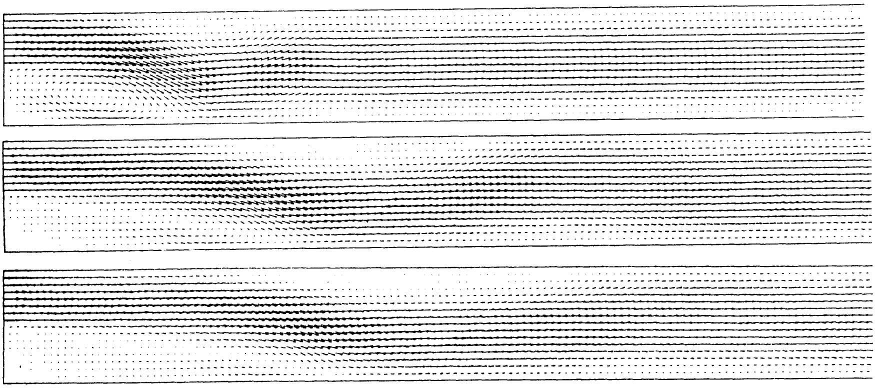

Figure 4: Velocity field, t = 1.6s, 5.2s and 8.2s, top to bottom (Re =400)

5 Conclusions

Domain decomposition method was used in BDIM for the solution of vorticity (energy) transport equation. The method has shown its potentials in both additive and multiplicative versions. Overlapping division of the domain was found to be the best choice regarding the speed of convergence. Krylov subspace methods (CGS) were successfully applied in solving partial problems. Computed test examples show that DDM-O-ASI and DDM-O-MSI are the most promising methods to use, especially when an integral approach to computing internal function derivatives will be derived.

References

[1] Skerget, L., Kuhn, G., Alujevic,A., Brebbia,C.A., : 'Time dependent transport problems by boundary element method', *Adv. Water Resources*, **12**, 9-20 (1989).

[2] Alujevic, A., Kuhn, G., Skerget, L.: ' Boundary elements for the solution of Navier-Stokes equations', *Comp. meth. appl. mech. eng.*, **91**, 1187-1201 (1991).

[3] Hribersek, M., Skerget, L., Zagar, I.: ' Boundary-Domain Integral method with Subdomain Technique for Time Dependent Viscous Fluid Flow ', *ZAMM Z. angew. Math. Mech.*, **73**, 7/8, 935-939 (1993).

[4] M.Hriberšek, L.Škerget, H.Mang: ' Preconditioned conjugate gradient methods for boundary-domain integral method' . *Engineering Analysis with Boundary Elements*, Southampton, - 12 (1993), 2, pp. 111-118.

[5] Hackbusch W.: 'Iterative Lösung grosser schwachbesetzter Gleichungsysteme', Teubner Studienbücher, Mathematik, Stuttgart, (1991).