Stationary incompressible viscous flow analysis by a domain decomposition method

Hiroshi Kanayama[1], Diasuke Tagami[2], and Masatsugu Chiba[3]

[1] Faculty of Engineering, Kyushu University kanayama@mech.kyushu-u.ac.jp
[2] Faculty of Engineering, Kyushu University tagami@mech.kyushu-u.ac.jp
[3] Department of Mechanical Engineering, Kyoto University

1 Introduction

There often encounter requirements to compute what flow pattern is generated in the stationary state. With progress of computer environment and increasing demand of precise analyses, numbers of degrees of freedom of such a computation become larger. However, as far as we know, computational codes may be rare, which are efficient for large scale, stationary, and nonlinear flow problems. Therefore, we have developed ADVENTURE_sFlow [3], which is one of modules included in the ADVENTURE project [1].

ADVENTURE_sFlow uses the Newton method as the nonlinear iteration, and to compute the problem at each step of the nonlinear iteration a stabilized finite element method is introduced. Moreover, to reduce the computational costs, an iterative domain decomposition method is applied to stabilized finite element approximations of stationary Navier–Stokes equations, where Generalized Product-type methods based on Bi-CG (GPBiCG) [6] is used as the iterative solver of the reduced linear system in each step of the nonlinear iteration. A parallel computing using the Hierarchical Domain Decomposition Method (HDDM) is also introduced.

Numerical results show that ADVENTURE_sFLow can analyze a stationary flow problem, where its degrees of freedom (DOF) is about 10 millions.

2 Formulation

Let Ω be a three-dimensional bounded domain with the Lipschitz continuous boundary Γ. We consider the stationary incompressible Navier–Stokes equations as follows:

$$\begin{cases} -\dfrac{1}{\rho}\nabla\cdot\sigma(u,p) + (u\cdot\nabla)\,u = \dfrac{1}{\rho}\,f & \text{in } \Omega, \quad (1a)\\ \nabla\cdot u = 0 & \text{in } \Omega, \quad (1b)\\ u = g & \text{on } \Gamma, \quad (1c)\end{cases}$$

where $u = (u_1, u_2, u_3)^T$ is the velocity [m/s], p is the pressure [N/m^2], ρ is the density [kg/m^3], $f = (f_1, f_2, f_3)^T$ is the body force [N/m^3], $g = (g_1, g_2, g_3)^T$ is the boundary velocity [m/s], and $\sigma(u, p)$ is the stress tensor [N/m^2] defined by

$$\sigma_{ij}(u,p) \equiv -p\delta_{ij} + 2\mu D_{ij}(u), \quad D_{ij}(u) \equiv \frac{1}{2}\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big), \qquad i,j = 1,2,3,$$

with the Kronecker delta δ_{ij} and the viscosity μ [kg/(ms)].

By application of the Newton method to (1) as the nonlinear iteration, the kth step linearized equations become the following: find (u^k, p^k) such that

$$\begin{cases} -\dfrac{1}{\rho}\nabla\cdot\sigma(u^k,p^k) + \big(u^{k-1}\cdot\nabla\big)\,u^k + \big(u^k\cdot\nabla\big)\,u^{k-1} \\ \qquad\qquad = \dfrac{1}{\rho}\,f + \big(u^{k-1}\cdot\nabla\big)\,u^{k-1} & \text{in } \Omega, \quad (2a) \\ \nabla\cdot u^k = 0 & \text{in } \Omega, \quad (2b) \\ u^k = g & \text{on } \Gamma. \quad (2c) \end{cases}$$

To avoid some intricate notations, we rewrite the linearized Navier–Stokes equations as follows: find (u, p) such that

$$\begin{cases} -\dfrac{1}{\rho}\nabla\cdot\sigma(u,p) + (w\cdot\nabla)\,u + (u\cdot\nabla)\,w = \widetilde{f} & \text{in } \Omega, \quad (3a) \\ \nabla\cdot u = 0 & \text{in } \Omega, \quad (3b) \\ u = g & \text{on } \Gamma, \quad (3c) \end{cases}$$

where w is a given velocity [m/s]. Obviously, the equations (3) yield (2) by substituting

$$u^{k-1}, \quad u^k, \quad p^k, \quad \text{and} \quad \frac{1}{\rho}f + \big(u^{k-1}\cdot\nabla\big)u^{k-1}$$

into w, u, p, and $\widetilde{f}$, respectively.

Let $\mathscr{T}_h$ be a decomposition of Ω consisting of a union of tetrahedra, and K a tetrahedron in $\mathscr{T}_h$. Let u_h and p_h be the velocity and the pressure approximated by $P1/P1$ elements. As in [3], the stabilized finite element method is introduced to (3) as follows: find $(u_h,\, p_h)$ satisfying (1c) such that

$$\begin{aligned} a_0(u_h,v_h) &+ a_1(w_h,u_h,v_h) + a_1(u_h,w_h,v_h) + b(v_h,p_h) + b(u_h,q_h) \\ &+ \sum_{K\in\mathscr{T}_h}\Big\{\tau_K\Big(\big(w_h\cdot\nabla\big)u_h + \big(u_h\cdot\nabla\big)w_h + \frac{1}{\rho}\nabla p_h, \\ &\qquad \big(w_h\cdot\nabla\big)v_h + \big(v_h\cdot\nabla\big)w_h - \frac{1}{\rho}\nabla q_h\Big)_K + \delta_K(\nabla\cdot u_h, \nabla\cdot v_h)_K\Big\} \\ &= (\widetilde{f}, v_h) + \sum_{K\in\mathscr{T}_h}\tau_K\Big(\widetilde{f}, \big(w_h\cdot\nabla\big)v_h + \big(v_h\cdot\nabla\big)w_h - \frac{1}{\rho}\nabla q_h\Big)_K, \qquad (4) \end{aligned}$$

where

$$a_0(u,v) \equiv \frac{2\mu}{\rho}\int_\Omega D(u):D(v)\,dx, \qquad a_1(w,u,v) \equiv \int_\Omega \big[(w\cdot\nabla)u\big]v\,dx,$$

$$b(v,q) \equiv -\frac{1}{\rho}\int_\Omega q\,\nabla\cdot v\,dx, \qquad (f,v) \equiv \int_\Omega fv\,dx, \qquad (f,v)_K \equiv \int_K fv\,dx,$$

v_h and q_h are the test functions satisfying $v_h = 0$ on Γ, w_h is the convection velocity approximated by $P1$ elements, and the notation ":" denotes the tensor product. The stabilized parameters τ_K and δ_K are defined by

$$\tau_K \equiv \min\left\{ \frac{h_K}{2\,\|w\|_\infty}, \frac{\rho\,h_K^2}{24\mu} \right\}, \quad \delta_K \equiv \min\left\{ \frac{\lambda\rho h_K^2 \|w\|_\infty^2}{12\mu}, \lambda h_K \|w\|_\infty \right\},$$

where λ denotes a positive constant, $\|w\|_\infty$ denotes the maximum norm of w in K, h_K denotes the diameter of K.

Let $\boldsymbol{Kx} = \boldsymbol{f}$ be the finite element system derived from (4), where $\boldsymbol{K}$ denotes the regular, asymmetric coefficient matrix corresponding to (4), $\boldsymbol{x}$ the vector corresponding to the velocity and the pressure, $\boldsymbol{f}$ the vector corresponding to the body force and the boundary velocity. Let Ω be divided into some subdomains. Let $\boldsymbol{x_i}$, $\boldsymbol{x_b}$, and $\boldsymbol{x_t}$ be vectors corresponding to DOF in the interior of Ω, on the interface between subdomains, and on Γ, where $\boldsymbol{x_t}$ is a given vector. Then, the system $\boldsymbol{Kx} = \boldsymbol{f}$ can be rewritten as follows:

$$\begin{bmatrix} \boldsymbol{K_{ii}} & \boldsymbol{K_{ib}} & \boldsymbol{K_{it}} \\ \boldsymbol{K_{bi}} & \boldsymbol{K_{bb}} & \boldsymbol{K_{bt}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x_i} \\ \boldsymbol{x_b} \\ \boldsymbol{x_t} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f_i} \\ \boldsymbol{f_b} \\ \boldsymbol{f_t} \end{Bmatrix}, \tag{5}$$

where $\boldsymbol{E}$ is a unit matrix. Eliminating $\boldsymbol{x_i}$ from (5), we can get the linear system on the interface:

$$\boldsymbol{Sx_b} = \boldsymbol{\chi}, \tag{6}$$

where

$$\boldsymbol{S} \equiv \boldsymbol{K_{bb}} - \boldsymbol{K_{bi}}\boldsymbol{K_{ii}}^{-1}\boldsymbol{K_{ib}},$$
$$\boldsymbol{\chi} \equiv \boldsymbol{f_b} - \boldsymbol{K_{bi}}\boldsymbol{K_{ii}}^{-1}\boldsymbol{f_i} - (\boldsymbol{K_{bt}} - \boldsymbol{K_{bi}}\boldsymbol{K_{ii}}^{-1}\boldsymbol{K_{it}})\boldsymbol{x_t}.$$

GPBiCG is allied to (6), and $\boldsymbol{x_b}$ is obtained. In the practical computing, the matrix $\boldsymbol{S}$ is not constructed explicitly. The products of matrices and vectors appearing in GPBiCG can be replaced by solving the Navier–Stokes equations in each subdomain, which implies that the method is fit for parallel computing; see, for example, [2]. The application of the skyline method to a problem in each subdomain yields $\boldsymbol{x_i}$ from $\boldsymbol{x_b}$. Therefore the solution in the whole domain at the nth step of the nonlinear iteration is obtained.

In the actual parallel computing, we adopt HDDM [5] for data and processor management to have the workload balanced among processors. It is already shown that HDDM is effective for a structural problem where the number of DOF is 100 millions [4].

3 Numerical examples

A station model is considered as a numerical example; see Fig. 1. The station has one plathome at the lower floor, one ticket gate at the upper floor, and three exits from the upper floor to the ground. The model considers the station that two trains are approaching along the red arrows in Fig. 1 with their speeds 1 [m/s], the fixed boundary conditions are imposed on the wall boundaries, and the air flows out from the other sides of the plathome and the exits with the stress-free conditions. The body force is set to be 0. The kinematic viscosity μ/ρ is set to be 1.0×10^{-1} [m^2/s].

As in Section 2, Ω is divided into a union of tetrahedra, and the flow field is approximated by $P1/P1$ elements: the number of elements and DOF are $18,873,133$ and $12,943,664$, respectively. The number of subdomains is set to be $300,000$. Throughout this section, λ is set to be 1.0.

As in Section 2, the Newton method is used for the nonlinear iteration. The initial value of the nonlinear iteration is the finite element solution of the corresponding Stokes problem. The nonlinear iteration is stopped when the relative rate of changes $\|\boldsymbol{x}^{n+1} - \boldsymbol{x}^n\|_\infty / \|\boldsymbol{x}^{n+1}\|_\infty$ becomes smaller than 1.0×10^{-4}, where $\boldsymbol{x}^n$ denotes the solution vector at the nth step, and $\|\,.\,\|_\infty$ the maximum norm.

In the Stokes equation for the initial condition, and in each step of the nonlinear iteration, the resultant linear systems on the interface are solved by GPBiCG with the simplified diagonal scaling preconditioner. The initial vector of the GPBiCG iteration is taken from zero vector in case of the Stokes equation for the initial condition of the nonlinear iteration, and is taken from the solution vector at the previous step at each step of the nonlinear iteration. The GPBiCG iteration is stopped when the relative residual norm $\|\boldsymbol{\chi} - \boldsymbol{S}\boldsymbol{x}_{\boldsymbol{b}}\|_2 / \|\boldsymbol{\chi}\|_2$ becomes smaller than 1.0×10^{-5}, where $\|\,.\,\|_2$ denotes the Euclidean norm. Computation of the model was performed on Alpha21264 with 30 CPUs at Computing and Communications Center, Kyushu University. It took about 100 hours to compute.

Fig. 2 shows the residual norm versus the number of GPBiCG iterations at each step of the nonlinear iteration. As the iteration progresses forward, the convergences of GPBiCG become faster. Fig. 3 shows relative rate of changes versus the number of nonlinear iterations. The nonlinear iteration by the Newton method goes well. Fig. 4 shows the streamlines in the station. In both cases, the flow comes into the station along the approaches of the trains, and goes out from the other sides of the plathome and from the exits.

At the end of this section, we consider the difficulty of computations in case of high Reynolds numbers and large scale problems. Table 1 shows the computational data on the mesh size and the numbers of DOF. Table 2 shows CPU time [min] in some cases of Reynolds numbers and meshes. In Cases I and II, the problem can be solved for six Reynolds numbers. However, as the scale is larger, the problem can not be solved for higher Reynolds numbers. Finally, in Case VI, the problem can be solved for only $Re = 50$.

4 Conclusion

To analyze the stationary Navier–Stokes equations, ADVENTURE_sFlow has been developed, which is one of the modules produced in the ADVENTURE project [1]. The Newton method has been introduced as the nonlinear iteration, and the stabilized finite element method as the approximation of the linearized equations at every steps of the nonlinear iteration. Moreover, for parallel computations, the iterative domain decomposition method and HDDM have been introduced, which are based on GPBiCG.

A station model, whose numbers of degrees of freedom is about 10 millions, has been analyzed.

We are going to analyze problems in case of higher Reynolds numbers or coupled problems.

References

1. ADVENTURE Project, ⟨http://adventure.q.t.u-tokyo.ac.jp/⟩
2. Glowinski, R., Dinh, Q. V., and Periaux, J., Domain decomposition methods for nonlinear problems in fluid dynamics, *Compt. Meths. Appl. Mech. Engrg.*, **40** (1983), pp.27–109.
3. Kanayama, H., Tagami, D., Araki, T., and Kume, H., A stabilization technique for steady flow problems, *Int. J. Comput. Fluid Dyn.*, **18** (2004), pp.297–301.
4. Shioya, R., and Yagawa, G., Iterative domain decomposition FEM with preconditioning technique for large scale problem, *ECM'99, Progress in Experimental and Computational Mechanics in Engineering and Material Behaviour*, (1999), pp.255–260.
5. Yagawa, G., and Shioya, R., Parallel finite elements on a massively parallel computer with domain decomposition, *Compt. Sys. Engrg.*, **4** (1993), pp.495–503.
6. Zhang, S.-L., GPBi-CG: generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems, *SIAM J. Sci. Comput.*, **18** (1997), pp.537–551.

Table 1. The maximum diameter of mesh and the numbers of DOF.

Case	I	II	III	IV	V	VI
Diameter [m]	1.60	0.90	0.80	0.71	0.59	0.50
DOF [$\times 10^5$]	0.5	2	3	4	7	10

DOF: in round numbers

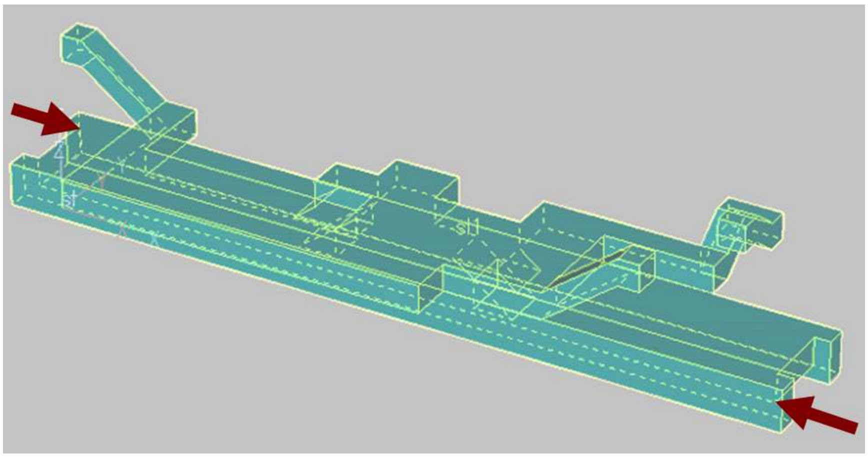

Fig. 1. A station model.

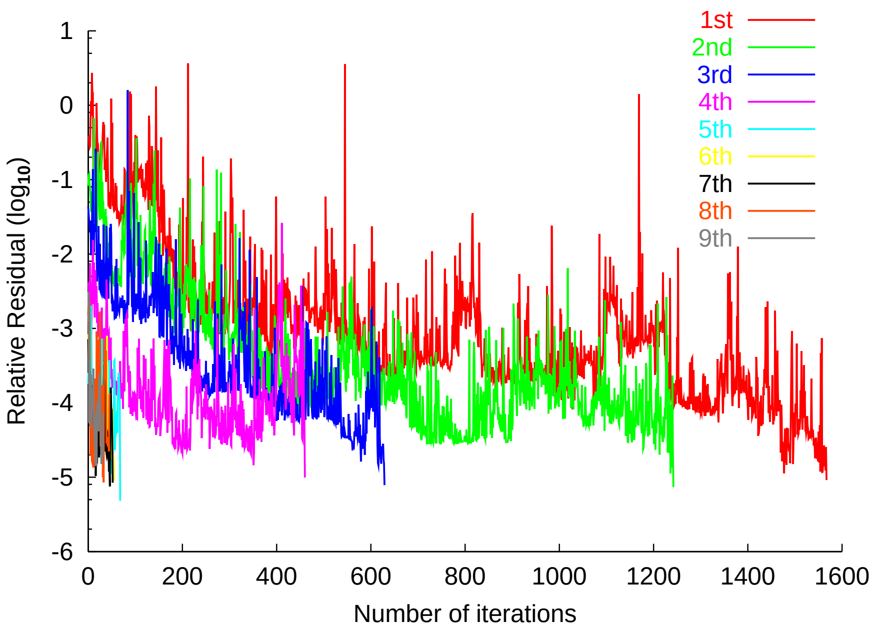

Fig. 2. Relative residuals of GPBiCG at each step of the nonlinear iteration.

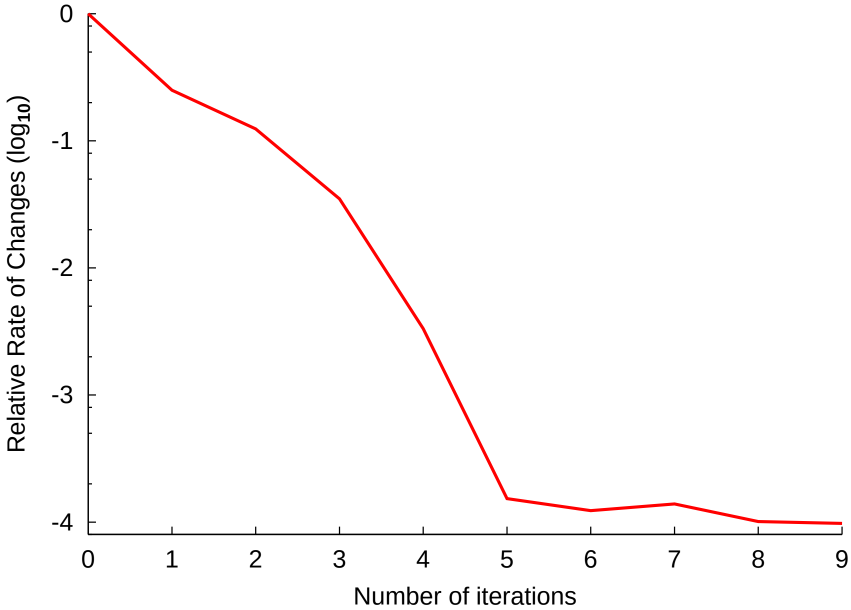

Fig. 3. Relative rates of changes in the Newton method.

Table 2. The number of iterations in case of some Reynolds numbers and meshes

Re	I	II	III	IV	V	VI
50	1.67	8.80	15.17	23.07	23.52	48.1
245	1.83	31.60	66.27	120.9	350.5	—
490	1.83	44.45	130.0	343.1	—	—
735	2.00	59.20	152.4	—	—	—
980	2.12	57.77	396.5	—	—	—
1225	2.25	63.91	—	—	—	—

Unit: [min], —: Not convergence

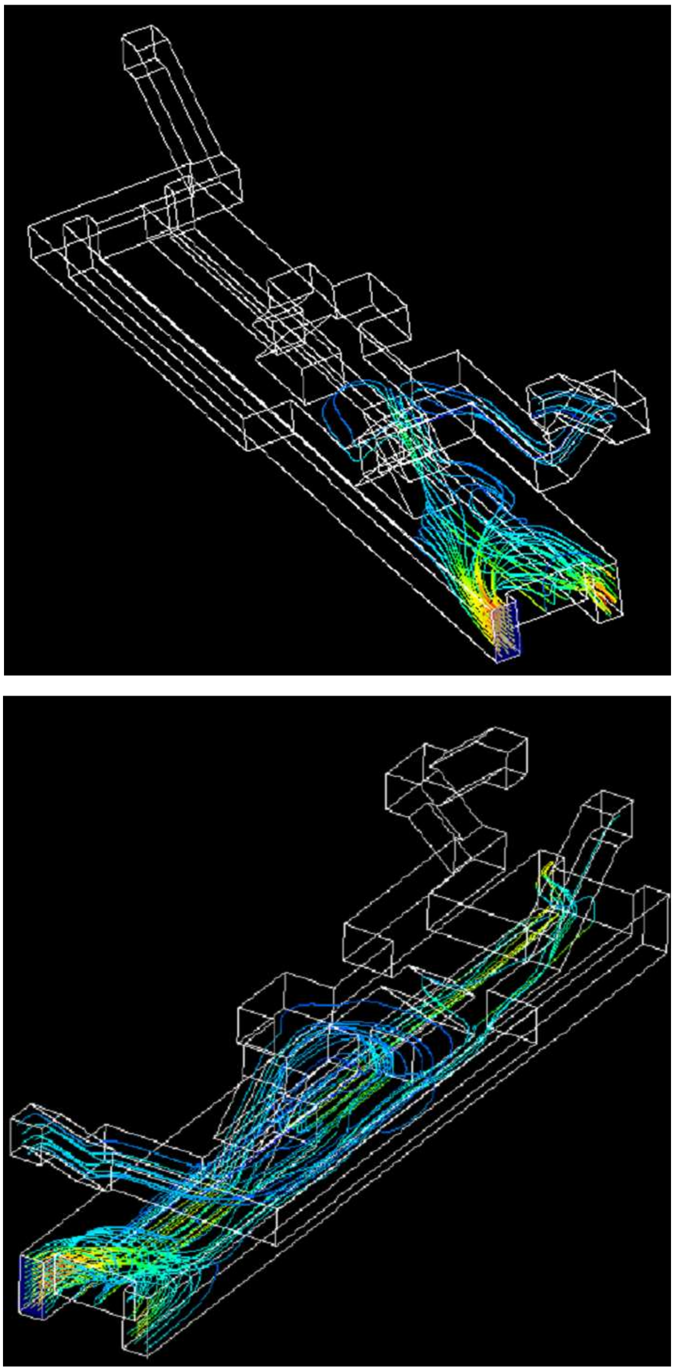

Fig. 4. The streamlines of the station model.