Parallel fluid-structure interaction simulation

Meng-Huo Chen[1][0000−0003−4058−332X]

Chung Cheng University, Taiwan

Abstract. In this work we implement the parallelization of a method for solving fluid-structure interactions: one-field monolithic fictitious domain (MFD). In this algorithm the velocity field for solid domain is interpolated into fluid velocity field through an appropriate L^2 projection, then the resulting combined equations are solved simultaneously (rather than sequentially). We parallelize the finite element discretization of spatial variables for fluid governing equations and linear system solver to accelerate the computation. Our goal is to reduce the simulation time for high resolution fluid-structure interaction simulation, such as collision of multiple immersed solids in fluid.

Keywords: fluid-structure interaction · finite element · parallel processing.

1 Introduction

1.1 Fluid-Structure interaction

Fluid-structure interaction (FSI) is a multiphysics problem that describes the interaction between a moving, sometimes deformable, structure and its surrounding incompressible fluid flow. In general, the solid materials deform largely and the deformations are strongly coupled to the flowing fluid. Numerical simulation of FSI is a computational challenge since the governing equations for solid and fluid regions are different and the algorithm needs to solve the locations of solid-fluid interfaces simultaneously with the dynamics in both regions where the kinematic (e.g., non-slipping) and dynamic (e.g., stress matched along the normal to solid-fluid interface) boundary conditions are imposed at the interface.

1.2 Numerical scheme

In general, the numerical schemes for solving the FSI problems may be classified into two approaches: the monolithic/fully-coupled method[1,2,3] and the partitioned/segregated method [4,5]. In addition, each method can be categorized further depends on the way to handle the mesh: fitted (conforming) mesh methods and unfitted (nonconforming) mesh methods.

Fitted mesh methods require solid and fluid meshes match each other at the interface, and both fluid and the solid regions share the nodes on the interface. In this way both a fluid velocity and a solid velocity (or displacement) are defined on each interface node. Clearly the fluid and solid velocities should agree on the interface nodes. Partitioned/segregated approach using fitted mesh to solve the governing equations. The solid and fluid equations are sequentially solved and then the steps are iterated until the velocities become consistent at the interface. This approach is easier to implement but not robust or fail to converge for problems where the fluid and solid appear to have significant energy exchange. On the other hand, monolithic/fully-coupled scheme solve the fluid and solid equations simultaneously on fitted mesh and often use a Lagrange Multiplier to weakly enforce the continuity of velocity on the interface. This method provides accurate and stable solutions but is computational challenging since one needs to solve the large size of nonlinear algebraic systems arising from the implicit discretization of the fully-coupled solid and fluid equations.

For unfitted mesh methods the solid and fluid regions are represented by two separate meshes and normally these do not agree to each other on the interface. Since there is no clear boundary for the solid problem, one of the approach to address the issue is to use Fictitious Domain method (FDM). In FDM the region representing solid is treated as (fictitious) fluid whose velocity/displacement is constrained to be the same as that of the solid. This constraint is enforced using a distributed Lagrange multiplier (DLM). There appear to be two situations for using unfitted meshes approach: either avoid solving the solid equations (such as Immersed Finite Element Method), or solve them with additional variables (two velocity fields and Lagrange multiplier) in the solid domain.

In this article, we parallelize one-field Fictitious Domain method can be categorized as a monolithic approach using an unfitted mesh. The main idea of the one-field FDM is as follows: (1) One-field formulation: re-write the governing equations for solid in the form of fluid equations. (2) L^2 projection (isoparametric interpolation): combining the fluid and solid equations and discretize them in an augmented domain. Then the problem is solved on a single field. The existing one-field FDM algorithm used in sequential simulation provide reasonable running time for 2D and 3D problems with low to moderate resolution, while its performance on highly resolved meshes ($256 \times 256 \times 256$ grid points and above) is not desired. Our goal is to accelerate the simulation algorithm using parallel computing and hope to extend its applications on FSI problems requiring high resolution, such as the 3D models describing the collision of multiple immersed solids in fluid.

2   Governing equations and FEM discretization

In this section we use a 2D FSI model (Figure 1) to describe the governing partial differential equations (PDEs) for FSI. Denote the regions representing solid

and fluid by $\Omega_t^s \subset \mathbb{R}^d$ and $\Omega_t^f \subset \mathbb{R}^d$, respectively. The subscript t reveals that both regions are time dependent. $\Omega_t^s \cup \Omega_t^f$ is the fixed domain and the moving interface between solid and fluid is denoted by $\Gamma_t = \partial \Omega_t^s \cap \Omega_t^f$. Components of variables along spatial directions are indicated by subscripts i, j, k. In addition, the repeated indices are implicitly summed over. For instance, u_j^s and u_j^f represent the j-component (along the j direction) of the solid velocity and fluid velocity, respectively, σ_{ij}^s and σ_{ij}^f denote the ij-component for stress tensor of solid and fluid respectively, and $(u_i^s)^n$ is the i-component of solid velocity at time t^n. Quantities denoted by bold letters implies that variables are vectors or matrices.
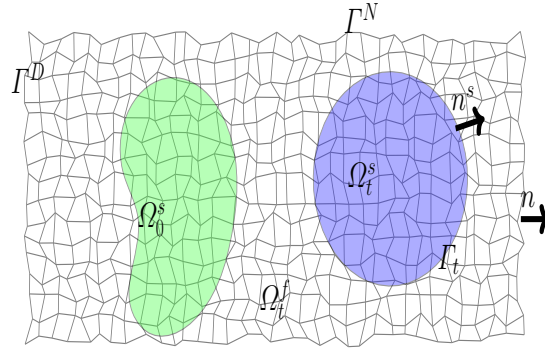

Fig. 1: 2D FSI problem and the boundary conditions.

Let ρ_s and ρ_f be the density of solid and fluid respectively, τ_{ij}^s and τ_{ij}^f be the deviatoric stress of the solid and fluid respectively, μ^f be the fluid viscosity, p^f be the fluid pressure, and g_i be the acceleration of gravity. Then the governing equations for incompressible fluid in Ω_t^f as shown in Figure 1 are

$$\frac{\partial u_j^f}{\partial x_j} = 0, \tag{1}$$

$$\rho^f \frac{Du_i^f}{Dt} - \frac{\partial \sigma_{ij}^f}{\partial x_j} = \rho^f g_i, \tag{2}$$

$$\sigma_{ij}^f = \mu^f \left(\frac{\partial u_i^f}{\partial x_j} + \frac{\partial u_j^f}{\partial x_i} \right) - p^f \delta_{ij} = \tau_{ij}^f - p^f \delta_{ij}, \tag{3}$$

For the evolution of variables in solid domain Ω_t^s, we assume the solid is neo-Hookean incompressible solid and the governing equations are:

$$\frac{\partial u_j^s}{\partial x_j} = 0, \tag{4}$$

$$\rho^s \frac{Du_i^s}{Dt} - \frac{\partial \sigma_{ij}^s}{\partial x_j} = \rho^s g_i, \tag{5}$$

$$\sigma_{ij}^s = \mu^s \left(\frac{\partial x_i^s}{\partial X_k} \frac{\partial x_j^s}{\partial X_k} - \delta_{ij} \right) - p^s \delta_{ij} = \tau_{ij}^s - p^s \delta_{ij}. \tag{6}$$

In the governing equations (4) - (6), μ^s and p^s are the solid shear modulus and the pressure of the solid, respectively, X_i represents the reference coordinates of the solid, and x_i refer to the current coordinates of the solid or fluid. Therefore the deformation tensor of the solid is denoted by $\mathbf{F} = [\partial x_i / \partial X_j]$. The incompressible neo-Hookean model described in equations (4) - (6) can be used for predicting stress-strain behavior of materials undertaking large deformation [12]. Note that D/Dt in equations (2) and (5) is the total time derivative and its form depends on what type of mesh (Eulerian or Lagrangian) used in the individual domain.

On the interface boundary Γ_t, the following conditions are imposed:

$$u_i^f = u_i^s, \tag{7}$$

$$\sigma_{ij}^f n_j^s = \sigma_{ij}^s n_j^s, \tag{8}$$

where n_j^s is the component of unit normal to the interface pointing outward, see Figure 1. Dirichlet and Neumann boundary conditions are imposed on the boundaries for the fluid accordingly:

$$u_i^f = u_i^s, \qquad \text{on} \qquad \Gamma^D \tag{9}$$

$$\sigma_{ij}^f n_j = \bar{h}_i \qquad \text{on} \qquad \Gamma^N. \tag{10}$$

The initial conditions used in this work is:

$$u_i^f \mid_{t=0} = u_i^s \mid_{t=0} = 0 \tag{11}$$

in which we assume the system starts from the rest.

The finite element discretization of the governing equations starts with weak formulation of equations (1),(2),(4) and (5). Let $(u, w)_\Omega = \int_\Omega uv \; d\Omega$, and

$$u_i = \begin{cases} u_i^f & \text{in} \quad \Omega_t^f \\ u_i^s & \text{in} \quad \Omega_t^s \end{cases}$$

$$p = \begin{cases} p^f & \text{in} \quad \Omega_t^f \\ p^s & \text{in} \quad \Omega_t^s \end{cases}$$

Using constitution equations (3) and (6) with boundary condition in (10), integrating the stress terms by parts for the test functions $v_i \in H_0^1(\Omega)$ and $q \in L^2(\Omega)$, we get the weak formulation of the FSI system: finding $u_i \in H^1(\Omega)$ and $p \in L_0^2(\Omega)$ so that

$$\rho^f \left(\frac{Du_i}{Dt}, v_i\right)_\Omega + \left(\tau_{ij}^f, \frac{\partial v_i}{\partial x_j}\right)_\Omega - \left(p, \frac{\partial v_j}{\partial x_j}\right)_\Omega - \left(\frac{\partial u_i}{\partial x_j}, q\right)_\Omega \tag{12}$$
$$+ (\rho^s - \rho^f)\left(\frac{Du_i}{Dt}, v_i\right)_{\Omega^s} + \left(\tau_{ij}^s - \tau_{ij}^f, \frac{\partial v_j}{\partial x_j}\right)_{\Omega^s}$$
$$= (\bar{h}, v_i)_{\Gamma^N} + \rho^f(g_i, v_i)_\Omega + (\rho^s - \rho^f)(g_i, v_i)_{\Omega^s},$$

$\forall v_i \in H_0^1(\Omega)$ and $\forall q \in H_0^1(\Omega)$.

The integrals in equation (12) are calculated in all domains as illustrated in Figure 1. Note that the fluid region is approximated by an Eulerian mesh and solid region is represented by an updated Lagrangian mesh in order to track the solid deformation/motion, therefore in these two different domains the total time derivatives are:

$$\frac{Du_i}{Dt} = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \quad \text{in} \quad \Omega \tag{13}$$

and

$$\frac{Du_i^s}{Dt} = \frac{\partial u_i^s}{\partial t} \quad \text{in} \quad \Omega^s. \tag{14}$$

Based on equations (13) and (14), the time discretization of equation (12) becomes

$$\rho^f \left(\frac{u_i - u_i^n}{\Delta t} + u_j \frac{\partial u_i}{\partial x_j}, v_i\right)_\Omega + \left(\tau_{ij}^f, \frac{\partial v_i}{\partial x_j}\right)_\Omega - \left(p, \frac{\partial v_j}{\partial x_j}\right)_\Omega - \left(\frac{\partial u_j}{\partial x_j}, q\right)_\Omega$$
$$+ (\rho^s - \rho^f)\left(\frac{u_i - u_i^n}{\Delta t}, v_i\right)_{\Omega_{n+1}^s} + \left(\tau_{ij}^s, \frac{\partial v_j}{\partial x_j}\right)_{\Omega_{n+1}^s} \tag{15}$$
$$= (\bar{h}_i, v_i)_{\Gamma^N} + \rho^f(g_i, v_i)_\Omega + (\rho^s - \rho^f)(g_i, v_i)_{\Omega_{n+1}^s},$$

where the superscript n of variable u_i represents the velocity at the nth time step. Note that we have replaced $\Omega_{t^{n+1}}^s$, the solid mesh at the $(n+1)$th time step, by Ω_{n+1}^s. By the spirit of splitting method introduced in [14], the above time evolution equation (15) can be viewed as the combination of two steps:

1. Convection step

$$\rho^f \left(\frac{u_i^\star - u_i^n}{\Delta t} + u_j^\star \frac{\partial u_i^\star}{\partial x_j}, v_i\right)_\Omega = 0 \tag{16}$$

2. Diffusion step

$$\rho^f \left(\frac{u_i - u_i^\star}{\Delta t}, v_i \right)_\Omega + \left(\tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_\Omega - \left(p, \frac{\partial v_j}{\partial x_j} \right)_\Omega - \left(\frac{\partial u_j}{\partial x_j}, q \right)_\Omega$$
$$+ (\rho^s - \rho^f) \left(\frac{u_i - u_i^n}{\Delta t}, v_i \right)_{\Omega_{n+1}^s} + \left(\tau_{ij}^s - \tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_{\Omega_{n+1}^s} \quad (17)$$
$$= (\bar{h}_i, v_i)_{\Gamma^N} + \rho^f (g_i, v_i)_\Omega + (\rho^s - \rho^f)(g_i, v_i)_{\Omega_{n+1}^s},$$

where the intermediate field $u_i^\star$ obtained from the convection step is used in the diffusion step to solve the "correct" field u_i. To obtain the system of linear algebraic equations, it is necessary to linearize equations (16) - (17). The details of the linearization of both equations are described in [13] and we list the final linearized form as follows:

1. Linearized weak form convection step using Talyor-Galerkin method:

$$(u_i^\star, v_i)_\Omega = \left(u_i^n - \Delta t u_j^n \frac{\partial u_i^n}{\partial x_j}, v_i \right)_\Omega - \frac{\Delta t^2}{2} \left(u_k^n \frac{\partial u_i^n}{\partial x_k}, u_j^n \frac{\partial v_i}{\partial x_j} \right)_\Omega \quad (18)$$

2. Linearized weak form of diffusion step

$$\rho^f \left(\frac{u_i - u_i^\star}{\Delta t}, v_i \right)_\Omega + (\rho^s - \rho^f) \left(\frac{u_i^s - (u_i^s)^n}{\Delta t}, v_i \right)_{\Omega_{n+1}^s}$$
$$+ \mu^f \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}, \frac{\partial v_i}{\partial x_j} \right)_\Omega - \left(p, \frac{\partial v_j}{\partial x_j} \right)_\Omega - \left(\frac{\partial u_j}{\partial x_j}, q \right)_\Omega$$
$$+ \mu^s \Delta t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \Delta t \frac{\partial u_i}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \Delta t \frac{\partial u_i^n}{\partial x_k} \frac{\partial u_j}{\partial x_k}, \frac{\partial v_i}{\partial x_j} \right)_{\Omega_{n+1}^s}$$
$$+ \Delta t^2 \left(\frac{\partial u_i}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l} + \frac{\partial u_i^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j}{\partial x_l}, \frac{\partial v_i}{\partial x_j} \right)_{\Omega_{n+1}^s} \quad (19)$$
$$+ \Delta t \left(\frac{\partial u_i}{\partial x_k} (\tau_{kj}^s)^n + (\tau_{il}^s)^n \frac{\partial u_j}{\partial x_l}, \frac{\partial v_i}{\partial x_j} \right)_{\Omega_{n+1}^s}$$
$$= (\bar{h}_i, v_i)_{\Gamma^N} + \rho^f (g_i, v_i)_\Omega + (\rho^s - \rho^f)(g_i, v_i)_{\Omega_{n+1}^s}$$
$$+ \left(\mu^s \Delta t^2 \frac{\partial u_i^n}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \Delta t^2 \frac{\partial u_i^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l} - (\tau_{ij}^s)^n, \frac{\partial v_i}{\partial x_j} \right)_{\Omega_{n+1}^s}$$

The spatial discretization in [13] uses a fixed Eulerian mesh for Ω and an updated Lagrangian mesh for Ω_{n+1}^s to discretize equation (19). The discretization Ω^h (for Ω) uses $\mathbf{P}_2$ (for velocities $\mathbf{u}$) $\mathbf{P}_1$ (for pressure p) elements (the Taylor-Hood element) with the corresponding finite element spaces

$$V^h(\Omega^h) = \text{span}\{\psi_1, ..., \psi_{N^u}\} \in H^1(\Omega)$$
$$L^h(\Omega^h) = \text{span}\{\phi_1, ..., \phi_{N^p}\} \in L^2(\Omega).$$

The approximated solution $\mathbf{u}^h$ and p^h can be expressed in terms of these basis functions as

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{N^{\mathbf{u}}} \mathbf{u}(\mathbf{x}_i)\psi_i(\mathbf{x}) \quad , \quad p^h(\mathbf{x}) = \sum_{i=1}^{N^p} p(\mathbf{x}_i)\phi_i(\mathbf{x}) \tag{20}$$

The solid domain Ω_{n+1}^s at the $n+1$ time step is discretized as Ω_{n+1}^{sh} using linear triangular elements with the corresponding finite element spaces as:

$$V^{sh}(\Omega_{n+1}^{sh}) = \text{span}\{\psi_1^s, ..., \psi_{N^s}^s\} \in H^1(\Omega_{n+1}^s), \tag{21}$$

and approximate $\mathbf{u}^h(\mathbf{x})\,|_{\mathbf{x}\in\Omega_{n+1}^{sh}}$ as:

$$\mathbf{u}^{sh}(\mathbf{x}) = \sum_{i=1}^{N^s} \mathbf{u}^h(\mathbf{x}_i^s)\psi_i^s(\mathbf{x}) = \sum_{i=1}^{N^s}\sum_{j=1}^{N^u} \mathbf{u}(\mathbf{x}_j)\psi_j(\mathbf{x}_i^s)\psi_i^s(\mathbf{x}) \tag{22}$$

where $\mathbf{x}_i^s$ is the nodal coordinate of the solid mesh.

Substituting (20), (22) and similar expressions for the test functions $\mathbf{v}^h$, q^h and $\mathbf{v}^{sh}$ into equation (18), we obtain the following matrix form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}, \tag{23}$$

where

$$\mathbf{A} = \mathbf{M}/\Delta t + \mathbf{K} + \mathbf{D}^T(\mathbf{M}^s/\Delta t + \mathbf{K}^s)\mathbf{D} \tag{24}$$

and

$$\mathbf{b} = \mathbf{f} + \mathbf{D}^T\mathbf{f}^s + \mathbf{M}\mathbf{u}^\star/\Delta t + \mathbf{D}^T\mathbf{M}^s\mathbf{D}\mathbf{u}^n/\Delta t \tag{25}$$

In equations (24) - (25), matrix $\mathbf{D}$ is the isoparametric interpolation matrix derived from equation (22) which can be expressed as

$$\mathbf{D} = \begin{bmatrix} \mathbf{P}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^T \end{bmatrix} \quad , \quad \mathbf{P}_{ij} = \psi_i(\mathbf{x}_j^s). \tag{26}$$

For the other matrices in (24) - (25), $\mathbf{M}$ and $\mathbf{K}$ are global mass matrix, global stiffness matrix from discretization of integrals in Ω^h. Similarly, $\mathbf{M}^s$ and $\mathbf{K}^s$ are mass matrix and stiffness matrix from discretization of integrals in Ω^{sh}. $\mathbf{u}$, $\mathbf{p}$, $\mathbf{f}$ are velocity, pressure and right hand side vectors, respectively. $\mathbf{B}$ and its transpose $\mathbf{B}^T$ represent the connections between pressure and velocities, which arise from the weak formulation in (19).

3   Parallelization of one-field FDM algorithm

3.1   Parallelism and related issues

The algorithm described in the previous section consists three parts: finite element discretization of the PDEs (calculation of element matrices, assembly of the global matrix), projecting matrix arising from solid equations into matrix arising from fluid equations (L^2 isoparametric projection) and solving the resulting system of linear algebraic equations. The computations of the first two parts are nearly embarrassingly parallel and require little or no communication, while the linear system solving using iterative scheme needs global communications (inner product) and communications from neighboring processes (matrix vector multiplications) for updating the residual and solution vectors.

3.2   Datatype and computation setup

The code implementation for solving the model described in the previous section is carried out in Campfire, where the structure grid generation is provided by PARAMESH. PARAMESH generates meshes as the union of blocks (arrays) of cells. Each block consists of internal cells and certain layer of guardcells (Figure 2). In finite element implementation each cell is treated as an (quadrilateral) element. For the 2D FSI simulation there are three variables to be solved: u (x-velocity), v (y-velocity) and p (pressure). For spatial variables we discretize the velocity field by quadratic quadrilateral ($P2$) elements and the pressure by linear quadrilateral ($P1$) elements. To accomplish storing the variable values in a 2D model, we employ four arrays from PARAMESH.
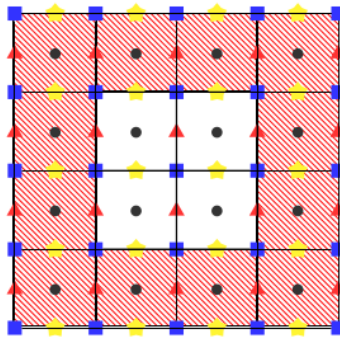

Fig. 2: A 2 × 2 block with one layer of guardcells

3.3   Linear system solver

As known, the solution of a linear system of equations constitutes an important part of the algorithm for solving PDEs numerically. Thus the properties of the linear system solver is crucial to the performance of PDE simulations. Generally, iterative methods consist of three core computational steps, which are vector update, matrix-vector multiplications and inner product computation. Vector updates is inherently parallel and there is no communication required. Parallel algorithm for matrix-vector multiplication relies on the data structure of matrix (sparsity, organization etc.) and typically local data communications between neighboring processes are sufficient for correct computation. Inner product calculation requires global data communications but the complexity could be moderate ($O(\log_2 P)$, where P is the number of processes). The parallel implementation of these three computation kernels will provide significant performance enhancement on iterative methods. To accelerate the MINRES iteration we use incomplete LU as the preconditioner. The computations for applying preconditioners involve backward/forward substitutions, which is inherently sequential. In the parallel implementation the preconditioner becomes block Jacobi type and the matrix entries connecting the neighboring processes are ignored. This causes the convergence properties of parallel preconditioning deteriorate significantly from sequential preconditioning.

4   Numerical experiments

In this work, we exam our parallel code on two 2D problems and compare the performance of parallel computation with the serial version in each case to access the parallel efficiencies. MINRES iterations stop when the 2-norm of the relative residual is less than 10^{-8} or the maximum number of iterations is reached (7500). For parallel simulations we use various number of processes up to 256 cores. All simulations were carried out on Taiwania, a supercomputer having a memory of 3.4 petabytes and delivering over 1.33 quadrillion flop/s of theoretical peak performance. The system has 630 compute nodes based on 40 core Intel Xeon Gold 6148 processors running at 2.4 GHz. Overall the system has a total of 25200 processor cores and 157 TB of aggregate memory. All the timing results are the average values from the running time of multiple simulations.

4.1   The motion of a disc in 2D Lid-driven cavity flow

In the first case we consider the motion of a deformable disc in a lid-driven cavity flow in 2D domain. Zhao et al. [15] studied the simulation for validating their methods. The experiment parameters are shown in Table 1 and Fig. 3 is the graphic demonstration of the problem setup. Initially, a round stress-free disc of radius 0.2 is centered at $(0.6, 0.5)$, then at $t = 0$ the top cavity wall starts
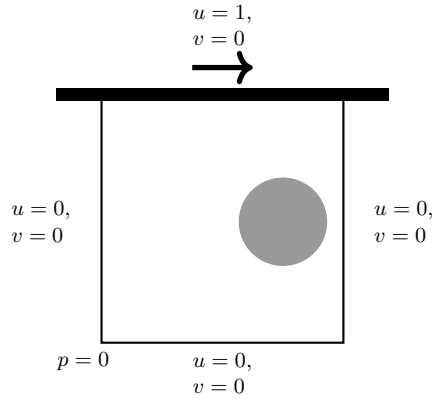
Fig. 3: A disc in lid-driven cavity flow with the boundary conditions.

moving horizontally at $u = 1$. For the material parameters, the density of the disc (ρ_s) and fluid (ρ_f) are both set to be 1. The elastic constant (μ_s) of the disc is 0.1 and the viscosity of the fluid (μ_f) is 0.01.

parameters	ρ_f	ρ_s	μ_f	μ_s
values	1	1	0.01	0.1

Table 1: Parameters for 2D solid (disc) in a lid-driven cavity flow

The fluid mesh resolution of the numerical simulation is 128×128, solid mesh contains 31163 elements and 15794 nodes. $\Delta t = 0.01$, 800 time steps. The running time for the last 100 time steps and speedup are shown in Table 1. As seen from the table, our method gains a roughly 47 speedup for 256 cores parallel simulations. Comparing with serial simulation, we reduce the overall running time (need 800 time steps) from roughly 65 hours to 1.5 hours. In the table we also list the average number of preconditioned MINRES iterations for the first 100 time steps in each simulation to show the impact of the parallelized preconditioning on the convergence of MINRES iterations. Notice that the number of MINRES iterations reach the maximum threshold (10000) after $t = 0.3$, corresponding to the scenario when the disc is near the wall.

The solid deformation are visualized in Fig. 4. The motion and the deformation of the solid are nearly identical to the result from [13] and [15]. We see that the disc deformation is asymmetric about the disc's vertical center line, and lubrication forces prevent the disk from touching the lid. The solid body ends up in a fixed position near the center of the cavity, and the velocity field becomes steady.

For the parallel simulation we gained speed up factor of 44 for 256 processes, corresponding to parallel efficiency of 17%. The increased number of MINRES iteration explains the reduction of the parallel efficiency from 82% (4 processes) to 17% (256 processes).

last 100 time steps	serial	4	16	64	256
time (minutes)	490	148	48	23	11
speed up	1	3.3	10.2	21.3	44.5
iterations (first 100 time steps)	2030	2420	2750	2980	3110

Table 2: Parallel simulation results

(a) $t = 1$ (b) $t = 4$ (c) $t = 5$
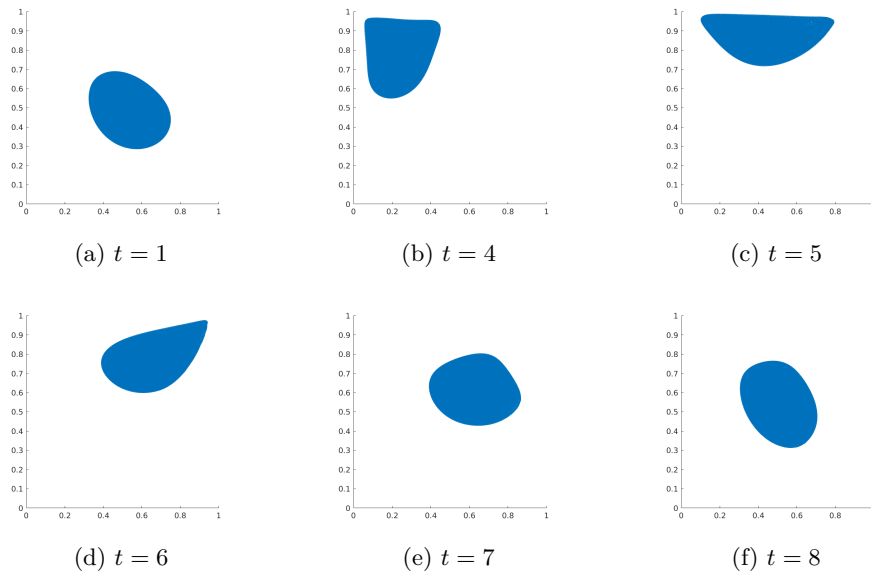
(d) $t = 6$ (e) $t = 7$ (f) $t = 8$

Fig. 4: The simulation results for a deformable solid motion in a lid-driven cavity flow.

4.2   Oscillation of a flexible leaflet oriented across the flow direction

The second example comes from [10,11,12], in which the problem was used for validation their methods. In this work we parallelize the one field monolithic FSI algorithm of this benchmark problem in [13]. The problem setup is described as follows. A leaflet is located at the center of the bottom horizontal side of a rectangular computational domain. The horizontal length and vertical length are 4 m and 1 m, respectively. The inflow velocity is in the x-direction and
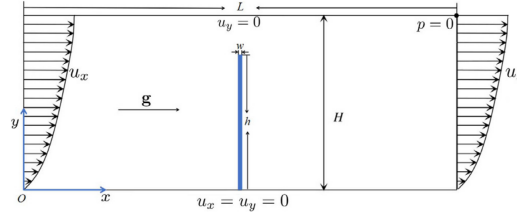
Fig. 5: Oscillation of a flexible leaflet across the flow direction.
Source: [13]

governed by $u_x = 15.0y(2-y)\sin(2\pi t)$. The computational domain and boundary conditions are illustrated in Fig.5.

The material parameters for the solid and fluid are listed in Table 3. The leaflet (solid region) is represented with 154 linear triangular elements with 116 nodes, and the corresponding fluid mesh is consist of 320×80 rectangular cells. The time step size for the evolution is $\Delta t = 5 \times 10^{-5}$s. All leaflet simulations run

parameters	ρ_f	ρ_s	μ_f	μ_s
values	100 kg/m^3	100 kg/m^3	10 N s/m^2	10^7 N/m^2

Table 3: Material parameters for oscillation of a flexible leaflet oriented across the flow direction.

from $t = 0$ to $t = 0.8$, which is 16000 time steps. Similar to the above example, we record simulation time for the middle 2000 time steps and the speedups of the parallel run are listed in Table 4. In this case the ratio of the number of solid mesh nodes to the number of fluid mesh nodes is much smaller than the example 1. For 256 processes simulation we gained a speed up factor of 28 corresponding to 11% parallel efficiency. Obviously the convergence of the preconditioned MINRES in this case is much worse than that in the example 1. This may attribute to the high elastic modulus (10^7 vs. 1) of the solid and the convergence of MINRES is more difficult.

Middle 2000 time steps	serial	16	64	256
time (minutes)	592	85	40	21
speed up	1	11.2	14.8	28.2
iterations (middle time steps)	2030	5010	> 7500	> 7500

Table 4: Parallel simulation results: leaflet in 2D channel flow
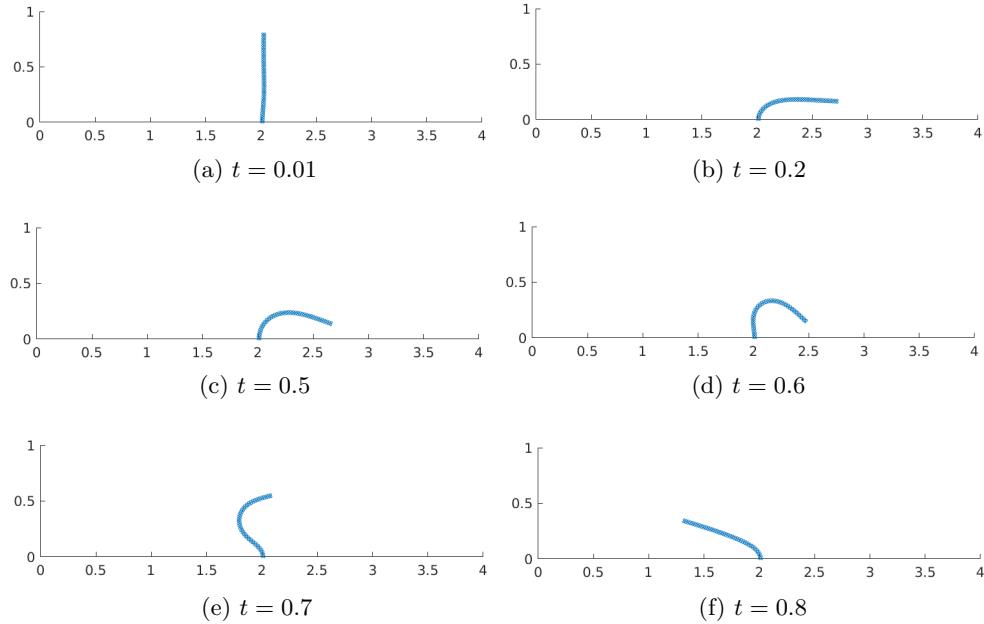
Fig. 6: The simulation results for oscillation of a flexible leaflet oriented across the flow direction.

5   Conclusion

In this research we have implemented the parallel computation on FSI 2D problem simulations. The convergence of preconditioned MINRES iteration is crucial to the parallel efficiency of the algorithm. For the first benchmark problem (2D disc in cavity-driven flow) the parallel simulation gain significant speed up although the parallel efficiency is not impressive. However, high elastic modulus of solid appeared in the second example (oscillating leaflet in channel flow) reduces the convergence performance of MINRES and hence the computation performance and parallel efficiency. One of the most important future work is to find preconditioners whose properties are minimally impacted by parallelization. With the modified preconditioner and algorithm we wish to extend our experiment to 3D FSI benchmark problems to verify our method.

References

1. M. Heil, *An efficient solver for the fully coupled solution of large displacement fluid-structure interaction problems*, Computer Methods in Applied Mechanics Engineer-

ing , 193 (1-2), pp. 1-23, 2004.

2. M. Heil, A. L. Hazel, J. Boyle, *Solvers for large-displacement fluid-structure interaction problems: segregated versus monolithic approaches*, Computational Mechanics, 43 (1), pp. 91-101, 2008.

3. R. L. Muddle, M. Mihajlovic, M. Heil, *An efficient preconditioner for monolithically-coupled large-displacement fluid-structure interaction problems with pseudo-solid mesh updates*, Journal of Computational Physics, 231 (21),, pp. 7315-7334, 2012.

4. U. Kuttler, W. A.Wall, *Fixed-point fluid-structure interaction solvers with dynamic relaxation*, Computational Mechanics, 43 (1), pp. 61-72, 2008.

5. J. Degroote, K. J. Bathe, J. Vierendeels, *Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction*, Computers & Structures, 87 (11-12), pp. 793-801, 2009.

6. G. Hou, J. Wang, A. Layton, *Numerical methods for fluid-structure interactiona review*, Communication Computational Physics, 12 (2), pp. 337-377, 2012

7. L. Zhang, A. Gerstenberger, X. Wang, W. K. Liu, *Immersed finite element method*, Computer Methods in Applied Mechanics and Engineering 193 (21), pp. 2051-2067, 2004.

8. L. Zhang, M. Gay, *Immersed finite element method for fluid-structure interactions*, Journal of Fluids and Structures, 23 (6), pp. 839-857, 2007.

9. X. Wang, L. T. Zhang, *Interpolation functions in the immersed boundary and finite element methods*, Computational Mechanics 45 (4), pp. 321-334, (2009).

10. R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Periaux, *A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow*, Journal of Computational Physics 169 (2), pp. 363-426, 2001.

11. Z. Yu, *A DLM/FD method for fluid/flexible-body interactions*, Journal of Computational Physics 207 (1), pp. 1-27, 2005.

12. F. P. Baaijens, *A fictitious domain/mortar element method for fluid-structure interaction*, International Journal for Numerical Methods in Fluids 35 (7), pp. 743-761, 2001.

13. Y. Wang, P. Jimack, M. Walkley, *A one-field monolithic fictitious domain method for fluid-structure interactions,* Computer Methods in Applied Mechanics Engineering, 317, pp. 1146-1168, 2017.

14. O. Zienkiewic,*The finite element method for fluid dynamics, 6th Edition*, Elsevier BV, 2005.

15. H. Zhao, J. B. Freund, R. D. Moser, *A fixed-mesh method for incompressible flow-structure systems with finite solid deformations*, Journal of Computational Physics, 227(6), pp. 3114-3140, 2008.

16. C.E. Goodyer, P.K. Jimack, A.M. Mullis, H.B. Dong, and Y. Xie. *On the fully implicit solution of a phase-field model for binary alloy solidification in three dimensions*, Advances in Applied Mathematics and Mechanics, 4, pp. 665 - 684, 2012.

17. P.C. Bollada, C.E. Goodyer, P.K. Jimack, A. Mullis and F.W. Yang. *Three dimensional thermal-solute phase field simulation of binary alloy solidification*, Journal of Computational Physics, 287, pp. 130-150, 2015.

18. W.A. Wall, *Fluid-struktur-interaktion mit stabilisierten finiten elementen*, Ph.D. thesis, Universitt Stuttgart, 1999.