Fluid-structure interactions: one-field monolithic fictitious domain method and its parallelization

Meng-Huo Chen

menghuo@ccu.edu.tw

January 13, 2022

Overview

Fluid-Structured Interaction

- Fluid-structure interaction is the interaction of some movable or deformable structure with a surrounding fluid flow.
- In this work, we use Fictitious Domain Method (FDM) to simulate the problems of Fluid-structure interaction. Also we parallelize the algorithm in order to speedup the computation.
- FDM uses two meshes to represent the fluid and solid separately.

The methods

- FDM approach solves for both velocity in the whole domain (fluid plus solid) and displacement of the solid simultaneously via a distributed Lagrange multiplier (DLM) to enforce the consistency of velocity/displacement in the overlapped solid domain.
- The solid is represented by a triangular mesh, while the mesh for the background fluid is rectangular.

Aim and approaches for this work

- The existing sequential FDM code for Fluid-Structure Interaction takes months to complete a 3D simulation (with low resolution).
- By parallelizing the computations for background fluid mesh, we hope to reduce the simulation time from months to few days.
- The parallel algorithm is designed based on the data structure and parallelism provided by software package PARAMESH.

The domain

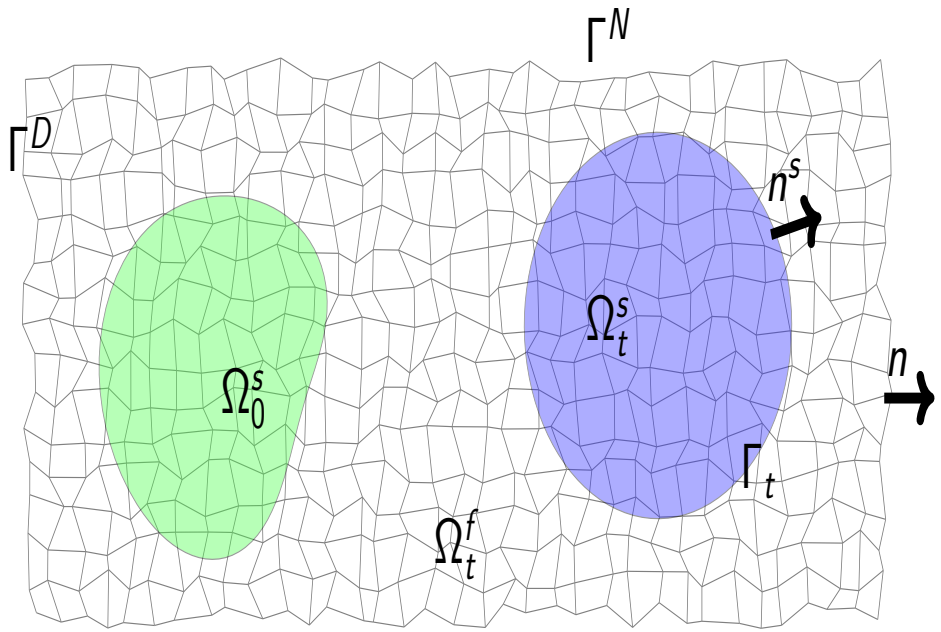


Figure: Lid-driven cavity flow and the boundary conditions.

Problem setup

- $\Omega_t^f \in \mathbb{R}^d$ and $\Omega_t^s \in \mathbb{R}^d$ ($d = 2$ in Fig. 1) denote the fluid and solid respectively which are time dependent regions.
- $\Omega_t^s \cup \Omega_t^f$ is the fixed domain and $\Gamma_t = \partial\Omega_t^f \cap \Omega_t^s$ is the moving interface between fluid and solid.
- All subscripts, such as i, j, k, represent spatial dimension. In addition, the repeated indices are implicitly summed over.
- u_i^f and u_i^s denote the velocity components of fluid and solid respectively, σ_{ij}^f and σ_{ij}^s denote the stress tensor components of fluid and solid respectively, and $(u_i^s)^n$ is a solid velocity component at time t^n.

Governing equation: fluid equation

- We assume an incompressible fluid governed by the following equations in Ω_i^f as shown in Fig. 1
  - Momentum equation:

$$\rho^f \frac{Du_i^f}{Dt} - \frac{\partial \sigma_{ij}^f}{\partial x_j} = \rho^f g_i. \tag{1}$$

  - Continuity equation:

$$\frac{\partial u_j^f}{\partial x_j} = 0 \tag{2}$$

  - Incompressible Newtonian fluid on fluid domain Ω^f:

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

Governing equation: solid equation

- On the solid domain, we assume an incompressible solid is governed by the following equations in Ω_i^s as shown in Fig. 1:
  - Momentum equation:

$$\rho^s \frac{Du_i^s}{Dt} - \frac{\partial \sigma_{ij}^s}{\partial x_j} = \rho^s g_i. \tag{4}$$

  - Continuity equation:

$$\frac{\partial u_j^s}{\partial x_j} = 0 \tag{5}$$

  - Incompressible viscous-hyperelastic solid on solid domain Ω^s

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

Governing equation: boundary and initial conditions

- On the interface boundary Γ_t:

$$\begin{aligned} u_i^f &= u_i^s \\ \sigma_{ij}^f n_j^s &= \sigma_{ij}^s n_j^s \end{aligned} \tag{7}$$

- Dirichlet and Neumann boundary conditions may be imposed for the fluid:

$$\begin{aligned} u_i^f &= \bar{u}_i^s \quad \text{on} \quad \Gamma^D \\ \sigma_{ij}^f n_j &= \bar{h}_i \quad \text{on} \quad \Gamma^N \end{aligned} \tag{8}$$

- The initial conditions are typically set as:

$$u_i^f\Big|_{t=0} = u_i^s|_{t=0} = 0 \tag{9}$$

Spatial discretization: weak formulation

- Let $(u, v)_\omega = \int_\omega uvd\omega$, and

$$u_i = \begin{cases} u_i^f \text{ in } \Omega_t^f \\ u_i^s \text{ in } \Omega_t^s \end{cases} \qquad p_i = \begin{cases} p_i^f \text{ in } \Omega_t^f \\ p_i^s \text{ in } \Omega_t^s \end{cases} \tag{10}$$

- Performing the following symbolic operation to obtain the spatial discretization.
  (Eq . (1), $v_i)_{\Omega_t^f}$ - (Eq .(2), $q)_{\Omega_t^f}$ + (Eq . (4), $v_i)_{\Omega_t^s}$ - (Eq . (5), $q)_{\Omega_t^s}$
  for independent test functions $v_i \in H_0^1(\Omega)$ and $q \in L_2(\Omega)$

Spatial discretization: weak formulation 2

Integrating the stress terms by parts, using constitutive equations (3) and (6) and boundary condition (10), gives the following weak form for the FSI system. Find $u_i \in H^1(\Omega)$ and $p \in L_0^2(\Omega)$ such that

$$\begin{aligned} & \rho^f \left(\frac{Du_i}{Dt}, v_i \right)_\Omega + \left(\tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_\Omega - \left(p, \frac{\partial v_j}{\partial x_j} \right)_\Omega - \left(\frac{\partial u_j}{\partial x_j}, q \right)_\Omega \quad (11) \\ + \quad & \left(\rho^s - \rho^f \right) \left(\frac{Du_i}{Dt}, v_i \right)_{\Omega_t^s} + \left(\tau_{ij}^s - \tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_{\Omega_t^s} \\ = \quad & (\bar{h}_i)_{\Gamma^N} + \rho^f (g_i, v_i)_\Omega + (\rho^s - \rho^f)(g_i, v_i)_{\Omega_t^s} \end{aligned}$$

for every $v_i \in H_0^1(\Omega)$ and $q \in L_2(\Omega)$.

Discretization in time

$\frac{D}{Dt}$ represents the total derivative of time, and

$$\frac{Du_i}{Dt} = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \quad \text{in} \quad \Omega \tag{12}$$

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

Using backward finite difference, equation (12) become

$$\begin{aligned}
& \rho^f \left(\frac{u_i - u_i^n}{\Delta t} + u_j \frac{\partial u_i}{\partial x_j}, v_i \right)_\Omega + \left(\tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_\Omega \\
- & \left(p, \frac{\partial v_j}{\partial x_j} \right)_\Omega - \left(\frac{\partial u_j}{\partial x_j}, q \right)_\Omega \\
+ & \left(\rho^s - \rho^f \right) \left(\frac{u_i - u_i^n}{\Delta t}, v_i \right)_{\Omega_{n+1}^s} + \left(\tau_{ij}^s - \tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_{\Omega_{n+1}^s} \\
= & (\bar{h}_i)_{\Gamma^N} + \rho^f (g_i, v_i)_\Omega + (\rho^s - \rho^f)(g_i, v_i)_{\Omega_{n+1}^s}
\end{aligned} \tag{14}$$

Two step splitting

Equation (14) is separated into convection part and diffusion part by the splitting method

- Convection step

$$\rho^f \left(\frac{u_i^\star - u_i^n}{\Delta t} + u_j^\star \frac{\partial u_i^\star}{\partial x_j}, v_i \right)_\Omega = 0 \tag{15}$$

- Diffusion step

$$\begin{aligned} & \rho^f \left(\frac{u_i - u_i^\star}{\Delta t}, v_i \right)_\Omega + \left(\tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_\Omega \\ - & \left(p, \frac{\partial v_j}{\partial x_j} \right)_\Omega - \left(\frac{\partial u_j}{\partial x_j}, q \right)_\Omega \\ + & \left(\rho^s - \rho^f \right) \left(\frac{u_i - u_i^n}{\Delta t}, v_i \right)_{\Omega_{n+1}^s} + \left(\tau_{ij}^s - \tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_{\Omega_{n+1}^s} \\ = & (\bar{h}_i)_{\Gamma^N} + \rho^f (g_i, v_i)_\Omega + (\rho^s - \rho^f)(g_i, v_i)_{\Omega_{n+1}^s} \end{aligned} \tag{16}$$

Discretization: finite element

- The spatial discretization for fluid domain use P_1P_2 rectangular element (the Taylor-Hood element). The solid domain is discretized by P_1 triangular elements. The corresponding finite element spaces are

$$V^h(\Omega^h) = span\{\psi_1, ..., \psi_{N^u}\} \subset H^1(\Omega) \quad (17)$$

$$L^h(\Omega^h) = span\{\phi_1, ..., \phi_{N^p}\} \subset L^2(\Omega) \quad (18)$$

and for the solid domain

$$V^{sh}(\Omega^{sh}_{n+1}) = span\{\phi^s_1, ..., \phi^s_{N^s}\} \subset H^1(\Omega^s_{n+1}).$$

- The approximate solutions for velocities and pressure are $\mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{N^u} \mathbf{u}(\mathbf{x}_i)\psi_i(\mathbf{x})$ and $p^h(\mathbf{x}) = \sum_{i=1}^{N^p} p(\mathbf{x}_i)\phi_i(\mathbf{x})$, respectively.
- The approximate solutions for solid domain is $\mathbf{u}^{sh}(\mathbf{x}) = \sum_{i=1}^{N^s}\sum_{j=1}^{N^u} \mathbf{u}(\mathbf{x}_j)\phi_j(\mathbf{x}^s_i)\phi^s_i(\mathbf{x})$

The global linear system

- The linear system arising from the above finite element discretization is

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}, \tag{19}$$

where $\mathbf{A} = \mathbf{M}/\Delta t + \mathbf{K} + \mathbf{D}^T(\mathbf{M}^s/\Delta t + \mathbf{K}^s)\mathbf{D}$, and $\mathbf{b} = f + \mathbf{D}^T f^s + \mathbf{M}\mathbf{u}^\star/\Delta t + \mathbf{D}^T\mathbf{M}^s\mathbf{D}\mathbf{u}^n/\Delta t$. Furthermore, $\mathbf{D}$ is the isoparametric interpolation matrix, $\mathbf{M}$ and $\mathbf{M}^s$ are mass matrices, $\mathbf{K}$ and $\mathbf{K}^s$ the stiffness matrices from discretization of the terms in the weak form related to fluid and solid, respectively. $\mathbf{B}$ from discretization of the linear functional related to the pressure variable. f and f^s are force vectors.

Overall solution algorithm

The solution algorithm at each time step is described as follows

1. Given the solid configuration $(\mathbf{x}^s)^n$ and velocity field at the time step n

$$\mathbf{u}^n = \begin{cases} (\mathbf{u}^f)^n & \text{in} \quad \Omega^f \\ (\mathbf{u}^s)^n & \text{in} \quad \Omega^s \end{cases}. \tag{20}$$

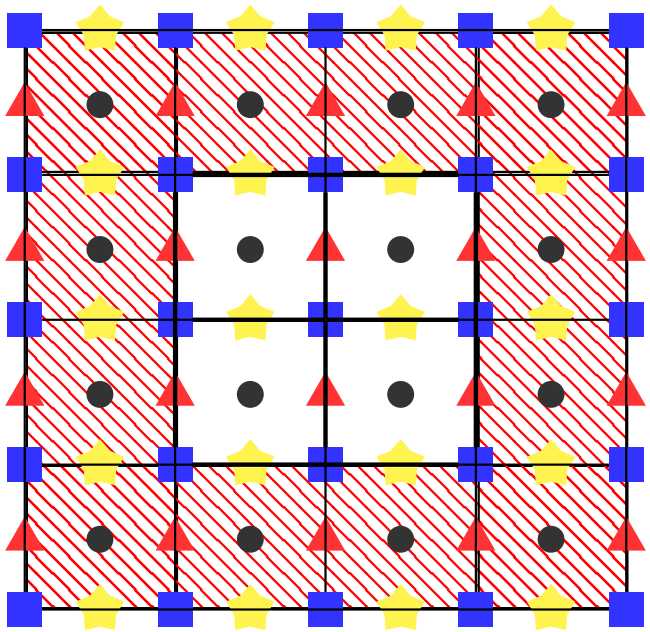
2. Discretize the convection equation and solve it to get an intermediate velocity $\mathbf{u}^\star$.
3. In diffusion step compute the interpolation matrix and solve linear system (19) using $\mathbf{u}^\star$ and $(\mathbf{u}^s)^n$ as initial values to get velocity field $\mathbf{u}^{n+1}$ at the time step $n+1$.
4. Compute solid velocity $(\mathbf{u}^s)^{n+1} = \mathbf{D}\mathbf{u}^{n+1}$ and update the solid mesh by $(\mathbf{x}^s)^{n+1} = (\mathbf{x}^s)^n + \Delta t(\mathbf{u}^s)^{n+1}$, then go to step (1) for the next time step.

Approaches for parallelization

- The parallelization implementation is carried out in 'Campfire', a software tool developed by Goodyear et al.
- In Campfire, the data structure and parallelism are provided by software package PARAMESH.
- At this stage we assess all the parallel issues for the fluid-structure cases on structured uniform grid in Campfire.

Data type: blocks

The basic data storage units in PARAMESH are blocks (shown as below). Each block consists of $n \times n$ cells, where generally n is an even integer. The outer layer cells of the block are guardcells which are used for communicating/exchanging data from neighboring blocks.



● : unk ★ : unk_e_x

▲ : unk_e_y ■ : unk_n

Data type: cells

In finite element implementation we view each cell as an rectangular element, consisting of 9 nodes for 2D domain.

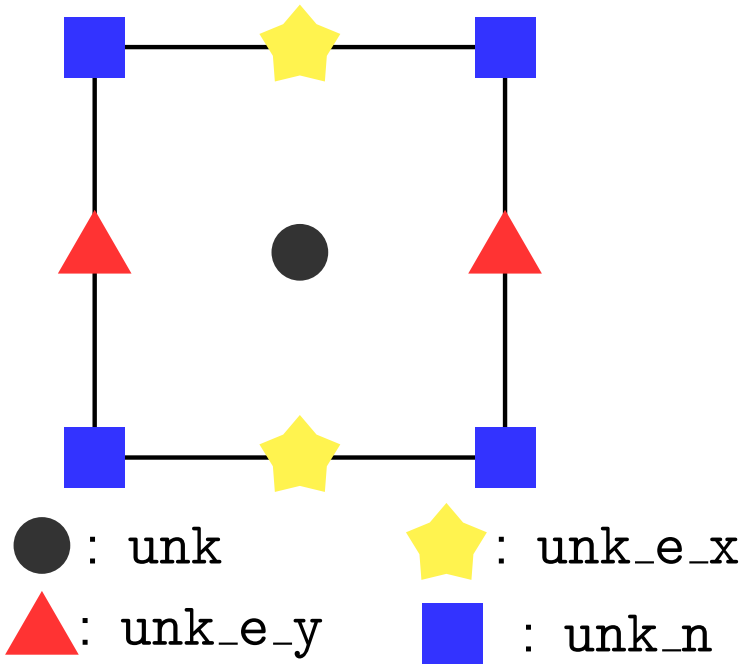


Figure: nodes in a cell, viewed as an element in our implementation.

Domain division

At the setup stage, the fluid domain is divided into blocks. Then each processor is assigned for one block.

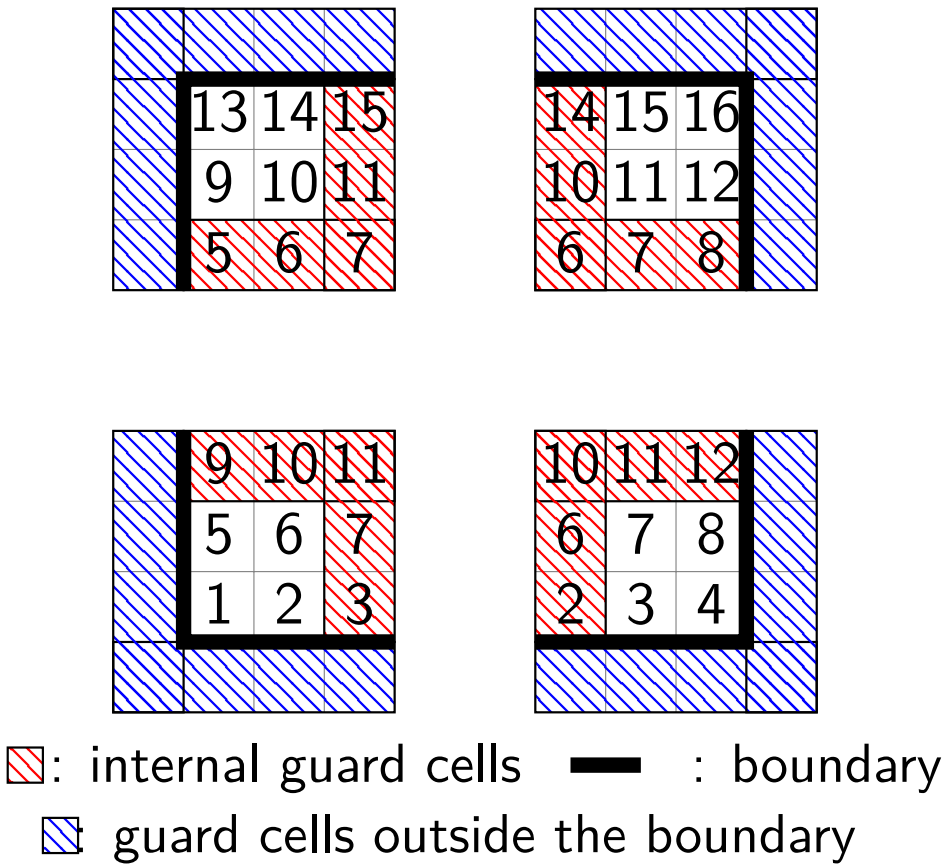


Figure: Graphic demonstration for four 2×2 block which have boundary cells and the guard cells containing the cells from neighbouring blocks.

Matrix assemble

- In our parallel implementation, each processor computes the whole stiffness and mass matrices for the solid mesh.
- On fluid domain the computations of element matrices are local for each process, which is one of the main source of the algorithm parallelism.
- The assembly of the global finite element matrix from element matrices is carried out in each process. In this way each process has a part of the rows of the global discretization matrix which are corresponding to the variables owned by the block in the process.
- To guarantee the correctness of the parallel computation, the largest radius of the triangle element in solid mesh can not be greater than two times of spatial grid space Δx or Δy. Therefore, we use two layers of guardcells for parallel communications.

Solving the linear system: GMRES

- The linear system (19) is solved using GMRES, an iterative linear system solver.
- The interpolation of solid matrix ($\mathbf{D}^T(\mathbf{M}^s/\Delta t + \mathbf{K}^s)\mathbf{D}$ in (19)) into fluid matrix is not done explicitly. Instead we compute the sum of the two products $(\mathbf{M}/\Delta t + \mathbf{K})\mathbf{v}$ and $(\mathbf{D}^T(\mathbf{M}^s/\Delta t + \mathbf{K}^s)\mathbf{D})\mathbf{v}$ in the matrix-vector multiplication step of GMRES. Both computations are local with respect to the block in a process.
- In each iteration there are one matrix-vector multiplication and two inner products.
- At the end of each iteration, communications between neighboring processes must be carried out so that the matrix-vector multiplication is the same as the result in serial computation.

GMRES preconditioner

- To speed up GMRES we use the preconditioner

$$\begin{bmatrix} \mathbf{A}' & 0 \\ 0 & \mathbf{I} \end{bmatrix} \tag{21}$$

  where $\mathbf{A}' = \mathbf{M}/\Delta t + K$

- The parallel version of the above preconditioner omits the connections between the variables located in different blocks. With proper change of ordering of variables, matrix $\mathbf{A}'$ is block diagonal. Therefore the parallel preconditioner is less effective than the serial preconditioner.
- In the cases tested so far, the iteration count for parallel preconditioned GMRES is 30% more than that for the serial preconditioned GMRES.

Case I: sphere in lid-driven cavity flow 2D

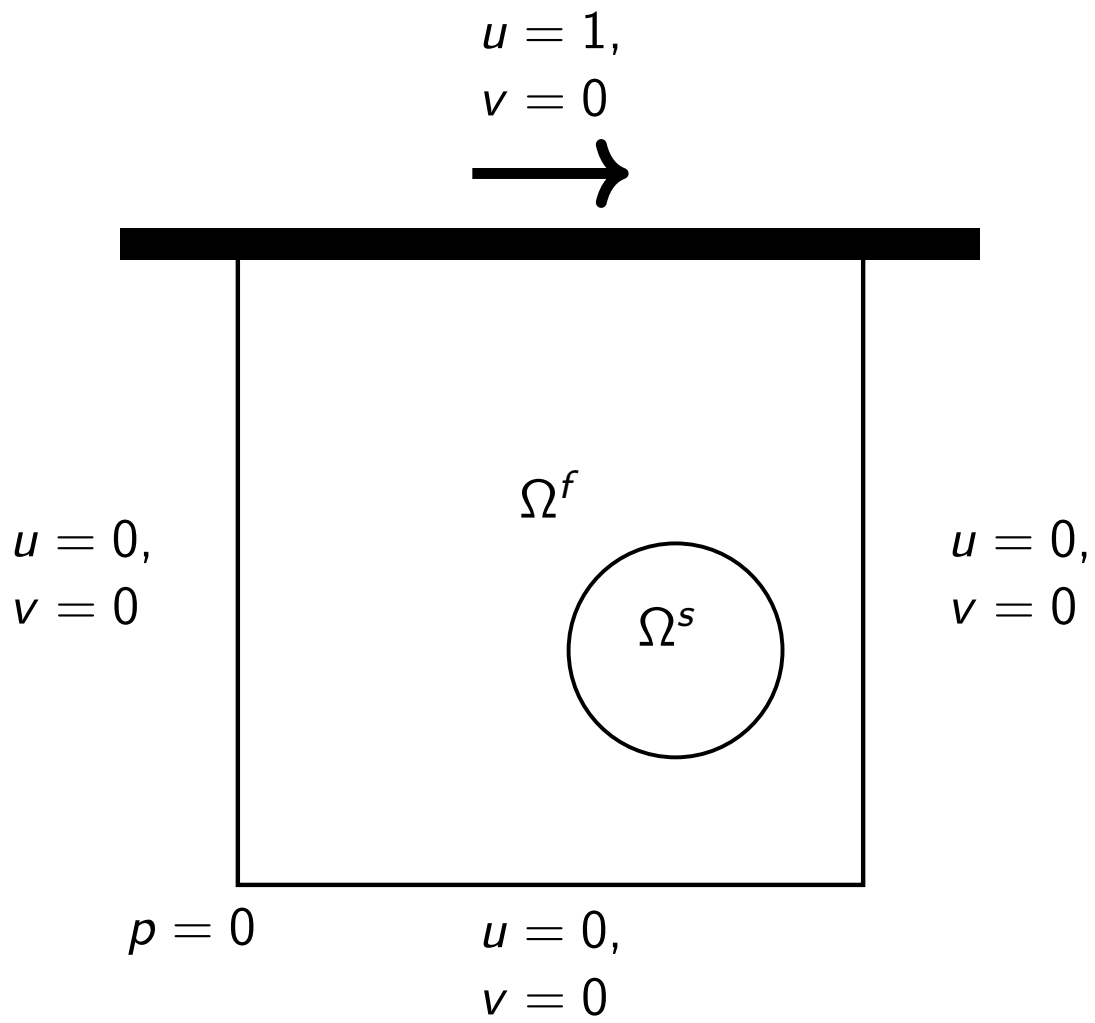


Figure: Lid-driven cavity flow and the boundary conditions.

Case I: sphere in lid-driven cavity flow 2D

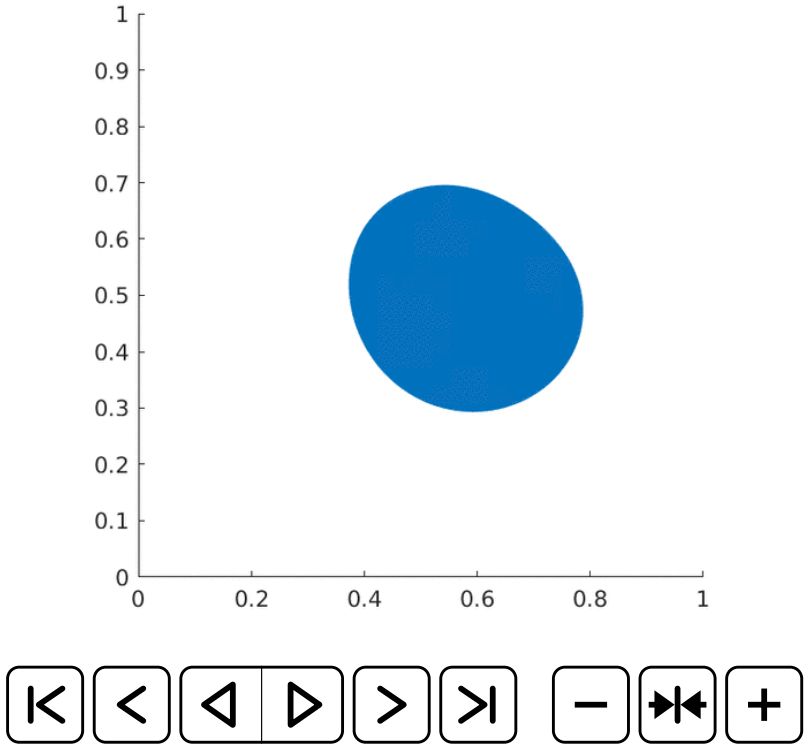
- we have the 2D parallel code running correctly. We tested the code on example with fluid mesh resolution 128×128, solid mesh contains 31163 elements and 15794 nodes. $\Delta t = 0.01$, 800 time steps.
- The running time for the last 100 time steps and speedup are shown as follows:

last 100 time steps	serial	4	16	64
time (minutes)	6512	1991	674	286
time (minutes)	1	3.27	9.66	22.77

Table: Triathlon results

Case I: sphere in lid-driven cavity flow 2D

- An output example of the simulation:



Case II: sphere in lid-driven cavity flow 3D

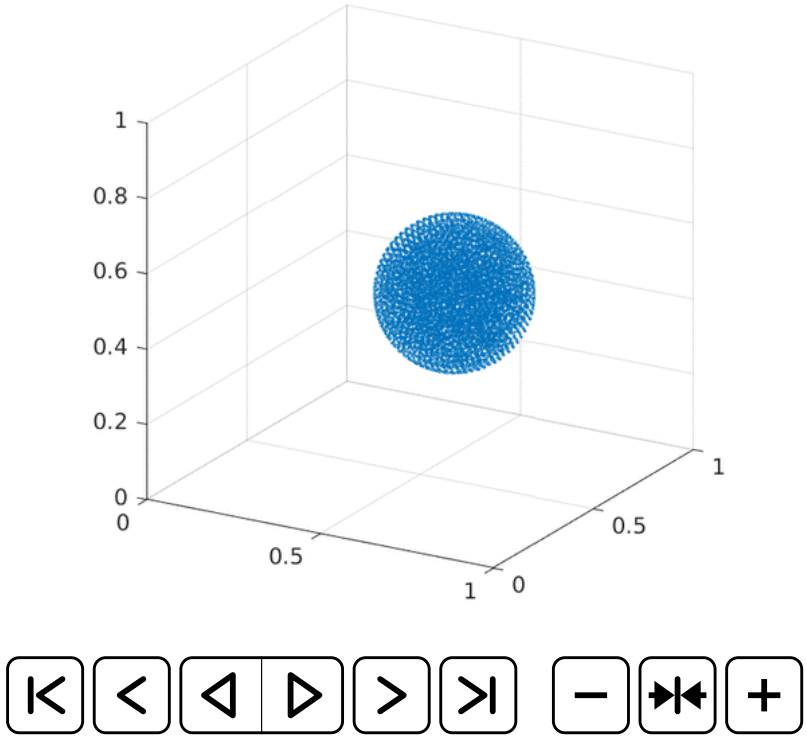
- we also have test the code for 3D case. The fluid mesh resolution is $32 \times 32 \times 32$, solid mesh contains 48649 elements and 8994 nodes. $\Delta t = 0.001$, 6000 time steps.
- The running time for the last 100 time steps and speedup are shown as follows:

100 time steps	serial	8	64
time (minutes)	8962	1273	315
speedup	1	7.04	28.45

Table: Triathlon results

Case II: sphere in lid-driven cavity flow 3D

- An output example of the simulation:



Case III: An oscillating Leaflet in 2D channel flow

- In this example the motion of a leaflet in 2D channel flow is simulated. The flow direction is perpendicular to the leaflet. The elastic modulus of the leaflet is 10^7, much higher than that in the previous cases. The fluid field is resolved by 384×96 elements and the solid mesh contains 154 elements and 116 nodes. At the entrance and exit of the channel the velocity is set to be

$$u = 15y(2 - y)\sin(2\pi t), \qquad v = 0.$$

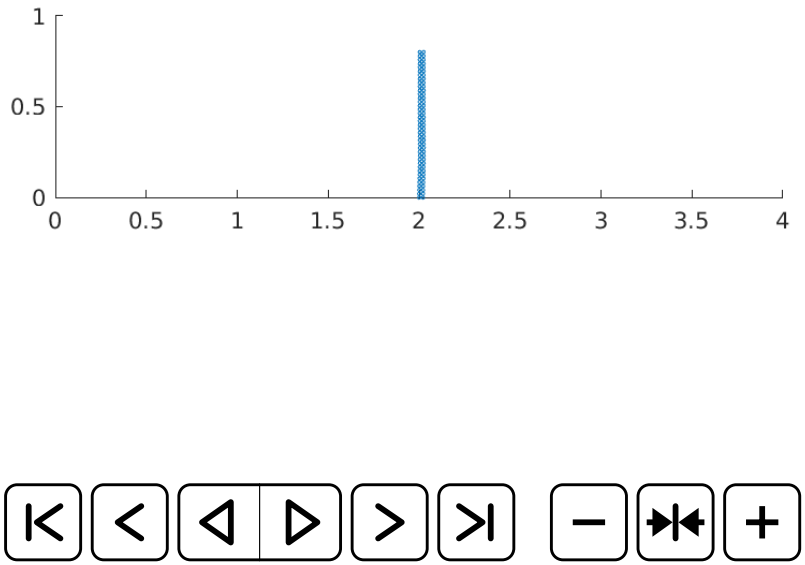
  The simulation's time step size is $\Delta t = 1 \times 10^{-4}$ and we ran for 10000 time steps.
- The running time from 1th time step to 2000th time step and the speed up are shown as follows:

first 2000 time steps	serial	8	32	128
time (minutes)	635	141	45	28
speed up	1	4.5	14.1	22.3

Table: Triathlon results

Case III: leaflet in channel flow 2D

- An output example of the simulation:



The End