



**HAL**  
open science

# Computational homogenization of unsteady flows with obstacles

Modesar Shakoor, Chung Hae Park

► **To cite this version:**

Modesar Shakoor, Chung Hae Park. Computational homogenization of unsteady flows with obstacles. International Journal for Numerical Methods in Fluids, 2022, 95 (4), pp.499-527. <10.1002/fld.5158>. <hal-03830690>

**HAL Id: hal-03830690**

**<https://hal.science/hal-03830690v1>**

Submitted on 26 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Computational homogenization of unsteady flows with obstacles

Modesar Shakoor\* <sup>1</sup> and Chung Hae Park<sup>1</sup>

<sup>1</sup>IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Materials and Processes, F-59000 Lille, France

October 26, 2022

## Abstract

Modeling flows in domains containing obstacles becomes challenging when the obstacles are very small compared to the domain. This problem is often known as modeling flow through permeable or porous media. Darcy's law is well-known to provide a robust solution to this problem for viscous flows, given that the permeability of the medium can be characterized. Solutions for inertial flows have not been proposed yet.

In this paper, a principle multiscale virtual power is formulated for the computational homogenization of unsteady incompressible flows in domains containing small obstacles. In this theory, the coarse scale of the domain is separated from the fine scale of the obstacles. The Finite Element (FE) implementation of this theory is developed in the form of a parallel FExFE (FE<sup>2</sup>) algorithm with two-way coupling between the two scales. The incompressibility constraint is handled with special care by introducing an independent pressure variable at both scales and relying on the Taylor-Hood P2/P1 pair.

Simulations are conducted to analyze the accuracy and efficiency of the proposed

---

\*Corresponding author: modesar.shakoor@imt-nord-europe.fr

multiscale approach. The results show that the method is robust and interesting from both the points of view of computational cost and accuracy when the ratio between the domain size and the obstacle size is larger than 100.

*Keywords: unsteady flow, porous media, multiscale, homogenization, FE<sup>2</sup>*

## 1 Introduction

While flows around obstacles have been widely studied and can be solved easily using widely available finite volume or Finite Element (FE) codes, this becomes more difficult when obstacles are small compared to the domain. This problem is often known as modeling flow through permeable or porous media. It is well understood and can be modeled accurately even when the small obstacles have a complex distribution and morphology as long as the flow is steady and laminar [1].

Indeed, in the steady and laminar case, scale separation is often assumed. The domain, which can be a reservoir for oil engineering applications, or a mold for material processing applications, is considered as homogeneous. This is the coarse scale domain. The heterogeneity due to the small obstacles, which can be pores for a reservoir or fibrous reinforcements for a mold, is modeled through a fictitious heterogeneous domain. This is the fine scale domain. This separation of scales avoids the discretization of the fine heterogeneity directly within the coarse scale domain. It is instead represented only within the fine scale domain [1].

A common issue with scale separation is that governing equations are clear at the fine scale, but not boundary conditions, while it is the opposite at the coarse scale. This is the reason why scale separation has to be completed by a scale transition theory. This is only possible with a principle of multiscale virtual power or a homogenization theory establishing the conservation of energy between the scales, and relating coarse scale variables to their fine scale counterparts [2]. For viscous flows, the governing equations at the fine scale are the Stokes equations. Numerous theories have been proposed to relate the fine scale velocity and pressure fields to their coarse scale counterparts, with various choices for the boundary conditions at the fine scale. These theories show that the governing equations for the coarse scale involve Darcy's law [1]. The latter relies on the definition of

a permeability tensor, which can be computed analytically or numerically depending on the complexity of the distribution and morphology of the heterogeneity at the fine scale. It is generally not possible to use Darcy’s law for inertial flows. Extensions such as Forchheimer’s law have been proposed to model inertial effects, but they are not based on homogenization theory [3, 4]. Recently, Blanco *et al.* proposed a principle of multi-scale virtual power for steady flows with obstacles and inertial effects [5]. The governing equations at the fine scale were the Navier-Stokes equations for incompressible Newtonian flows at the steady state but with the auto-advection term. Blanco *et al.* showed that it is possible to use a wide range of boundary conditions for the fine scale and establish the conservation of energy between the scales. In this approach, it was necessary to solve a nonlinear fine scale problem for each point of the coarse scale domain. However, the authors did not implement their approach with a full coupling between coarse and fine scale variables. This simplification was possible because they only addressed steady flows.

Full or two-way coupling between coarse and fine scale variables is widely used in computational solid mechanics, where it is known as computational homogenization [6], concurrent multiscale method [7] or FExFE (FE<sup>2</sup>) scheme [8]. The basis for these techniques is the same as for fluid mechanics: scale separation and scale transition. The difference mainly resides in the numerical implementation of the multiscale theory.

In the present work, an extension of Blanco *et al.*’s approach to the unsteady case is proposed, as well as a numerical implementation with full coupling between the two scales. In this approach, the governing equations for the fine scale are the unsteady Navier-Stokes equations for incompressible Newtonian flows with obstacles. A principle of multiscale virtual power with Lagrange multipliers is presented in Sec. 2 to establish the boundary conditions for the fine scale problem as well as the conservation of energy between the scales. Details on the numerical implementation are given in Sec. 3, in particular regarding the incompressibility constraint and the two-way coupling. Indeed, at each time increment, a nonlinear problem has to be solved for the coarse scale, and at each iteration of the coarse scale nonlinear solver, a nonlinear fine scale problem has to be solved at each integration point of the coarse scale domain. This elaborate multiscale

computational model is deployed and compared to single-scale models in Sec. 4.

## 2 Multiscale model

The coarse scale domain is denoted  $\Omega^M \subset \mathbb{R}^d$ , with Dirichlet boundary conditions  $\mathbf{v}_D^M$  on  $\Gamma_D^M$  and Neumann boundary conditions  $\mathbf{t}_N^M$  on  $\Gamma_N^M$ . Regarding time, it is defined as  $t \in [0, T]$ , with  $T \in \mathbb{R}^{+*}$ . The variational equation to satisfy at the coarse scale is to find  $\mathbf{v}^M(\cdot, t) \in \mathcal{V}^M(t), p^M \in L^2(\Omega^M)$ , such that

$$\begin{aligned} \int_{\Omega^M} (\mathbf{f}^M(\mathbf{x}, t) \cdot \delta \mathbf{v}^M(\mathbf{x}) + \boldsymbol{\sigma}^{M, dev}(\mathbf{x}, t) : \nabla_{\mathbf{x}} \delta \mathbf{v}^M(\mathbf{x})) \, d\mathbf{x} &= \int_{\Gamma_N^M} \mathbf{t}_N^M(\mathbf{x}, t) \cdot \delta \mathbf{v}^M(\mathbf{x}) \, d\mathbf{x}, \\ \int_{\Omega^M} (-p^M(\mathbf{x}) \nabla_{\mathbf{x}} \cdot \delta \mathbf{v}^M(\mathbf{x}) - \delta p^M(\mathbf{x}) \nabla_{\mathbf{x}} \cdot \mathbf{v}^M(\mathbf{x})) \, d\mathbf{x} & \\ \forall \delta \mathbf{v}^M \in \mathcal{V}^M(t), & \\ \forall \delta p^M \in L^2(\Omega^M), & \end{aligned} \quad (1)$$

with functional spaces

$$\begin{aligned} \mathcal{V}^M(t) &= \left\{ \mathbf{w} \in (H^1(\Omega^M))^d, \mathbf{w}(\mathbf{x}) = \mathbf{v}_D^M(\mathbf{x}, t), \forall (\mathbf{x}, t) \in \Gamma_D^M \times [0, T] \right\}, \\ H^1(\Omega^M) &= \left\{ w \in L^2(\Omega^M), \nabla_{\mathbf{x}} w \in (L^2(\Omega^M))^d \right\}, \\ L^2(\Omega^M) &= \left\{ w : \Omega^M \mapsto \mathbb{R}, \int_{\Omega^M} w(\mathbf{x})^2 \, d\mathbf{x} < +\infty \right\}. \end{aligned}$$

This is standard and easier to understand if incompressible Newtonian flow is modeled directly at the coarse scale. Navier-Stokes equations then lead to:

$$\begin{cases} \mathbf{f}^M(\mathbf{x}, t) = \rho^M(\mathbf{x}) \left( \frac{\partial \mathbf{v}^M}{\partial t}(\mathbf{x}, t) + \mathbf{v}^M(\mathbf{x}, t) \cdot \nabla_{\mathbf{x}} \mathbf{v}^M(\mathbf{x}, t) \right), & \forall \mathbf{x} \in \Omega^M, \forall t \in [0, T], \\ \boldsymbol{\sigma}^{M, dev}(\mathbf{x}, t) = 2\mu^M(\mathbf{x}) \nabla_{\mathbf{x}}^{S, dev} \mathbf{v}^M(\mathbf{x}, t), \end{cases}$$

where the deviatoric part is

$$\nabla^{S, dev} \mathbf{v}^M = \nabla^S \mathbf{v}^M - \frac{1}{d} \text{tr}(\nabla^S \mathbf{v}^M) \mathbf{I},$$

and the symmetric part

$$\nabla^S \mathbf{v}^M = \frac{1}{2} (\nabla \mathbf{v}^M + \nabla^T \mathbf{v}^M).$$

The problem with this direct approach is that the computational cost is going to blow up if there are obstacles of a small size compared to  $\Omega^M$ , as the element or cell size should be

smaller than the obstacle size. The alternative proposed herein is a multiscale approach, in which  $\mathbf{f}^M(\mathbf{x}, t)$  and  $\boldsymbol{\sigma}^{M,dev}(\mathbf{x}, t)$  are unknown functions of  $\mathbf{v}^M(\mathbf{x}, t)$  and  $\nabla_{\mathbf{x}}\mathbf{v}^M(\mathbf{x}, t)$ , while  $p^M$  is still a Lagrange multiplier.

Consequently, a fine scale domain  $\Omega^m = \Omega^m(\mathbf{x}) \subset \mathbb{R}^d$  is introduced at each point  $\mathbf{x}$  of the coarse scale domain. Small obstacles are represented within this domain instead of the coarse scale domain. The boundary of those obstacles is  $\Gamma_O^m = \Gamma_O^m(\mathbf{x}) \subset \partial\Omega^m(\mathbf{x})$ , and Dirichlet boundary conditions  $\mathbf{0}$  are imposed on  $\Gamma_O^m$ . The proposed multiscale approach is kinematic, which means averaging constraints are going to be imposed on the velocity field, while the expressions of  $\mathbf{f}^M(\mathbf{x}, t)$  and  $\boldsymbol{\sigma}^{M,dev}(\mathbf{x}, t)$  are going to be derived from a so-called principle of multiscale virtual power [2].

## 2.1 Kinematic averaging

The average of the fine scale velocity field is constrained to be equal to the coarse scale velocity:

$$\mathbf{v}^M(\mathbf{x}, t) = \frac{1}{|\Omega^m|} \int_{\Omega^m} \mathbf{v}^m(\mathbf{y}, t) d\mathbf{y}.$$

Similarly, the average of the fine scale velocity gradient field is constrained to be equal to the coarse scale velocity gradient:

$$\nabla_{\mathbf{x}}\mathbf{v}^M(\mathbf{x}, t) = \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}}\mathbf{v}^m(\mathbf{y}, t) d\mathbf{y}.$$

While it is theoretically possible to add constraints on the higher order derivatives and introduce additional terms in the virtual powers, it is assumed that a first order approach is enough for unsteady incompressible Newtonian flows with obstacles. This will be verified in Sec. 4.

## 2.2 Kinematic admissibility

Kinematic admissibility requires the averaging constraints for the velocity field to be satisfied. These constraints will all be added using Lagrange multipliers except for the Dirichlet boundary conditions on the obstacles, thus the functional spaces for the fine scale unknowns are

$$\mathbf{v}^m(\cdot, t) \in \mathcal{V}^m,$$

$$\mathcal{V}^m = \left\{ \mathbf{w} \in (H^1(\Omega^m))^d, \mathbf{w}(\mathbf{y}) = \mathbf{0}, \forall \mathbf{y} \in \Gamma_O^m \right\}.$$

Given that additional Lagrange multipliers are set, these functional spaces can also be used for virtual variations.

## 2.3 Virtual powers

The coarse scale virtual power  $P^M$  is based on the coarse scale problem stated in Eq. (1):

$$\begin{aligned} P^M(\delta \mathbf{V}^M, \delta \mathbf{G}^M, \delta p^M) &= \mathbf{f}^M \cdot \delta \mathbf{V}^M + \boldsymbol{\sigma}^{M,dev} : \delta \mathbf{G}^M - p^M \text{tr}(\delta \mathbf{G}^M) - \delta p^M \nabla_{\mathbf{x}} \cdot \mathbf{v}^M, \\ \forall (\delta \mathbf{V}^M, \delta \mathbf{G}^M, \delta p^M) &\in \mathbb{R}^d \times \mathbb{R}^{d \times d} \times \mathbb{R}. \end{aligned}$$

It is interesting to see that thanks to the Lagrange multiplier  $p^M$ , the space for the virtual variation  $\delta \mathbf{G}^M$  is simple and very large.

The averaged fine scale virtual power  $P^m$  is based on unsteady Navier-Stokes equations for incompressible Newtonian flow written at the fine scale:

$$\begin{aligned} P^m(\delta \mathbf{V}^M, \delta \mathbf{G}^M, \delta \mathbf{v}^m, \delta p^m, \delta \boldsymbol{\alpha}, \delta \boldsymbol{\beta}) &= \\ \frac{1}{|\Omega^m|} \int_{\Omega^m} &\left[ \begin{aligned} &\rho^m(\mathbf{y}) \left( \frac{\partial \mathbf{v}^m}{\partial t}(\mathbf{y}, t) + \mathbf{v}^m(\mathbf{y}, t) \cdot \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t) \right) \cdot \delta \mathbf{v}^m(\mathbf{y}) \\ &+ 2\mu^m(\mathbf{y}) \nabla_{\mathbf{y}}^{S,dev} \mathbf{v}^m(\mathbf{y}, t) : \nabla_{\mathbf{y}} \delta \mathbf{v}^m(\mathbf{y}) \\ &- p^m(\mathbf{y}) \nabla_{\mathbf{y}} \cdot \delta \mathbf{v}^m(\mathbf{y}) - \delta p^m(\mathbf{y}) \nabla_{\mathbf{y}} \cdot \mathbf{v}^m(\mathbf{y}, t) \end{aligned} \right] d\mathbf{y} \\ &- \delta \boldsymbol{\alpha} \cdot \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \mathbf{v}^m(\mathbf{y}, t) d\mathbf{y} - \mathbf{v}^M(\mathbf{x}, t) \right) \\ &- \delta \boldsymbol{\beta} : \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t) d\mathbf{y} - \nabla_{\mathbf{x}} \mathbf{v}^M(\mathbf{x}, t) \right) \\ &- \delta \boldsymbol{\alpha} \cdot \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \delta \mathbf{v}^m(\mathbf{y}) d\mathbf{y} - \delta \mathbf{V}^M \right) \\ &- \delta \boldsymbol{\beta} : \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}} \delta \mathbf{v}^m(\mathbf{y}) d\mathbf{y} - \delta \mathbf{G}^M \right), \\ &\forall (\delta \mathbf{V}^M, \delta \mathbf{G}^M) \in \mathbb{R}^d \times \mathbb{R}^{d \times d}, \\ &\forall (\delta \mathbf{v}^m, \delta p^m) \in \mathcal{V}^m \times L^2(\Omega^m), \\ &\forall (\delta \boldsymbol{\alpha}, \delta \boldsymbol{\beta}) \in \mathbb{R}^d \times \mathbb{R}^{d \times d}. \end{aligned}$$

The divergence constraint on the fine scale velocity gradient has been added through Lagrange multiplier  $p^m \in L^2(\Omega^m)$  and the kinematic averaging constraints on the fine

scale velocity field and its gradient have been added through Lagrange multipliers

$$(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{R}^d \times \mathbb{R}^{d \times d}.$$

Similarly, Lagrange multipliers have been added for the variations of the fine scale unknowns. This allows keeping the solutions and variations spaces simple and very large.

## 2.4 Principle of multiscale virtual power

The principle of multiscale virtual power establishes the balance between the two scales. Therefore, the problem is to find  $(\mathbf{v}^m(\cdot, t), p^m, \boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{V}^m \times L^2(\Omega^m) \times \mathbb{R}^d \times \mathbb{R}^{d \times d}$ , with  $(\mathbf{v}^M(\cdot, t), p^M) \in \mathcal{V}^M(t) \times L^2(\Omega^M)$  such that

$$\begin{aligned} P^M(\delta \mathbf{V}^M, \delta \mathbf{G}^M, \delta p^M) &= P^m(\delta \mathbf{V}^M, \delta \mathbf{G}^M, \delta \mathbf{v}^m, \delta p^m, \delta \boldsymbol{\alpha}, \delta \boldsymbol{\beta}), \\ \mathbf{f}^M \cdot \delta \mathbf{V}^M + \boldsymbol{\sigma}^{M, dev} : \delta \mathbf{G}^M - p^M \text{tr}(\delta \mathbf{G}^M) - \delta p^M \nabla_{\mathbf{x}} \cdot \mathbf{v}^M &= \\ \frac{1}{|\Omega^m|} \int_{\Omega^m} \left[ \begin{aligned} &\rho^m(\mathbf{y}) \left( \frac{\partial \mathbf{v}^m}{\partial t}(\mathbf{y}, t) + \mathbf{v}^m(\mathbf{y}, t) \cdot \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t) \right) \cdot \delta \mathbf{v}^m(\mathbf{y}) \\ &+ 2\mu^m(\mathbf{y}) \nabla_{\mathbf{y}}^{S, dev} \mathbf{v}^m(\mathbf{y}, t) : \nabla_{\mathbf{y}} \delta \mathbf{v}^m(\mathbf{y}) \\ &- p^m(\mathbf{y}) \nabla_{\mathbf{y}} \cdot \delta \mathbf{v}^m(\mathbf{y}) - \delta p^m(\mathbf{y}) \nabla_{\mathbf{y}} \cdot \mathbf{v}^m(\mathbf{y}, t) \end{aligned} \right] d\mathbf{y} \\ \Leftrightarrow & -\delta \boldsymbol{\alpha} \cdot \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \mathbf{v}^m(\mathbf{y}, t) d\mathbf{y} - \mathbf{v}^M(\mathbf{x}, t) \right) \\ & -\delta \boldsymbol{\beta} : \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t) d\mathbf{y} - \nabla_{\mathbf{x}} \mathbf{v}^M(\mathbf{x}, t) \right) \\ & -\boldsymbol{\alpha} \cdot \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \delta \mathbf{v}^m(\mathbf{y}) d\mathbf{y} - \delta \mathbf{V}^M \right) \\ & -\boldsymbol{\beta} : \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}} \delta \mathbf{v}^m(\mathbf{y}) d\mathbf{y} - \delta \mathbf{G}^M \right), \\ & \forall (\delta \mathbf{V}^M, \delta \mathbf{G}^M, \delta p^M) \in \mathbb{R}^d \times \mathbb{R}^{d \times d} \times \mathbb{R}, \\ & \forall (\delta \mathbf{v}^m, \delta p^m) \in \mathcal{V}^m \times L^2(\Omega^m), \\ & \forall (\delta \boldsymbol{\alpha}, \delta \boldsymbol{\beta}) \in \mathbb{R}^d \times \mathbb{R}^{d \times d}. \end{aligned} \tag{2}$$

By taking arbitrary variations from Eq. (2), it is possible to derive the problem to solve at the fine scale with all necessary boundary conditions, and also to derive the expressions of the missing terms for the coarse scale problem in Eq. (1).

## 2.5 Fine scale problem

As the spaces for the variations are very large, it is possible to zero out  $\delta \mathbf{V}^M, \delta \mathbf{G}^M, \delta p^M$ . This leads to the problem of finding  $(\mathbf{v}^m(\cdot, t), p^m, \boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{V}^m \times L^2(\Omega^m) \times \mathbb{R}^d \times \mathbb{R}^{d \times d}$ , with

$(\mathbf{v}^M(\cdot, t), p^M) \in \mathcal{V}^M(t) \times L^2(\Omega^M)$  such that

$$\begin{aligned}
& \frac{1}{|\Omega^m|} \int_{\Omega^m} \left[ \begin{aligned} & \rho^m(\mathbf{y}) \left( \frac{\partial \mathbf{v}^m}{\partial t}(\mathbf{y}, t) + \mathbf{v}^m(\mathbf{y}, t) \cdot \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t) \right) \cdot \delta \mathbf{v}^m(\mathbf{y}) \\ & + 2\mu^m(\mathbf{y}) \nabla_{\mathbf{y}}^{S,dev} \mathbf{v}^m(\mathbf{y}, t) : \nabla_{\mathbf{y}} \delta \mathbf{v}^m(\mathbf{y}) \\ & - p^m(\mathbf{y}) \nabla_{\mathbf{y}} \cdot \delta \mathbf{v}^m(\mathbf{y}) - \delta p^m(\mathbf{y}) \nabla_{\mathbf{y}} \cdot \mathbf{v}^m(\mathbf{y}, t) \end{aligned} \right] d\mathbf{y} \\
& - \delta \boldsymbol{\alpha} \cdot \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \mathbf{v}^m(\mathbf{y}, t) d\mathbf{y} - \mathbf{v}^M(\mathbf{x}, t) \right) \\
& - \delta \boldsymbol{\beta} : \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t) d\mathbf{y} - \nabla_{\mathbf{x}} \mathbf{v}^M(\mathbf{x}, t) \right) \\
& - \boldsymbol{\alpha} \cdot \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \delta \mathbf{v}^m(\mathbf{y}) d\mathbf{y} \right) \\
& - \boldsymbol{\beta} : \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}} \delta \mathbf{v}^m(\mathbf{y}) d\mathbf{y} \right) = 0, \\
& \forall (\delta \mathbf{v}^m, \delta p^m) \in \mathcal{V}^m \times L^2(\Omega^m), \\
& \forall (\delta \boldsymbol{\alpha}, \delta \boldsymbol{\beta}) \in \mathbb{R}^d \times \mathbb{R}^{d \times d}.
\end{aligned} \tag{3}$$

This system of variational equations will be solved for each fine scale domain using the FE method. In a strong form it can be written as

$$\left\{ \begin{aligned} & \rho^m(\mathbf{y}) \left( \frac{\partial \mathbf{v}^m}{\partial t}(\mathbf{y}, t) + \mathbf{v}^m(\mathbf{y}, t) \cdot \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t) \right) \\ & - \nabla_{\mathbf{y}} \cdot (2\mu^m(\mathbf{y}) \nabla_{\mathbf{y}}^{S,dev} \mathbf{v}^m(\mathbf{y}, t)) \\ & + \nabla_{\mathbf{y}} p^m(\mathbf{y}) - \boldsymbol{\alpha} \end{aligned} \right\} = \mathbf{0}, \forall \mathbf{y} \in \Omega^m, \\
& \mathbf{v}^m(\mathbf{y}, t) = \mathbf{0}, \forall \mathbf{y} \in \Gamma_O^m, \\
& (2\mu^m(\mathbf{y}) \nabla_{\mathbf{y}}^{S,dev} \mathbf{v}^m(\mathbf{y}, t) - p^m(\mathbf{y}) \mathbf{I}) \cdot \mathbf{n}^m(\mathbf{y}) = \boldsymbol{\beta} \cdot \mathbf{n}^m(\mathbf{y}), \forall \mathbf{y} \in \partial \Omega^m \setminus \Gamma_O^m, \\
& \nabla_{\mathbf{y}} \cdot \mathbf{v}^m(\mathbf{y}, t) = 0, \forall \mathbf{y} \in \Omega^m, \\
& \frac{1}{|\Omega^m|} \int_{\Omega^m} \mathbf{v}^m(\mathbf{y}, t) d\mathbf{y} = \mathbf{v}^M(\mathbf{x}, t), \\
& \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t) d\mathbf{y} = \nabla_{\mathbf{x}} \mathbf{v}^M(\mathbf{x}, t),
\end{aligned} \right.$$

where  $\mathbf{n}^m(\mathbf{y})$  is the outgoing normal vector at the fine scale domain boundary. It can be observed that  $\boldsymbol{\alpha}$  is the force per unit volume and  $\boldsymbol{\beta}$  is the stress. This dimensional analysis is important because there is an error in Ref. [5] which leads to inconsistencies.

In addition, it should be pointed out that  $\boldsymbol{\beta}$  is not deviatoric.

## 2.6 Coarse scale force per unit volume and stress

By zeroing out all variations except  $\delta\mathbf{V}^M$  in Eq. (2), one obtains

$$\mathbf{f}^M \cdot \delta\mathbf{V}^M = -\boldsymbol{\alpha} \cdot (-\delta\mathbf{V}^M) = \boldsymbol{\alpha} \cdot \delta\mathbf{V}^M, \forall \delta\mathbf{V}^M \in \mathbb{R}^d \Leftrightarrow \mathbf{f}^M = \boldsymbol{\alpha}.$$

This is consistent for the dimensional analysis as  $\boldsymbol{\alpha}$  is indeed the force per unit volume.

Similarly, it is possible to obtain

$$\begin{aligned} \boldsymbol{\sigma}^{M,dev} : \delta\mathbf{G}^M - p^M \text{tr}(\delta\mathbf{G}^M) &= -\boldsymbol{\beta} : (-\delta\mathbf{G}^M) = \boldsymbol{\beta} : \delta\mathbf{G}^M, \forall \delta\mathbf{G}^M \in \mathbb{R}^{d \times d}, \\ \Leftrightarrow \boldsymbol{\sigma}^{M,dev} &= \boldsymbol{\beta} + p^M \mathbf{I}, \end{aligned}$$

which is also consistent with the dimensional analysis. This expression shows again that  $\boldsymbol{\beta}$  is not deviatoric, so a small treatment is necessary in order to compute the deviatoric stress  $\boldsymbol{\sigma}^{M,dev}$ . The role of  $p^M$  is important in that regard, which distinguishes the present work from other approaches where the pressure field is treated as a homogenized variable instead of a Lagrange multiplier [9, 10].

To summarize:

- The coarse scale problem in Eq. (1) is similar to the well-known Navier-Stokes equations for unsteady incompressible Newtonian flows, except that it is written in terms of an unknown force per unit volume and an unknown deviatoric stress.
- Small obstacles are not represented directly within the coarse scale. Instead, they are embedded in fine scale domains, which are placed at each point of the coarse scale domain.
- For each fine scale domain, the boundary value problem in Eq. (3) should be solved. The boundary conditions for this problem come from averaging constraints relating the fine scale velocity field and its gradient to their coarse scale counterparts.
- The force per unit volume and deviatoric stress for the coarse scale are computed from the solution of the fine scale problem.

Because the fine scale problem in Eq. (3) is nonlinear, it is not possible to compute some kind of constant permeability tensor or tangent modulus beforehand and use it for running coarse scale simulations. An FExFE (FE<sup>2</sup>) strategy with two-way coupling between the scales should be implemented.

### 3 Numerical method

Under different names, computational homogenization, concurrent multiscale methods or FE<sup>2</sup> strategies are quite popular in solid mechanics. This means an FE mesh is used for the coarse scale and for each fine scale domain (multiple geometries may be used for the fine scale domains, for instance if the obstacles are not evenly distributed). In the present work, there are two main difficulties to tackle:

- Incompressible flows are considered. Thankfully, mixed weak forms have been introduced at both scales, so it is only a matter of using compatible discretization spaces.
- Because of the auto-advection term in Eq. (3), the boundary value problem for the fine scale is nonlinear and may be very hard to solve depending on the applied coarse scale velocity and velocity gradient.

#### 3.1 FE discretization

The Taylor-Hood P2/P1 pair is chosen to tackle the first difficulty. At both scales, quadratic interpolation is used for the velocity and linear interpolation for the pressure. This requires a special care for numerical integration, which is chosen to be of order 4, resulting in 6 integration points per triangle in two dimensions (2D), and 16 per tetrahedron in three dimensions (3D). This is of particular importance because it means there are 6 fine scale domains per element in 2D, and 16 in 3D. This is illustrated for the 2D case in Fig. 1.

To avoid introducing an additional level of complexity, no stabilization nor turbulence model is used. The two-way coupling algorithm to solve simultaneously the coarse and

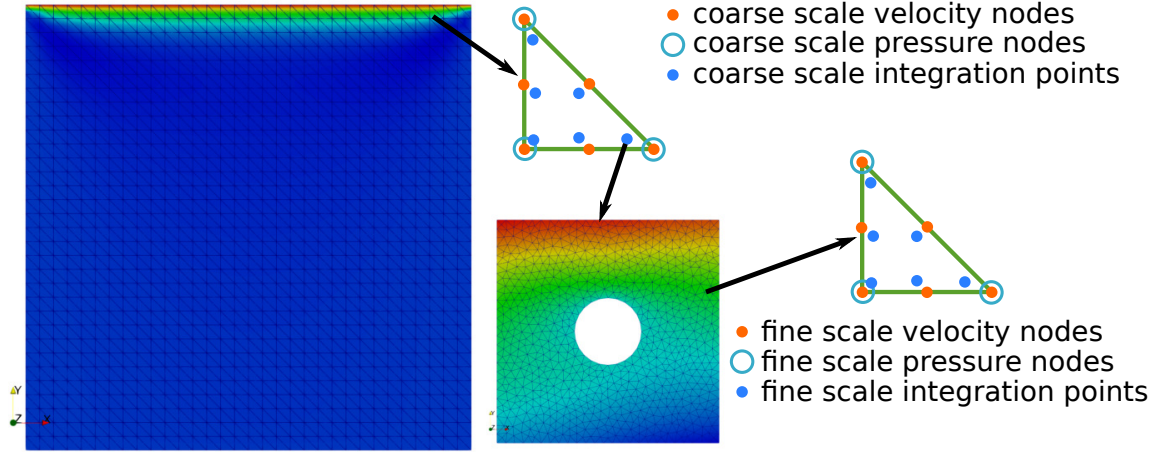


Figure 1:  $FE^2$  discretization with the Taylor-Hood P2/P1 pair showing the coarse scale mesh, one of the coarse scale mesh elements, one of the fine scale meshes and one of the fine scale mesh elements. In the example shown herein, there are 2,048 coarse scale mesh elements,  $2,048 \times 6 = 12,288$  coarse scale integration points and thus fine scale meshes,  $12,288 \times 1,524 = 18,726,912$  fine scale mesh elements and  $12,288 \times 3,172 \times 2 = 77,955,072$  fine scale velocity degrees of freedom.

fine scale problems is however quite evolved.

### 3.2 Fine scale problem

For a given fine scale domain, Eq. (3) may actually be considered independently from the coarse scale problem. At a given instant  $t^n$ , the solution at  $t^{n+1}$  can be computed for any imposed  $\mathbf{v}^M(\mathbf{x}, t^{n+1}) \in \mathbb{R}^d$  but regarding  $\nabla_{\mathbf{x}} \mathbf{v}^M(\mathbf{x}, t^{n+1}) \in \mathbb{R}^{d \times d}$  a pre-processing step is necessary to ensure that it is trace-less. Then, solving the fine scale problem requires finding the zeros of the following residuals:

$$\begin{aligned}
& \mathcal{R}_{\mathbf{v}^m}(\mathbf{v}^m, p^m, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta \mathbf{v}^m) = \\
& \frac{1}{|\Omega^m|} \int_{\Omega^m} \left[ \begin{aligned}
& \rho^m(\mathbf{y}) \left( \frac{\partial \mathbf{v}^m}{\partial t}(\mathbf{y}, t^{n+1}) + \mathbf{v}^m(\mathbf{y}, t^{n+1}) \cdot \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t^{n+1}) \right) \cdot \delta \mathbf{v}^m(\mathbf{y}) \\
& + 2\mu^m(\mathbf{y}) \nabla_{\mathbf{y}}^{S, dev} \mathbf{v}^m(\mathbf{y}, t^{n+1}) : \nabla_{\mathbf{y}} \delta \mathbf{v}^m(\mathbf{y}) \\
& - p^m(\mathbf{y}) \nabla_{\mathbf{y}} \cdot \delta \mathbf{v}^m(\mathbf{y}) \\
& - \boldsymbol{\alpha} \cdot \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \delta \mathbf{v}^m(\mathbf{y}) d\mathbf{y} \right) \\
& - \boldsymbol{\beta} : \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}} \delta \mathbf{v}^m(\mathbf{y}) d\mathbf{y} \right),
\end{aligned} \right] d\mathbf{y}
\end{aligned} \tag{4}$$

$$\mathcal{R}_{p^m}(\mathbf{v}^m, \delta p^m) = -\frac{1}{|\Omega^m|} \int_{\Omega^m} \delta p^m(\mathbf{y}) \nabla_{\mathbf{y}} \cdot \mathbf{v}^m(\mathbf{y}, t^{n+1}) d\mathbf{y},$$

$$\mathcal{R}_{\boldsymbol{\alpha}}(\mathbf{v}^m, \delta \boldsymbol{\alpha}) = -\delta \boldsymbol{\alpha} \cdot \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \mathbf{v}^m(\mathbf{y}, t^{n+1}) d\mathbf{y} - \mathbf{v}^M(\mathbf{x}, t^{n+1}) \right),$$

$$\mathcal{R}_{\boldsymbol{\beta}}(\mathbf{v}^m, \delta \boldsymbol{\beta}) = -\delta \boldsymbol{\beta} : \left( \frac{1}{|\Omega^m|} \int_{\Omega^m} \nabla_{\mathbf{y}} \mathbf{v}^m(\mathbf{y}, t^{n+1}) d\mathbf{y} - \nabla_{\mathbf{x}} \mathbf{v}^M(\mathbf{x}, t^{n+1}) \right),$$

where all solutions  $\mathbf{v}^m, p^m, \boldsymbol{\alpha}, \boldsymbol{\beta}$  are computed at time  $t^{n+1}$ . The Newton-Raphson scheme to find the zeros of these residuals starts from initial guesses  $\mathbf{v}_0^m, p_0^m, \boldsymbol{\alpha}_0, \boldsymbol{\beta}_0$ . A Newton-Raphson iteration  $j \rightarrow j+1$  requires solving the linear system

$$\left\{ \begin{aligned}
& \frac{\partial \mathcal{R}_{\mathbf{v}^m}}{\partial \mathbf{v}^m}(\mathbf{v}_j^m, p_j^m, \boldsymbol{\alpha}_j, \boldsymbol{\beta}_j, \delta \mathbf{v}^m) \Delta \mathbf{v}_{j+1}^m \\
& + \frac{\partial \mathcal{R}_{\mathbf{v}^m}}{\partial p^m}(\mathbf{v}_j^m, p_j^m, \boldsymbol{\alpha}_j, \boldsymbol{\beta}_j, \delta \mathbf{v}^m) \Delta p_{j+1}^m \\
& + \frac{\partial \mathcal{R}_{\mathbf{v}^m}}{\partial \boldsymbol{\alpha}}(\mathbf{v}_j^m, p_j^m, \boldsymbol{\alpha}_j, \boldsymbol{\beta}_j, \delta \mathbf{v}^m) \Delta \boldsymbol{\alpha}_{j+1} \\
& + \frac{\partial \mathcal{R}_{\mathbf{v}^m}}{\partial \boldsymbol{\beta}}(\mathbf{v}_j^m, p_j^m, \boldsymbol{\alpha}_j, \boldsymbol{\beta}_j, \delta \mathbf{v}^m) \Delta \boldsymbol{\beta}_{j+1}
\end{aligned} \right\} = -\mathcal{R}_{\mathbf{v}^m}(\mathbf{v}_j^m, p_j^m, \boldsymbol{\alpha}_j, \boldsymbol{\beta}_j, \delta \mathbf{v}^m),$$

$$\begin{aligned}
& \frac{\partial \mathcal{R}_{p^m}}{\partial p^m}(\mathbf{v}_j^m, \delta p^m) \Delta p_{j+1}^m = -\mathcal{R}_{p^m}(\mathbf{v}_j^m, \delta p_j^m), \\
& \frac{\partial \mathcal{R}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}(\mathbf{v}_j^m, \delta \boldsymbol{\alpha}) \Delta \boldsymbol{\alpha}_{j+1}^m = -\mathcal{R}_{\boldsymbol{\alpha}}(\mathbf{v}_j^m, \delta \boldsymbol{\alpha}_j), \\
& \frac{\partial \mathcal{R}_{\boldsymbol{\beta}}}{\partial \boldsymbol{\beta}}(\mathbf{v}_j^m, \delta \boldsymbol{\beta}) \Delta \boldsymbol{\beta}_{j+1}^m = -\mathcal{R}_{\boldsymbol{\beta}}(\mathbf{v}_j^m, \delta \boldsymbol{\beta}_j),
\end{aligned}$$

for the unknowns

$$\Delta \mathbf{v}_{j+1}^m = \mathbf{v}_{j+1}^m - \mathbf{v}_j^m, \Delta p_{j+1}^m = p_{j+1}^m - p_j^m, \Delta \boldsymbol{\alpha}_{j+1} = \boldsymbol{\alpha}_{j+1} - \boldsymbol{\alpha}_j, \Delta \boldsymbol{\beta}_{j+1} = \boldsymbol{\beta}_{j+1} - \boldsymbol{\beta}_j.$$

There is no particular difficulty in computing those derivatives since all terms are linear except the auto-advection term. It is linearized using

$$\frac{\partial (\mathbf{v}^m \cdot \nabla \mathbf{v}^m)}{\partial \mathbf{v}^m}(\mathbf{v}_j^m) \Delta \mathbf{v}_{j+1}^m = \Delta \mathbf{v}_{j+1}^m \cdot \nabla \mathbf{v}_j^m + \mathbf{v}_j^m \cdot \nabla \Delta \mathbf{v}_{j+1}^m.$$

After FE discretization, the linear system has the following form:

$$\begin{pmatrix} \mathbf{A}_{\mathbf{v}^m \mathbf{v}^m} & \mathbf{A}_{\mathbf{v}^m p^m} & \mathbf{A}_{\mathbf{v}^m \boldsymbol{\alpha}} & \mathbf{A}_{\mathbf{v}^m \boldsymbol{\beta}} \\ \mathbf{A}_{\mathbf{v}^m p^m}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{\mathbf{v}^m \boldsymbol{\alpha}}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{\mathbf{v}^m \boldsymbol{\beta}}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{U}_{\mathbf{v}^m} \\ \Delta \mathbf{U}_{p^m} \\ \Delta \mathbf{U}_{\boldsymbol{\alpha}} \\ \Delta \mathbf{U}_{\boldsymbol{\beta}} \end{pmatrix} = \begin{pmatrix} \mathbf{B}_{\mathbf{v}^m} \\ \mathbf{B}_{p^m} \\ \mathbf{B}_{\boldsymbol{\alpha}} \\ \mathbf{B}_{\boldsymbol{\beta}} \end{pmatrix} \quad (5)$$

where  $\Delta \mathbf{U}_{\mathbf{v}^m}$  is a vector whose size is the number of P2 nodes of the FE mesh times  $d$ ,  $\Delta \mathbf{U}_{p^m}$  is a vector whose size is the number of P1 nodes of the FE mesh,  $\Delta \mathbf{U}_{\boldsymbol{\alpha}}$  is a vector whose size is  $d$ , and  $\Delta \mathbf{U}_{\boldsymbol{\beta}}$  is a vector whose size is  $d \times d$ .

Rows and columns of the linear system in Eq. (5) that are associated to nodes on  $\Gamma_{\mathcal{O}}^m$  are modified to set the velocity to zero at those nodes.

Although it is symmetric, the linear problem in Eq. (5) is poorly conditioned because of the presence of zeros on the diagonal. This is typical of Taylor-Hood FEs, but it is aggravated here due to the additional Lagrange multipliers for the kinematic constraints. In particular,  $\mathbf{A}_{\mathbf{v}^m \boldsymbol{\alpha}}$  and  $\mathbf{A}_{\mathbf{v}^m \boldsymbol{\beta}}$  are dense matrices. A direct LU solver is used throughout this work to solve this problem.

To summarize, at each time increment, a new coarse scale velocity and velocity gradient are imposed. The fine scale problem is solved using a Newton-Raphson scheme to deal with the nonlinearity of the auto-advection term. Each iteration of the Newton-Raphson scheme requires solving a linear problem of the form given in Eq. (5), which is achieved using a direct solver.

Any time discretization can be used for the  $\frac{\partial \mathbf{v}^m}{\partial t}$  term. The backward differentiation formula of order 2 is used in this work.

### 3.3 Coarse scale problem

The weak form of the coarse scale problem has already been given in Eq. (1). At a given instant  $t^n$ , the solution at  $t^{n+1}$  can be computed for any imposed  $\mathbf{v}_D^M$  and  $\mathbf{t}_N^M$ . It requires finding the zeros of the following residuals:

$$\mathcal{R}_{\mathbf{v}^M}(\mathbf{v}^M, p^M) = \begin{cases} \int_{\Omega^M} (\mathbf{f}^M(\mathbf{x}, t^{n+1}) \cdot \delta \mathbf{v}^M(\mathbf{x}) + \boldsymbol{\sigma}^{M, dev}(\mathbf{x}, t^{n+1}) : \nabla_{\mathbf{x}} \delta \mathbf{v}^M(\mathbf{x})) \, d\mathbf{x} \\ - \int_{\Omega^M} p^M(\mathbf{x}) \nabla_{\mathbf{x}} \cdot \delta \mathbf{v}^M(\mathbf{x}) \, d\mathbf{x} - \int_{\Gamma_N^M} \mathbf{t}_N^M(\mathbf{x}, t^{n+1}) \cdot \delta \mathbf{v}^M(\mathbf{x}) \, d\mathbf{x}, \end{cases}$$

$$\mathcal{R}_{p^M}(\mathbf{v}^M) = - \int_{\Omega^M} \delta p^M(\mathbf{x}) \nabla_{\mathbf{x}} \cdot \mathbf{v}^M(\mathbf{x}) d\mathbf{x},$$

where all solutions  $\mathbf{v}^M, p^M$  are computed at time  $t^{n+1}$ . The Newton-Raphson scheme to find the zeros of these residuals starts from initial guesses  $\mathbf{v}_0^M, p_0^M$ . A Newton-Raphson iteration  $i \rightarrow i+1$  requires solving the linear system

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{R}_{\mathbf{v}^M}}{\partial \mathbf{v}^M}(\mathbf{v}_i^M, p_i^M, \delta \mathbf{v}^M) \Delta \mathbf{v}_{i+1}^M \\ + \frac{\partial \mathcal{R}_{\mathbf{v}^M}}{\partial p^M}(\mathbf{v}_i^M, p_i^M, \delta \mathbf{v}^M) \Delta p_{i+1}^M \end{array} \right\} = -\mathcal{R}_{\mathbf{v}^M}(\mathbf{v}_i^M, p_i^M, \delta \mathbf{v}^M),$$

$$\frac{\partial \mathcal{R}_{p^M}}{\partial \mathbf{v}^M}(\mathbf{v}_i^M, \delta p^M) \Delta \mathbf{v}_{i+1}^M = -\mathcal{R}_{p^M}(\mathbf{v}_i^M, \delta p_i^M),$$

for the unknowns

$$\Delta \mathbf{v}_{i+1}^M = \mathbf{v}_{i+1}^M - \mathbf{v}_i^M, \Delta p_{i+1}^M = p_{i+1}^M - p_i^M.$$

The derivatives are not as trivial as for the fine scale:

$$\frac{\partial \mathcal{R}_{\mathbf{v}^M}}{\partial \mathbf{v}^M}(\mathbf{v}_i^M, p_i^M, \delta \mathbf{v}^M) \Delta \mathbf{v}_{i+1}^M = \left\{ \begin{array}{l} \int_{\Omega^M} \left( \frac{\partial \mathbf{f}^M}{\partial \mathbf{v}^M}(\mathbf{x}, t^{n+1}) \cdot \Delta \mathbf{v}_{i+1}^M(\mathbf{x}, t^{n+1}) \right) \cdot \delta \mathbf{v}^M(\mathbf{x}) d\mathbf{x} \\ + \int_{\Omega^M} \left( \frac{\partial \boldsymbol{\sigma}^{M,dev}}{\partial \nabla_{\mathbf{x}} \mathbf{v}^M}(\mathbf{x}, t^{n+1}) : \nabla_{\mathbf{x}} \Delta \mathbf{v}^M(\mathbf{x}, t^{n+1}) \right) : \nabla_{\mathbf{x}} \delta \mathbf{v}^M(\mathbf{x}) d\mathbf{x}, \end{array} \right.$$

$$\frac{\partial \mathcal{R}_{\mathbf{v}^M}}{\partial p^M}(\mathbf{v}_i^M, p_i^M, \delta \mathbf{v}^M) \Delta p_{i+1}^M = - \int_{\Omega^M} \Delta p^M(\mathbf{x}) \nabla_{\mathbf{x}} \cdot \delta \mathbf{v}^M(\mathbf{x}) d\mathbf{x},$$

$$\frac{\partial \mathcal{R}_{p^M}}{\partial \mathbf{v}^M}(\mathbf{v}_i^M, \delta p^M) \Delta \mathbf{v}_{i+1}^M = - \int_{\Omega^M} \delta p^M(\mathbf{x}) \nabla_{\mathbf{x}} \cdot \Delta \mathbf{v}^M(\mathbf{x}) d\mathbf{x}.$$

The two terms that are difficult to compute are  $\frac{\partial \mathbf{f}^M}{\partial \mathbf{v}^M}(\mathbf{x}, t^{n+1})$  and  $\frac{\partial \boldsymbol{\sigma}^{M,dev}}{\partial \nabla_{\mathbf{x}} \mathbf{v}^M}(\mathbf{x}, t^{n+1})$ . In fact, both  $\mathbf{f}^M$  and  $\boldsymbol{\sigma}^{M,dev}$  already require some implementation details. Concretely, at each iteration  $i \rightarrow i+1$ , it is necessary to compute the derivatives once per integration point (and hence per fine scale domain) and the residuals should be computed multiple times depending on whether a line search procedure is used.

Computing the residuals requires solving the fine scale problem once per fine scale domain by imposing  $\mathbf{v}_i^M(\mathbf{x}, t^{n+1})$  and  $\nabla_{\mathbf{x}}^{dev} \mathbf{v}_i^M(\mathbf{x}, t^{n+1})$ . From those fine scale solutions, it is important to collect  $\boldsymbol{\alpha} = \mathbf{f}^M(\mathbf{x}, t^{n+1})$  and  $\boldsymbol{\beta}^{dev} = \boldsymbol{\sigma}^{M,dev}(\mathbf{x}, t^{n+1})$ . It is then straightforward to compute the coarse scale residuals.

The derivatives  $\frac{\partial \mathbf{f}^M}{\partial \mathbf{v}^M}(\mathbf{x}, t^{n+1})$  and  $\frac{\partial \boldsymbol{\sigma}^{M,dev}}{\partial \nabla_{\mathbf{x}} \mathbf{v}^M}(\mathbf{x}, t^{n+1})$  can be computed by inverting the first line in Eq. (5) and injecting it in the third and fourth lines. This method has not been chosen in this work because it might not always be possible to invert the first line

separately, for instance in the steady case. Automatic differentiation has been preferred, although it requires solving the fine scale problem multiple times, each with a different small fluctuation added to  $\mathbf{v}_i^M(\mathbf{x}, t^{n+1})$  and  $\nabla_{\mathbf{x}}^{dev} \mathbf{v}_i^M(\mathbf{x}, t^{n+1})$ .

The linear system to solve after FE discretization is no different from the standard Taylor-Hood saddle point problem:

$$\begin{pmatrix} \mathbf{A}_{\mathbf{v}^M \mathbf{v}^M} & \mathbf{A}_{\mathbf{v}^M p^M} \\ \mathbf{A}_{\mathbf{v}^M p^M}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{U}_{\mathbf{v}^M} \\ \Delta \mathbf{U}_{p^M} \end{pmatrix} = \begin{pmatrix} \mathbf{B}_{\mathbf{v}^M} \\ \mathbf{B}_{p^M} \end{pmatrix}. \quad (6)$$

In fact, once the residuals and their derivatives are computed, there is no specific difficulty in assembling and solving this linear system as compared to standard Taylor-Hood FEs. This is also true for setting up the Dirichlet boundary conditions.

### 3.4 Two-way coupling

The coupling algorithm stems naturally from the coarse scale Newton-Raphson algorithm. Both coarse and fine scale solutions at instant  $t^{n+1}$  are simultaneously computed using the following process:

1. Setup initial guesses  $\mathbf{v}_0^M, p_0^M$  for the coarse scale Newton-Raphson algorithm from the last computed solutions
2. For each coarse scale Newton-Raphson algorithm iteration  $i \rightarrow i + 1$ 
  - (a) For each fine scale domain
    - i. Setup initial guesses  $\mathbf{v}_0^m, p_0^m, \boldsymbol{\alpha}_0, \boldsymbol{\beta}_0$  for the fine scale Newton-Raphson algorithm from the last computed solutions.
    - ii. For each fine scale Newton-Raphson algorithm iteration  $j \rightarrow j + 1$ 
      - A. Compute the fine scale residuals and their derivatives.
      - B. Assemble and solve the linear system of the form given in Eq. (5).
      - C. Test for convergence and go back to 2(a)ii if necessary.
    - iii. Get the converged  $\boldsymbol{\alpha} = \mathbf{f}^M(\mathbf{x}, t^{n+1})$  and  $\boldsymbol{\beta}^{dev} = \boldsymbol{\sigma}^{M,dev}(\mathbf{x}, t^{n+1})$ .
  - (b) Compute the coarse scale residuals and their derivatives.
  - (c) Assemble and solve the linear system of the form given in Eq. (6).

- (d) Test for convergence and go back to 2 if necessary.
- (e) Get the converged coarse scale velocity-pressure solutions.

In practice, advanced procedures such as line search or sub-stepping might be added to improve the convergence of this algorithm, especially regarding the solution of the fine scale problems.

Finally, the multiscale  $FE^2$  algorithm presented in this section has been implemented in the *FEMS* software [11]. It is expected to require a significant amount of computational resources, both regarding computation time and memory consumption. For some problems, the discretization of the fine scale domains might not even fit in the memory of a recent workstation.

To avoid such issue, the coarse scale mesh is partitioned among several processes using the *Message Passing Interface (MPI)* standard and the *METIS* mesh partitioning library [12]. Each coarse mesh element and its integration points belong to a single partition, while communications for coarse mesh nodes shared by different partitions are handled by *FEMS*.

Regarding the solution of linear problems, it is performed at the coarse scale using either the Schur complement solver available in the *PETSc* suite [13] or the *MUMPS* sparse direct solver [14], which is also available through the *PETSc* suite. Since each coarse scale integration point belongs to a single partition, this is also true for each fine scale domain. Therefore, the fine scale domains are not partitioned and the solution of fine scale problems is completely sequential. They are performed using *UMFPACK*'s direct solver [15].

## 4 Numerical results

To keep the computational cost reasonable, only 2D simulations are considered in this section. All multiscale simulations use 2D square fine scale domains of size  $l$ . For solving linear problems at the coarse scale, the Schur complement solver is used for the first problem presented in Subsec. 4.1, and the direct solver for the second problem presented

in Subsec. 4.2.

## 4.1 Lid-driven cavity

The well-known lid-driven cavity problem is considered for the coarse scale. The domain is the square of area  $1 \text{ m}^2$  shown in Fig. 2(a), discretized with a reference mesh size of  $0.03125 \text{ m}$ . Unless otherwise mentioned, the fluid mass density is fixed to  $1 \text{ kg m}^{-3}$ , gravity is neglected and the fluid dynamic viscosity is fixed to  $0.2 \text{ Pa s}$  so that the Reynolds number is equal to 5. The problem is solved with a time step of  $\Delta t = 1 \text{ s}$  and the steady state is considered as reached when the  $L^2$  norm of the relative velocity change between two time increments is below  $10^{-6}$ .

Two reference fine scale domains are considered hereafter. The first one is homogeneous, has a size  $l = 0.08 \text{ m}$  and is discretized with a mesh size of  $2.5 \text{ mm}$ , as shown in Fig. 2(b). The second one has the same size but has a circular obstacle of radius  $R = 0.012 \text{ m}$  placed at its center, and is discretized with a mesh size of  $3.2 \text{ mm}$ , as shown in Fig. 2(c).

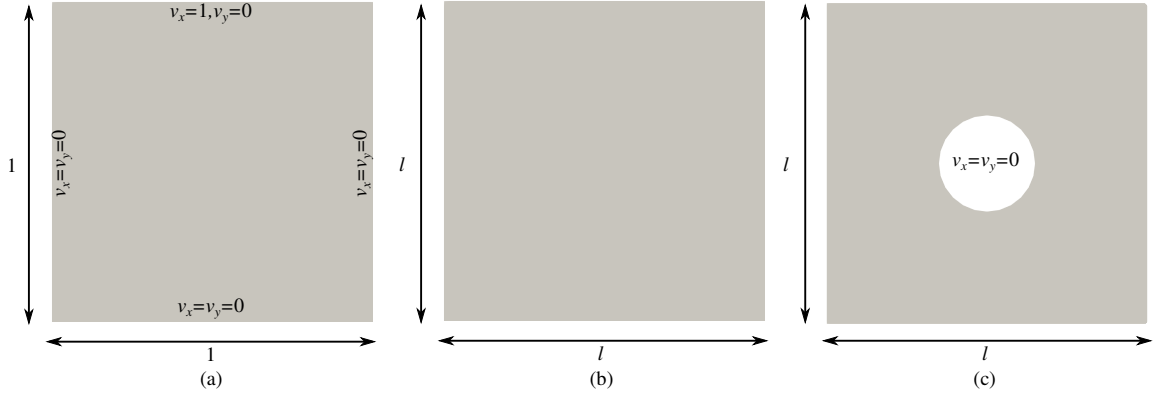


Figure 2: Domains and boundary conditions used for the lid-driven cavity problem: (a) coarse scale domain, (b) homogeneous fine scale domain, (c) holed fine scale domain. Lengths are in meters and velocities in meters per second.

### 4.1.1 Influence of mesh size

To verify the robustness of the principle of multiscale virtual power in Eq. (2), the lid-driven cavity problem is solved using a multiscale model with the homogeneous fine scale domain. Because there is no obstacle at the fine scale, the solution should be equivalent

to that obtained using a single-scale model.

In addition to the reference multiscale model, both the coarse and fine mesh sizes are doubled to generate an intermediate discretization, and they are doubled again to generate a coarse discretization. The comparison between the three results is shown in Fig. 3.

Clearly, for this low Reynolds number of 5, there is no need to use an excessively refined mesh, especially since a quadratic interpolation is used for the velocity.

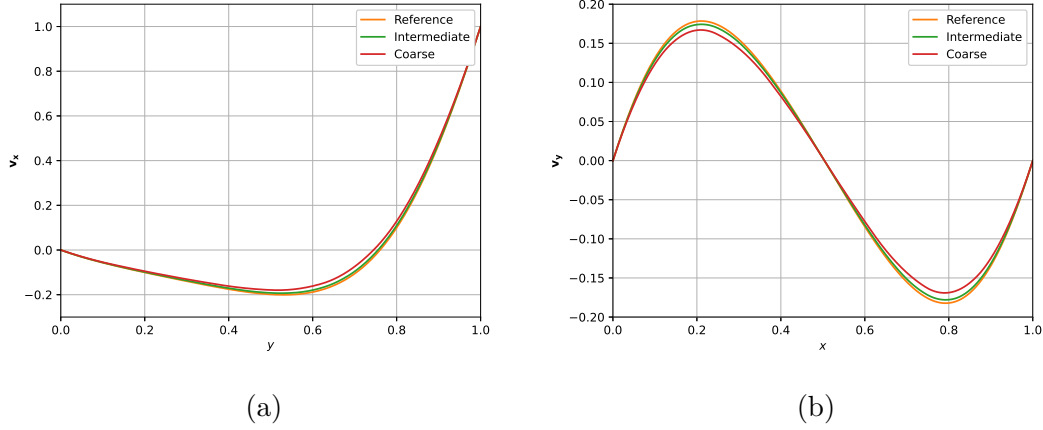


Figure 3: Velocity curves at steady state using different coarse and fine meshes for the lid-driven cavity problem with homogeneous fine scale: (a)  $v_x^M$  along the vertical line going through the center of the cavity, (b)  $v_y^M$  along the horizontal line going through the center of the cavity. Lengths are in meters and velocities in meters per second.

#### 4.1.2 Influence of fine scale domain size

In addition to the reference model, two simulations are run; the one with a fine scale domain size  $l = 0.04$  m and the other with  $l = 0.16$  m. Note that the fine scale mesh size is modified proportionally to  $l$  so that the number of elements remains constant. The three results are compared in Fig. 4.

The size effect is clearly negligible when all fine scale domains are homogeneous. This should be investigated again with the second reference fine scale domain to see the effect of the obstacle.

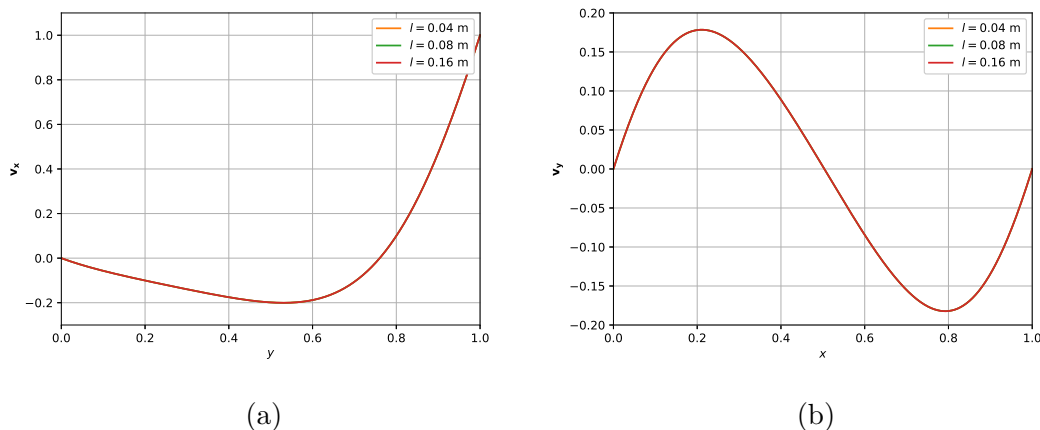


Figure 4: Velocity curves at steady state using different fine scale domain sizes  $l$  for the lid-driven cavity problem with homogeneous fine scale: (a)  $\mathbf{v}_x^M$  along the vertical line going through the center of the cavity, (b)  $\mathbf{v}_y^M$  along the horizontal line going through the center of the cavity. Lengths are in meters and velocities in meters per second.

#### 4.1.3 Comparison with single-scale models: influence of Reynolds number

Finally, to validate the multiscale model, a comparison with two single-scale simulations at different Reynolds numbers is considered. The first one solves the Navier-Stokes equations for Newtonian incompressible flow using the Taylor-Hood P2/P1 pair directly at the coarse scale, while the second one adds stabilization terms based on the Residual-Based Variational MultiScale (RBVMS) formulation [9], which has been implemented in FEMS in a previous work [16].

Simulations are run with a different dynamic viscosity in order to obtain a Reynolds number of 5 and 50. All results are shown in Fig. 5.

The curves coincide, which suggests that for these low Reynolds numbers the effect of the RBVMS formulation is negligible, and also that the multiscale model agrees well with single-scale simulations.

#### 4.1.4 With obstacles: influence of fine scale domain size

The reference fine scale domain with an obstacle is now considered. Similarly to Par. 4.1.1, a prior analysis (not reported herein) showed that the reference mesh size of 3.2 mm is fine enough for the fine scale domain. Three simulations are run by varying the fine

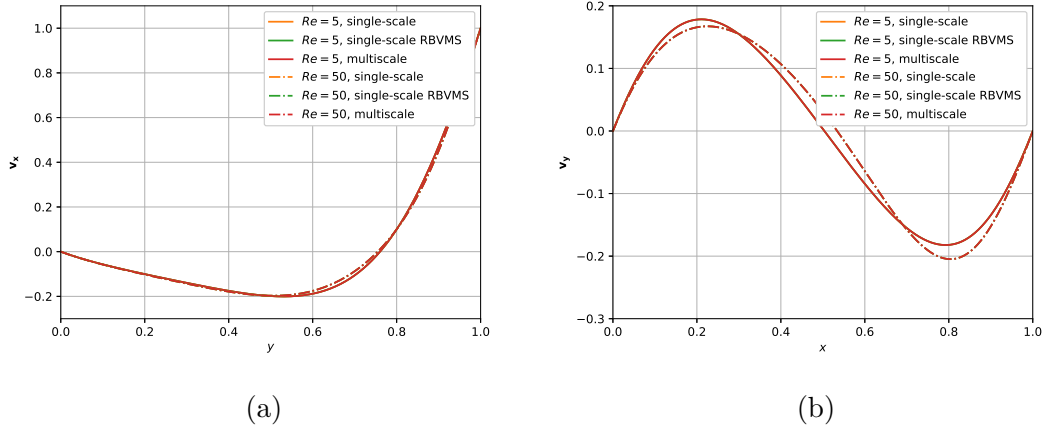


Figure 5: Velocity curves at steady state compared with single-scale simulations at different Reynolds numbers for the lid-driven cavity problem with homogeneous fine scale: (a)  $\mathbf{v}_x^M$  along the vertical line going through the center of the cavity, (b)  $\mathbf{v}_y^M$  along the horizontal line going through the center of the cavity. Lengths are in meters and velocities in meters per second.

scale domain size following the same procedure as in Par. 4.1.2. Note that this means that the obstacle size also changes, while the obstacle area fraction relative to the fine scale domain area remains constant. The three results are compared in Fig. 6.

Although it has not been observed for the simulations with homogeneous fine scale domain, it is clear that the fine scale domain size has an effect when there is an obstacle at the fine scale. Increasing the length  $l$  has the consequence of smoothing the velocity field, as the boundary layer in Fig. 6(a) becomes more pronounced when  $l$  is decreased. Additionally, as shown in Fig. 6(b), the fluid motion imposed at the top penetrates deeper in the cavity when  $l$  is increased.

#### 4.1.5 With obstacles: influence of fine scale obstacle area fraction

In previous simulations, the fine scale obstacle area fraction has been kept constant. It is interesting to investigate also the effect of this fraction on the results, at a constant  $l$ . Two additional fine scale domains are generated. All parameters are kept identical to the reference domain with obstacle, but the obstacle radius is changed to  $R \approx 0.0085$  m to have half the obstacle area fraction, and  $R \approx 0.0170$  m to have twice the obstacle area

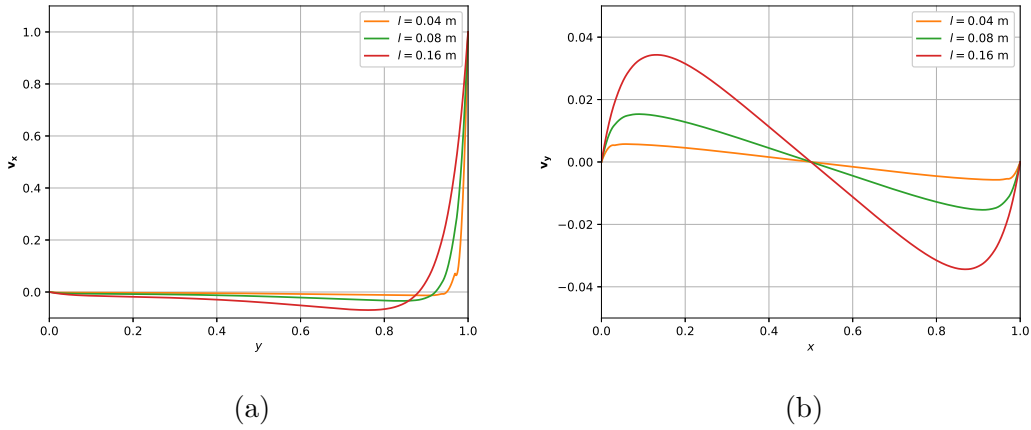


Figure 6: Velocity curves at steady state using different fine scale domain sizes  $l$  for the lid-driven cavity problem with an obstacle at the fine scale: (a)  $\mathbf{v}_x^M$  along the vertical line going through the center of the cavity, (b)  $\mathbf{v}_y^M$  along the horizontal line going through the center of the cavity. Lengths are in meters and velocities in meters per second.

fraction. The comparison is shown in Fig. 7.

Once again, the multiscale approach predicts a size effect. When the obstacle area fraction is low, the fluid motion penetrates deep in the cavity. The result is then similar to the no obstacle case shown in Fig. 4(b). On the opposite, with a higher obstacle area fraction, the space for the fluid flow becomes smaller, resulting in a slower flow deep in the cavity, as shown in Fig. 7(b).

#### 4.1.6 With obstacles: comparison with single-scale models

As in Par. 4.1.3, a multiscale simulation with a circular obstacle of radius  $R = 0.015$  m and a fine scale domain size  $l = 0.1$  m is compared with a single-scale model with stabilization. Note that for this problem, a single-scale simulation without stabilization was also conducted but it did not converge. The difference with Par. 4.1.3 is that there are obstacles that should be modeled and discretized directly at the coarse scale in the single-scale simulations, as shown in Fig. 8. The mesh size for these single-scale simulations is identical to that used at the fine scale in the multiscale model, *i.e.*, 4 mm. The obstacle size is intentionally quite large (one tenth) compared to the coarse scale domain size in order to keep the computational cost of the single-scale simulations reasonable.

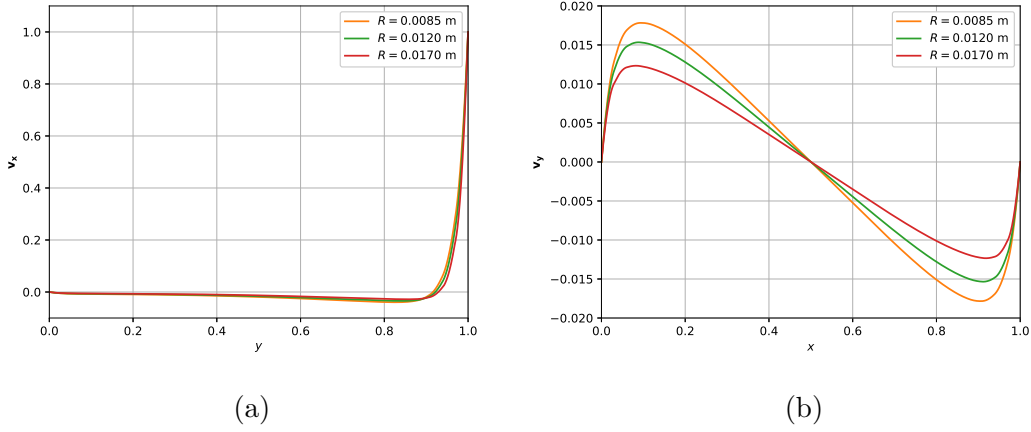


Figure 7: Velocity curves at steady state using different fine scale obstacle sizes for the lid-driven cavity problem with an obstacle at the fine scale: (a)  $\mathbf{v}_x^M$  along the vertical line going through the center of the cavity, (b)  $\mathbf{v}_y^M$  along the horizontal line going through the center of the cavity. Lengths are in meters and velocities in meters per second.

The comparison between the single-scale simulation and the multiscale model is shown in Fig. 9. The coarse scale velocity field is relatively smooth because of scale separation. The fluctuation due to the obstacle is captured at the fine scale. Overall, the isocontours match well between the multiscale simulation and the single-scale simulation.

The velocity curves are shown in Fig. 10. We can see that the multiscale model's prediction is accurate but only in an average sense, with the size of the averaging window being defined by the fine scale domain size  $l$ . The error relative to the single-scale model is large in this case because  $l$  is only one tenth of the coarse scale domain size. An assessment of the evolution of this error with respect to  $l$  is proposed in Par. 4.2.1.

In the multiscale model, there are 2,048 coarse scale mesh elements,  $2,048 \times 6 = 12,288$  coarse scale integration points and thus fine scale meshes,  $12,288 \times 1,524 = 18,726,912$  fine scale mesh elements and  $12,288 \times 3,172 \times 2 = 77,955,072$  fine scale velocity degrees of freedom. In comparison, there are only 159,164 elements and  $328,627 \times 2 = 657,254$  velocity degrees of freedom in the single-scale model. The multiscale model is clearly not relevant when the obstacle size is large.

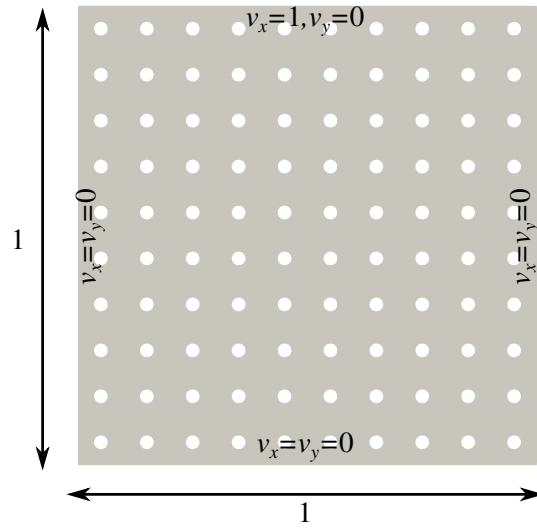


Figure 8: Domain and boundary conditions used for the single-scale lid-driven cavity problem with obstacles. Lengths are in meters and velocities in meters per second.

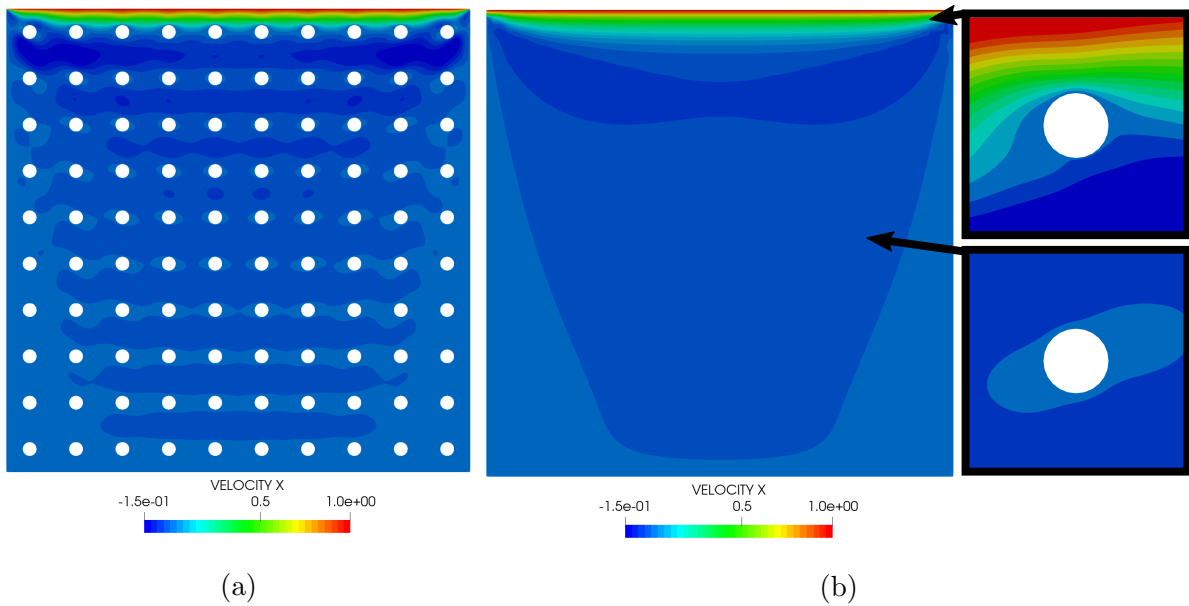


Figure 9: Velocity component  $v_x$  (m s<sup>-1</sup>) at steady state compared with a single-scale simulation with stabilization for the lid-driven cavity problem with an obstacle at the fine scale: (a) single-scale simulation with stabilization, (b) multiscale simulation with fine scale domains at two different locations (fine scale domains are enlarged for the visualization).

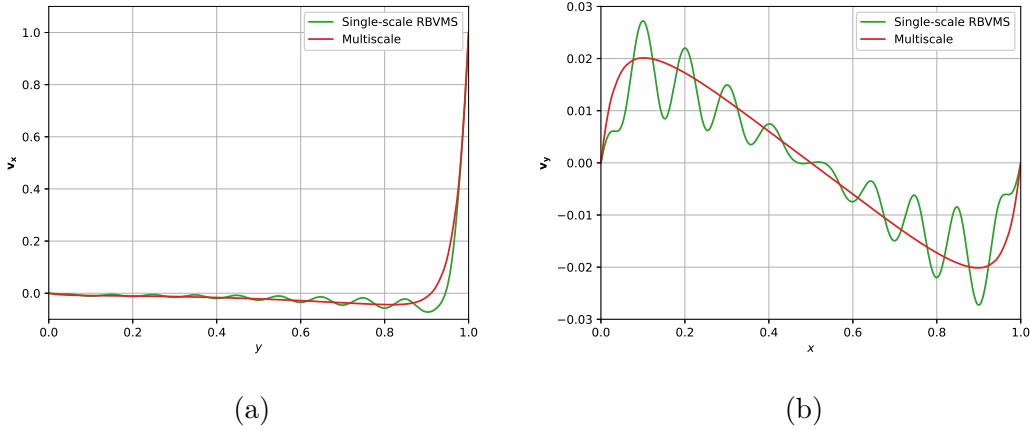


Figure 10: Velocity curves at steady state compared with a single-scale simulation for the lid-driven cavity problem with an obstacle at the fine scale: (a)  $\mathbf{v}_x^M$  along the vertical line going through the center of the cavity, (b)  $\mathbf{v}_y^M$  along the horizontal line going through the center of the cavity. Lengths are in meters and velocities in meters per second.

As a summary, these first calculations on the lid-driven cavity problem show that:

- The multiscale model and its implementation are consistent and robust when the fine scale domain is homogeneous. In this case, the fine scale domain size has a negligible influence.
- When the fine scale domain includes an obstacle, the effects of the obstacle area fraction and the fine scale domain size are well predicted by the model.
- The multiscale model agrees well with single-scale simulations, but only in an average sense.

For the Reynolds numbers considered herein, the lid-driven cavity problem converges to a steady state. Although it is modeled with transient solvers, unsteady flow has not been addressed yet.

## 4.2 L-shaped domain with obstacles

We now consider unsteady flow through an L-shaped channel at the coarse scale. The domain is still the square of area  $1 \text{ m}^2$ , but it is composed of a dark phase and a light

phase as shown in Fig. 11(a). The domain is discretized with a reference mesh size of 0.04 m.

The traction vector is set to  $\mathbf{t}_N^M(\mathbf{x}, t) = (\sin(2\pi t), 0)$  to obtain a time-varying pressure at the left boundary, and the top and bottom boundaries are defined as non-penetrating walls. Unless otherwise mentioned, the fluid mass density is fixed to  $1 \text{ kg m}^{-3}$ , gravity is neglected and the fluid dynamic viscosity is fixed to  $1 \text{ Pa s}$ . The problem is solved with a time step of  $\Delta t = 0.03 \text{ s}$  for 100 time steps.

The reference fine scale domain for the dark phase is the same as that in Figure 2(c), except that it has a size  $l = 0.1 \text{ m}$  and an obstacle radius  $R = 0.015 \text{ m}$ . It is discretized with a mesh size of 4 mm, as shown in Fig. 11(b). The obstacle size is once again quite large (one tenth) compared to the coarse scale domain size in order to keep the computational cost of the single-scale simulations reasonable. The reference fine scale domain for the light phase is the same as that in Figure 2(b), except that it has a size  $l = 0.1 \text{ m}$  and is discretized with a mesh size of 0.1 m. This is possible because we showed in Pars. 4.1.2 and 4.1.1 that for a homogeneous fine scale domain both  $l$  and the mesh size have a negligible influence on the computational results. A single-scale model could, in fact, be used for this part of the coarse scale domain, but this is not possible in our implementation.

#### 4.2.1 Comparison with single-scale models: influence of fine scale domain size

To assess the performance of the multiscale approach for increasing numbers of obstacles, we introduce for the dark phase two fine scale domain sizes  $l = 0.05 \text{ m}$  and  $l = 0.025 \text{ m}$  in addition to the reference one. We use the multiscale model and also the single-scale models shown in Figure 12. Near obstacles in these single-scale models, the mesh size is set to 4 mm for  $l = 0.1 \text{ m}$ , 2 mm for  $l = 0.05 \text{ m}$ , and 1 mm for  $l = 0.025 \text{ m}$ . Away from obstacles, it is set to 0.04 m. These values are chosen so that the multiscale models and the single-scale models are comparable.

We end up with 48,914 elements and  $99,662 \times 2 = 199,324$  velocity degrees of freedom for  $l = 0.1 \text{ m}$  (Figure 12(a)), 177,528 elements and  $359,649 \times 2 = 719,298$  velocity degrees of freedom for  $l = 0.05 \text{ m}$  (Figure 12(b)) and 673,914 elements and

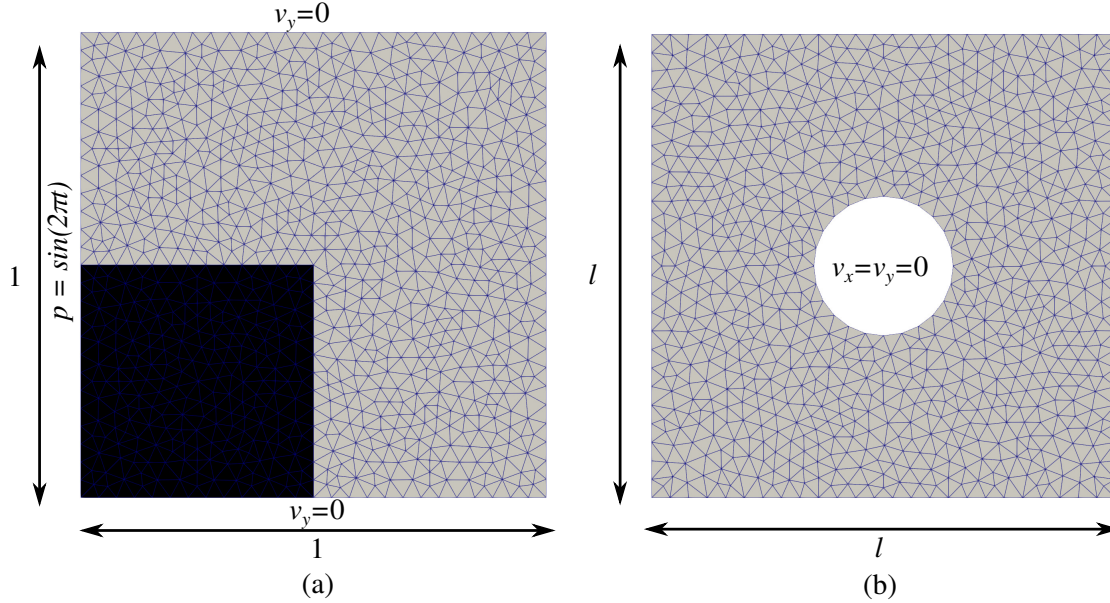


Figure 11: Domains, boundary conditions and reference discretizations used for the L-shaped domain problem: (a) coarse scale domain, (b) holed fine scale domain. Lengths are in meters, velocities in meters per second and pressures in Pa.

$1,359,871 \times 2 = 2,719,742$  velocity degrees of freedom for  $l = 0.025$  m (Figure 12(c)). For the dark phase in the multiscale model, for all values of  $l$ , there are 452 coarse scale mesh elements,  $452 \times 6 = 2,712$  coarse scale integration points and thus fine scale meshes,  $2,712 \times 1,524 = 4,133,088$  fine scale mesh elements and  $2,712 \times 3,172 \times 2 = 20,133,888$  fine scale velocity degrees of freedom. For the light phase in the multiscale model, for all values of  $l$ , there are 1,298 coarse scale mesh elements,  $1,298 \times 6 = 7,788$  coarse scale integration points and thus fine scale meshes,  $7,788 \times 2 = 15,576$  fine scale mesh elements and  $7,788 \times 9 \times 2 = 140,184$  fine scale velocity degrees of freedom. This amounts to a total of 4,148,664 fine scale mesh elements and 20,274,072 fine scale velocity degrees of freedom.

On the one hand, although the single-scale model still involves less degrees of freedom than the multiscale model, we can see that its computational cost would eventually blow up with small obstacle sizes, as in 2D the number of degrees of freedom is multiplied by nearly four when the fine scale domain size is divided by two. On the other hand, the number of degrees of freedom in the multiscale model does not depend on the obstacle size. From this analysis, we can estimate that in 2D the multiscale model would become

interesting from the computational cost viewpoint for obstacles at least a hundred times smaller than the coarse scale domain size. The resin transfer molding process, for instance, is a process used to manufacture fiber-reinforced polymer composite parts. The resin is injected progressively into the mold and permeates through the fiber reinforcement. This process involves molds of a few meters and fibers of a few microns, the ratio is hence around one million.

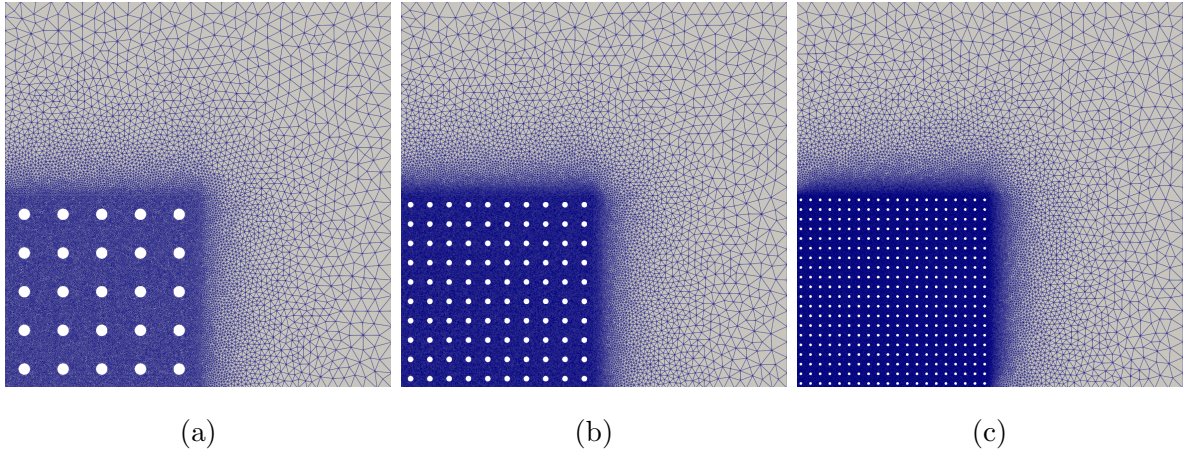


Figure 12: Domains and reference discretizations used for the single-scale models of the L-shaped domain problem: (a)  $l = 0.1$  m, (b)  $l = 0.05$  m, (c)  $l = 0.025$  m.

The velocity field at  $t = 0.36$  s is shown in Figure 13. We can see that the dark phase acts as a coarse obstacle, which is actually composed of a large number of fine obstacles. Similarly as for the lid-driven cavity problem, we observe a slight decrease of the flow velocity when the obstacle size is decreased, as the size of the high velocity region at the top of the domain is smaller in Figures 13(c,f) as compared to 13(a,d).

The single-scale and the multiscale simulation results match well with each other based on those figures. This can also be observed for the pressure fields shown in Figure 14. Note that these fields have been normalized so that the integral of the coarse scale pressure field over the coarse scale domain is equal to zero. We can see in Figure 14(c) that the small fluctuations of the pressure field due to the obstacles is less visible at the coarse scale when the obstacles are smaller. As a result, the pressure fields in Figures 14(c) and 14(f) match very well.

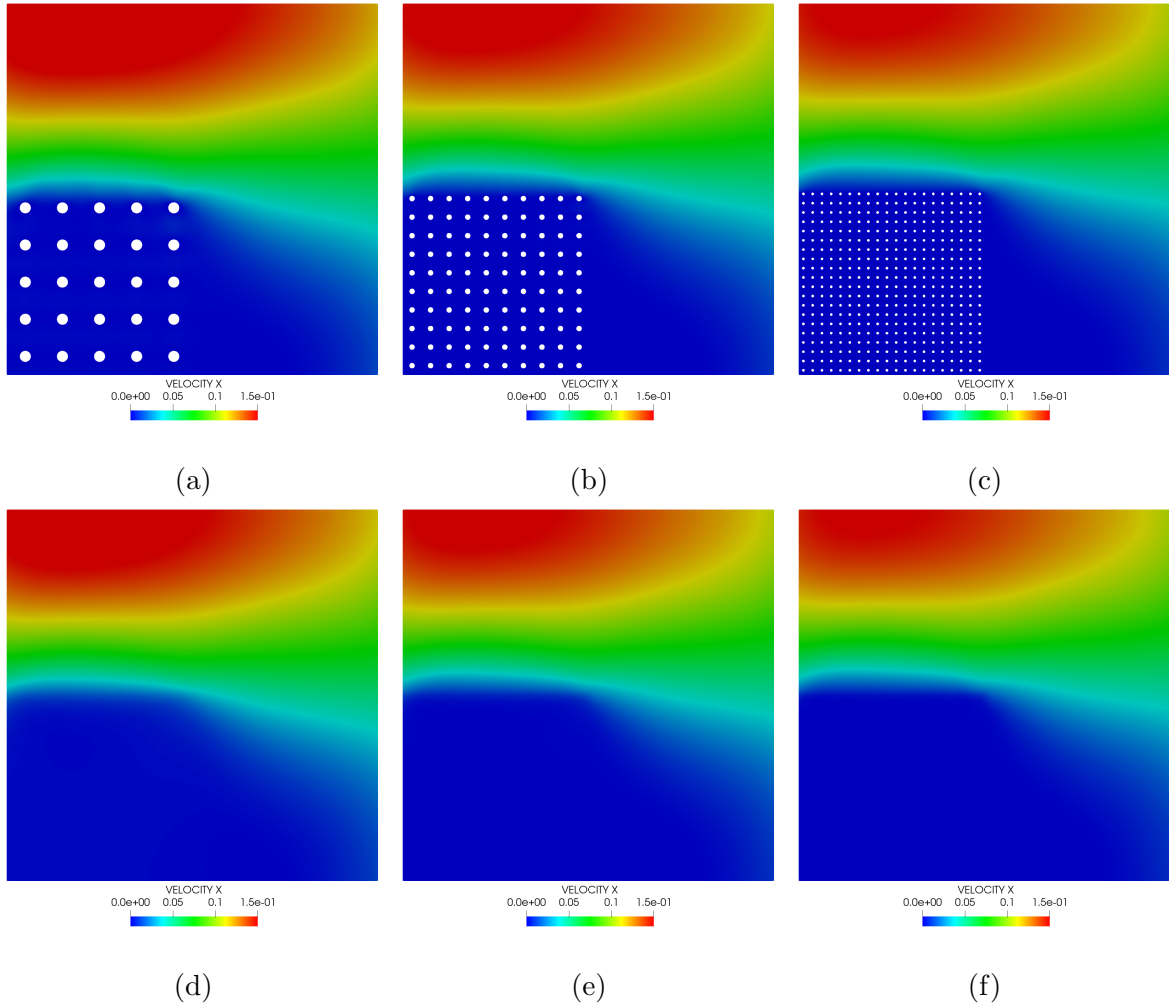


Figure 13: Velocity component  $\mathbf{v}_x$  ( $\text{m s}^{-1}$ ) at  $t = 0.36\text{ s}$  compared with single-scale simulations for the L-shaped domain problem: (a) single-scale simulation with  $l = 0.1\text{ m}$ , (b) single-scale simulation with  $l = 0.05\text{ m}$ , (c) single-scale simulation with  $l = 0.025\text{ m}$ , (d) multiscale simulation with  $l = 0.1\text{ m}$ , (e) multiscale simulation with  $l = 0.05\text{ m}$ , (f) multiscale simulation with  $l = 0.025\text{ m}$ .

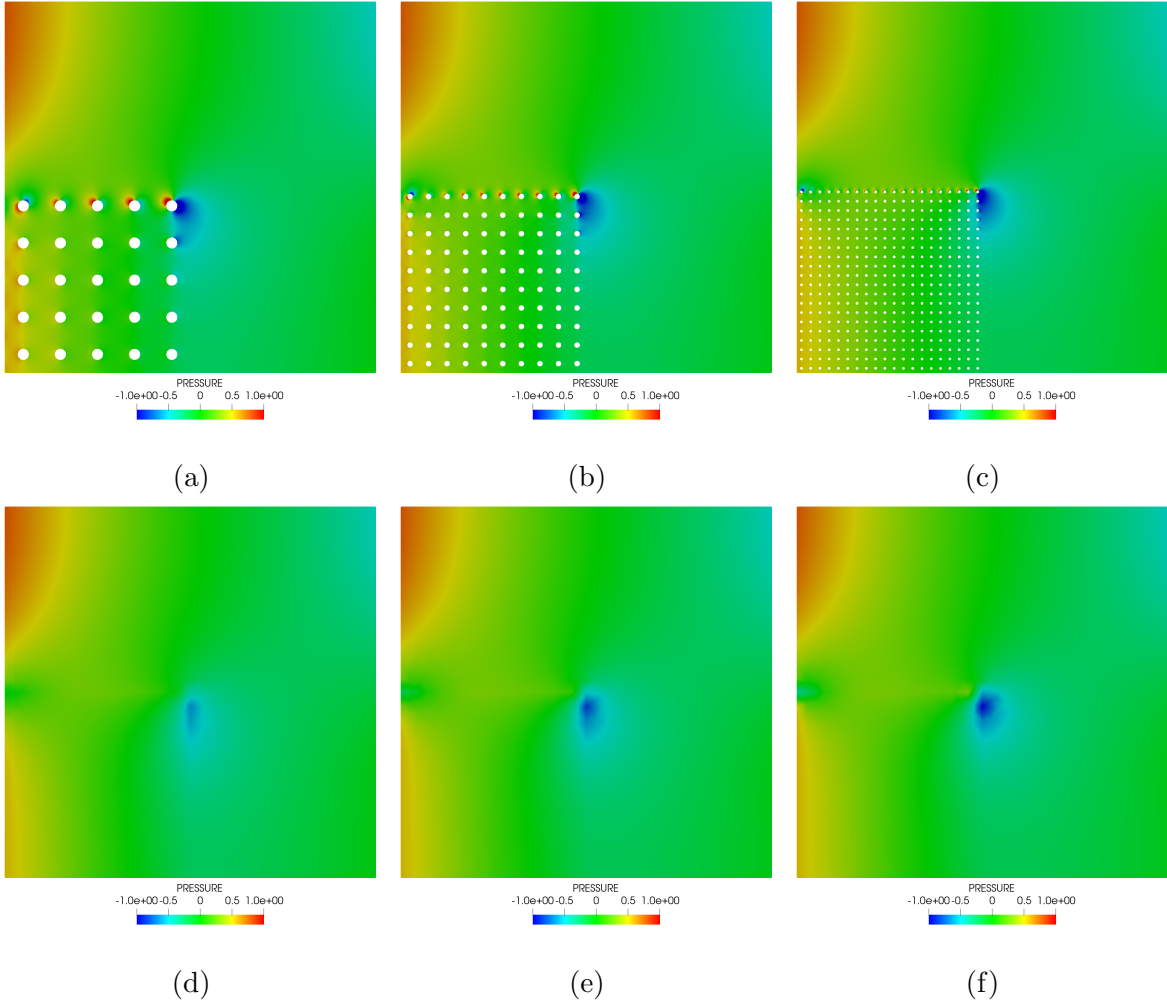


Figure 14: Normalized pressure  $p$  at  $t = 0.36$  s compared with single-scale simulations for the L-shaped domain problem: (a) single-scale simulation with  $l = 0.1$  m, (b) single-scale simulation with  $l = 0.05$  m, (c) single-scale simulation with  $l = 0.025$  m, (d) multiscale simulation with  $l = 0.1$  m, (e) multiscale simulation with  $l = 0.05$  m, (f) multiscale simulation with  $l = 0.025$  m.

This influence of the fine scale domain size on the error of the multiscale model is analyzed quantitatively in Figure 15, where we report the evolution of the velocity along a horizontal line at  $t = 0.36$  s. On the one hand, we can see that the multiscale model's accuracy is always quite good away from obstacles, but that for large obstacles the error is significant in the dark phase. On the other hand, the multiscale model becomes very accurate even in the dark phase for very small obstacles as in Figure 15(c).

We obtain a relative error in 2-norm between the multiscale and single-scale curves of 12% for  $l = 0.1$  m, 4.1% for  $l = 0.05$  m and 2.5% for  $l = 0.025$  m. These results show that the error of the multiscale model converges to zero with decreasing obstacle sizes. This is interesting as the computational cost of the multiscale approach is also decreased for very small obstacles as compared to the single-scale approach.

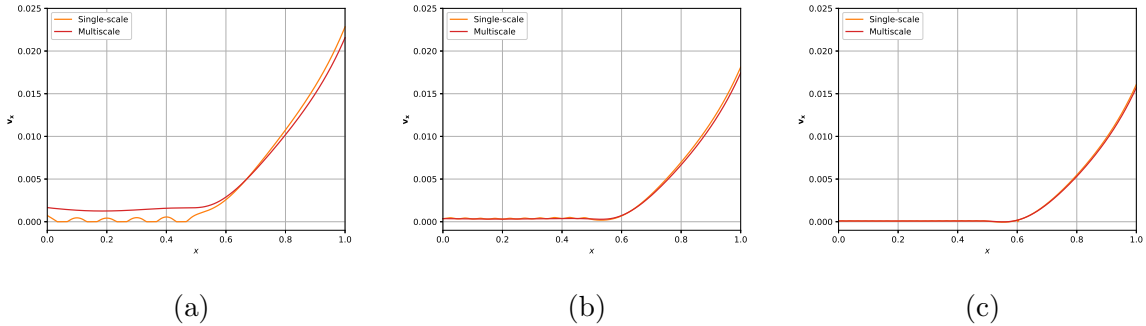


Figure 15: Velocity component  $\mathbf{v}_x$  along the horizontal line  $y = 0.25$  m curves at  $t = 0.36$  s compared with single-scale simulations for the L-shaped domain problem using: (a)  $l = 0.1$  m, (b)  $l = 0.05$  m, (c)  $l = 0.025$  m. Note that in (a) the line goes through obstacles, while it is not the case in (b,c). Lengths are in meters and velocities in meters per second.

#### 4.2.2 Comparison with single-scale models: influence of Reynolds number

Similarly as for the lid-driven cavity problem, we evaluate the accuracy of the multiscale approach for different Reynolds numbers by varying the dynamic viscosity. Because the pressure is imposed instead of the velocity, we can only compute the Reynolds number in post-processing. To compute the Reynolds number, we use the maximum of the velocity component  $\mathbf{v}_x$  throughout the time and space domains, and the coarse scale domain size (1 m). We obtain the values given in Table 1. We denote each configuration according to the table. Note that the fine scale domain size could have been used for the computation of the Reynolds number, in which case the values given in Table 1 would have been ten times smaller.

In Figure 16, we show the evolution of the velocity components at  $x = y = 0.6$  m,

Dynamic viscosity (Pa s)	Reynolds number (multiscale model)	Reynolds number (single-scale model)	Denotation
$10^0$	$1.6 \times 10^{-1}$	$1.7 \times 10^{-1}$	$Re \approx 10^{-1}$
$10^{-1}$	$3.6 \times 10^0$	$3.6 \times 10^0$	$Re \approx 10^0$
$10^{-2}$	$3.8 \times 10^1$	$3.8 \times 10^1$	$Re \approx 10^1$
$10^{-3}$	$3.7 \times 10^2$	$4.4 \times 10^2$	$Re \approx 10^2$

Table 1: Post-processed Reynolds numbers

which is a point located inside the light phase but near the corner of the dark phase. We obtain a relative error in 2-norm between the multiscale and single-scale curves shown in Figure 16(a) of 6.9% for  $Re \approx 10^{-1}$ , 4.4% for  $Re \approx 10^0$ , 5.1% for  $Re \approx 10^1$  and 0.7% for  $Re \approx 10^2$ . The results are very good. Surprisingly, the accuracy even seems to improve for higher Reynolds numbers.

As shown in Figure 16(b), this surprising observation is not confirmed when looking at the second component. We obtain a relative error in 2-norm between the multiscale and single-scale curves shown in Figure 16(b) of 4.0% for  $Re \approx 10^{-1}$ , 3.6% for  $Re \approx 10^0$ , 9.7% for  $Re \approx 10^1$  and 13% for  $Re \approx 10^2$ . Since viscous flows can be modeled with Darcy's law, it is not surprising that our method is more accurate for small Reynolds numbers. These results additionally show that the error increases with the Reynolds number.

Figure 16(b) also reveals that for higher Reynolds numbers the error increases during the simulation. This is clear for the  $Re \approx 10^2$  simulations where we see the multiscale curve slowly drifting away from the single-scale curve. To analyze the flow at the end of the simulation, we show the velocity component  $v_y$  computed by the multiscale and single-scale models for  $Re \approx 10^0$  at  $t = 2.25$  s, 2.40 s, 2.55 s, 2.70 s in Figures 17 and 18. The multiscale model is very accurate in the light phase where there is no obstacle, except for a slight underestimation of the velocity that is clearly visible in Figures 17(d,e). For the comparison in the dark phase, we also show in Figures 17 and 18 the results for the fine scale domain closest to  $x = y = 0.25$  m, which corresponds to the position of the obstacle at the center of the  $5 \times 5$  array. These additional results show that the fine

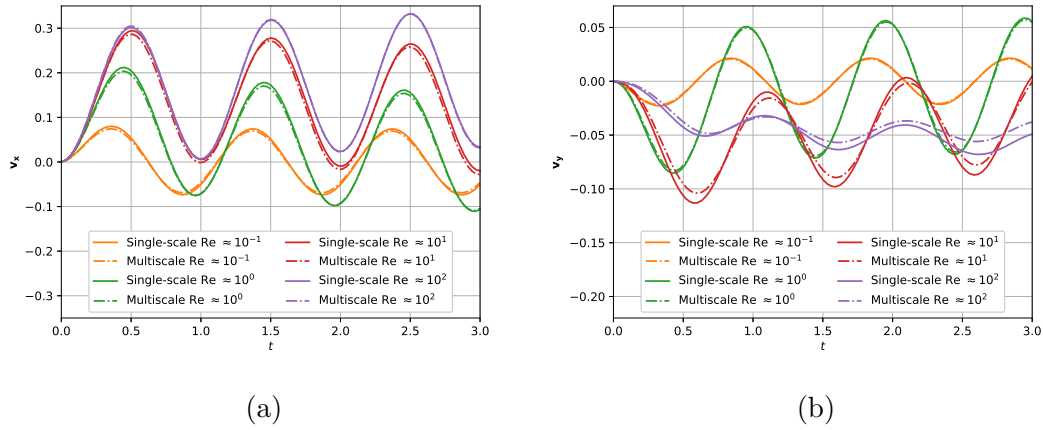


Figure 16: Velocity components evolution curves at  $x = y = 0.6$  m compared with single-scale simulations for the L-shaped domain problem using different Reynolds numbers: (a)  $v_x$ , (b)  $v_y$ . Time is in seconds and velocities in meters per second.

fluctuations of the flow are well captured at the fine scale, while they are smoothed out at the coarse scale in the multiscale approach.

The same results are shown for  $Re \approx 10^1$  in Figures 19 and 20. Although the multiscale simulation results still corresponds well to the single-scale ones, the differences are more visible for this higher Reynolds number.

These results show that the multiscale approach is reliable for a wide range of inertial flows, but not for turbulent flows. The approach in fact lacks not only a turbulence model but also stabilization.

### 4.2.3 Computational cost

The obstacle size was previously kept large in order to limit the computational cost of the single-scale simulations. In the following, we demonstrate the capabilities of the proposed multiscale approach for a case which could not be solved using a single-scale model.

The fine scale domain for the light phase is the same as that presented in Figure 2(c), except that it has a size  $l = 1$  mm and an obstacle radius  $R = 0.15$  mm. We use the same fine scale domain for the dark phase, but with an obstacle radius changed to  $R = 0.39$  mm.

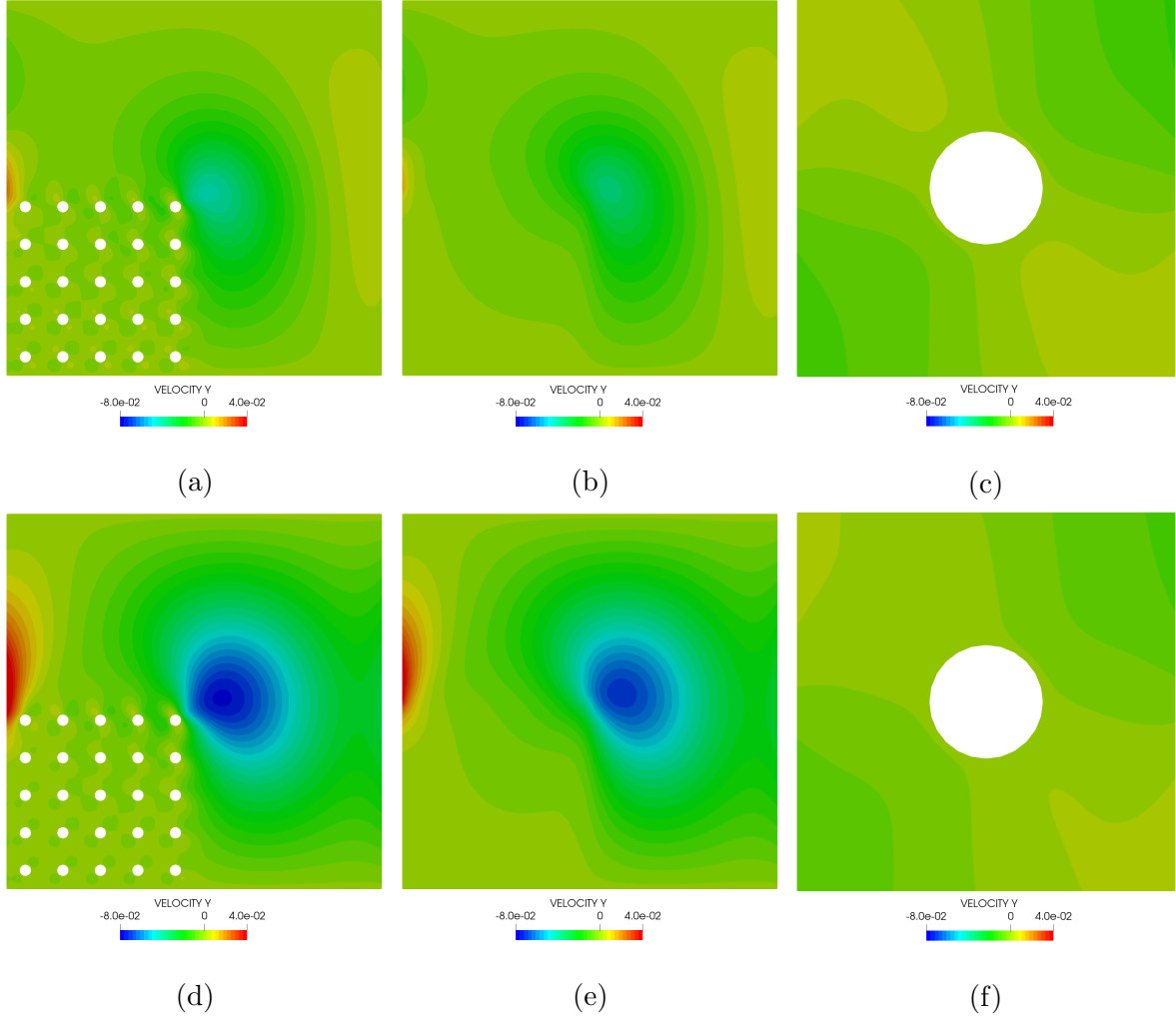


Figure 17: Velocity component  $v_y$  ( $\text{m s}^{-1}$ ) evolution compared with single-scale simulations for the L-shaped domain problem with  $Re \approx 10^0$ : (a) single-scale simulation at  $t = 2.25$  s, (b) multiscale simulation at  $t = 2.25$  s, (c) chosen fine scale domain at  $t = 2.25$  s, (d) single-scale simulation at  $t = 2.40$  s, (e) multiscale simulation at  $t = 2.40$  s, (f) chosen fine scale domain at  $t = 2.40$  s. The chosen fine scale domain is at  $x = y = 0.25$  m and is enlarged for the visualization.

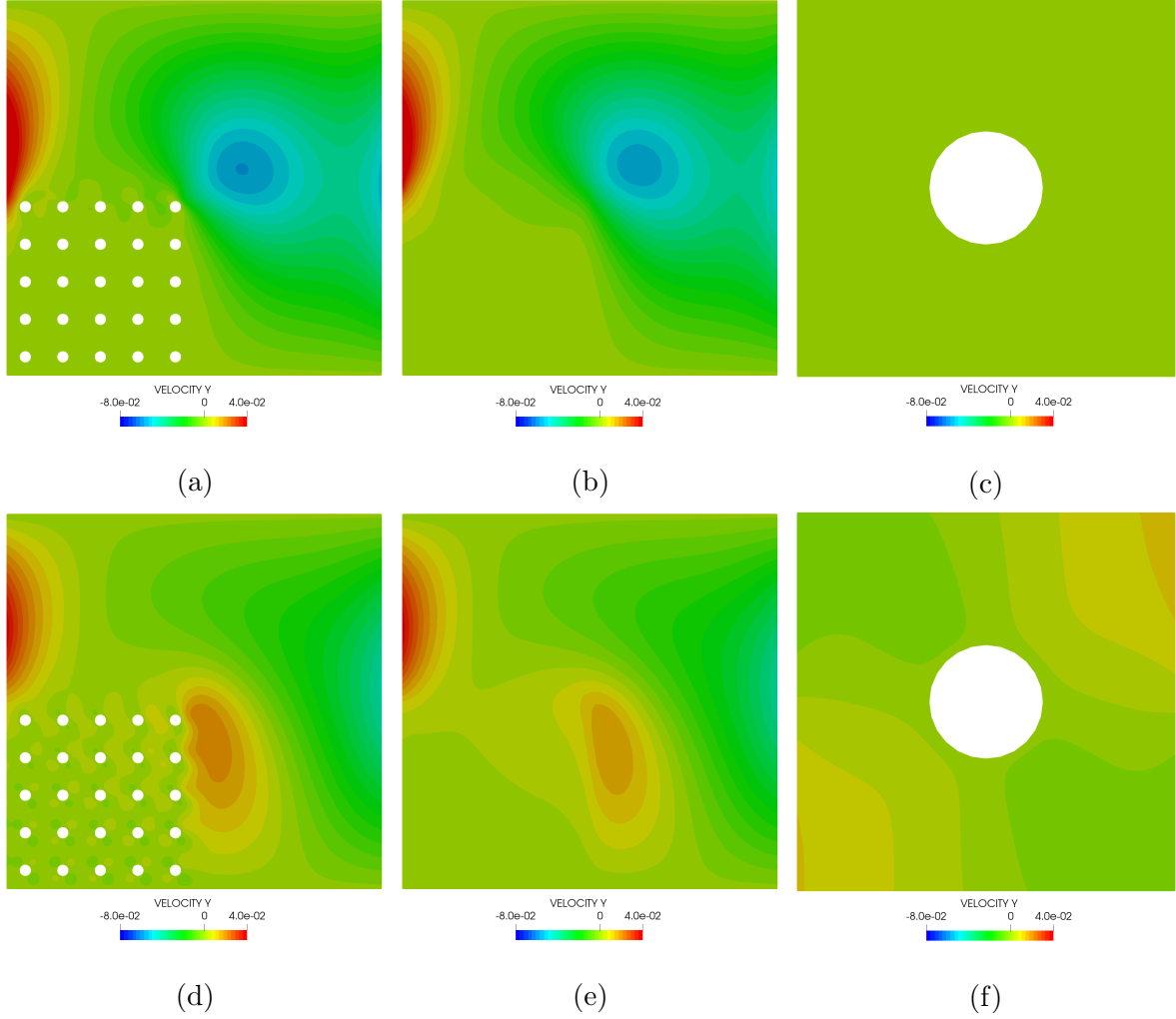


Figure 18: Velocity component  $v_y$  ( $\text{m s}^{-1}$ ) evolution compared with single-scale simulations for the L-shaped domain problem with  $Re \approx 10^0$ : (a) single-scale simulation at  $t = 2.55$  s, (b) multiscale simulation at  $t = 2.55$  s, (c) chosen fine scale domain at  $t = 2.55$  s, (d) single-scale simulation at  $t = 2.70$  s, (e) multiscale simulation at  $t = 2.70$  s, (f) chosen fine scale domain at  $t = 2.70$  s. The chosen fine scale domain is at  $x = y = 0.25$  m and is enlarged for the visualization.

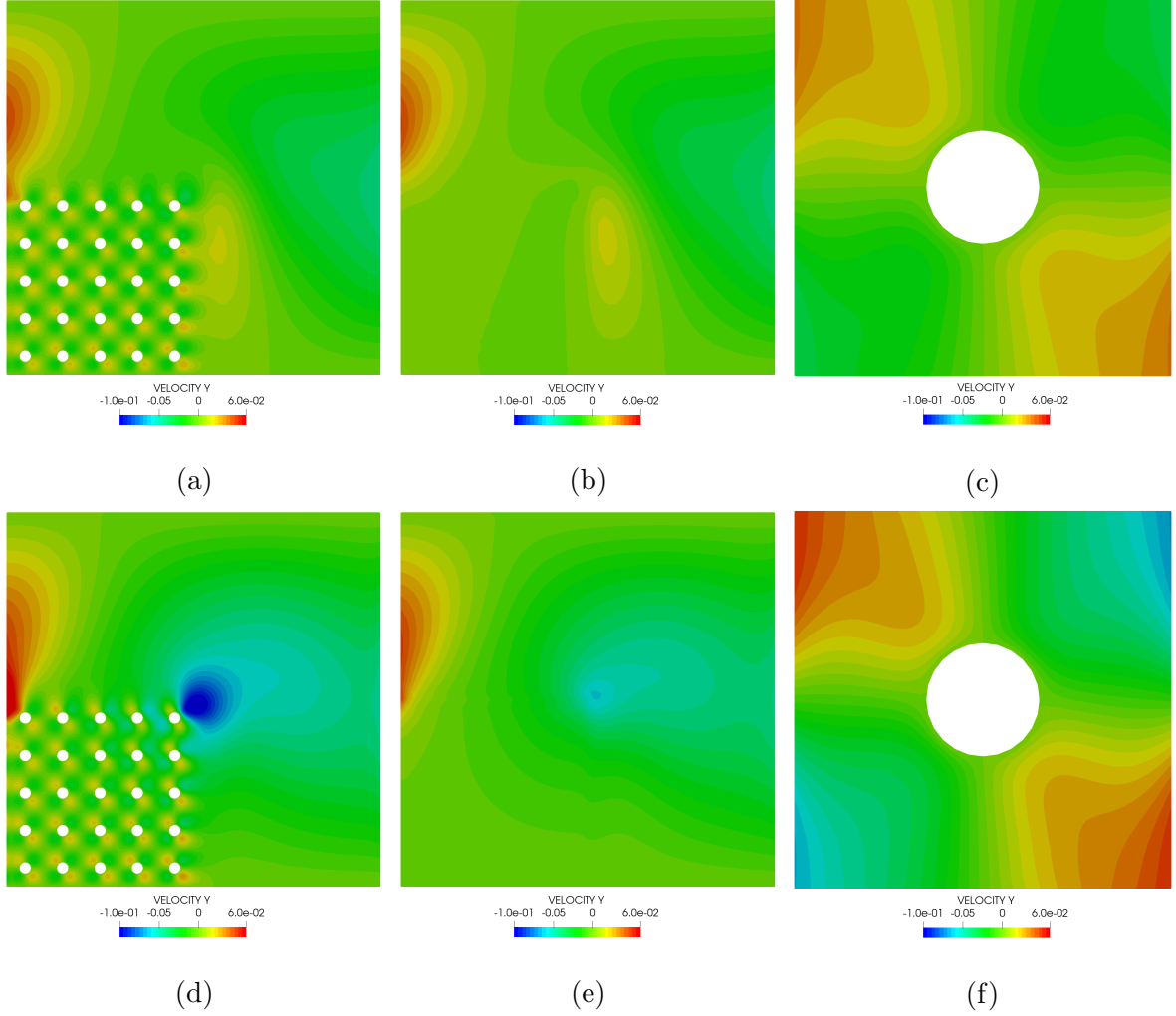


Figure 19: Velocity component  $v_y$  ( $\text{m s}^{-1}$ ) evolution compared with single-scale simulations for the L-shaped domain problem with  $Re \approx 10^1$ : (a) single-scale simulation at  $t = 2.25$  s, (b) multiscale simulation at  $t = 2.25$  s, (c) chosen fine scale domain at  $t = 2.25$  s, (d) single-scale simulation at  $t = 2.40$  s, (e) multiscale simulation at  $t = 2.40$  s, (f) chosen fine scale domain at  $t = 2.40$  s. The chosen fine scale domain is at  $x = y = 0.25$  m and is enlarged for the visualization.

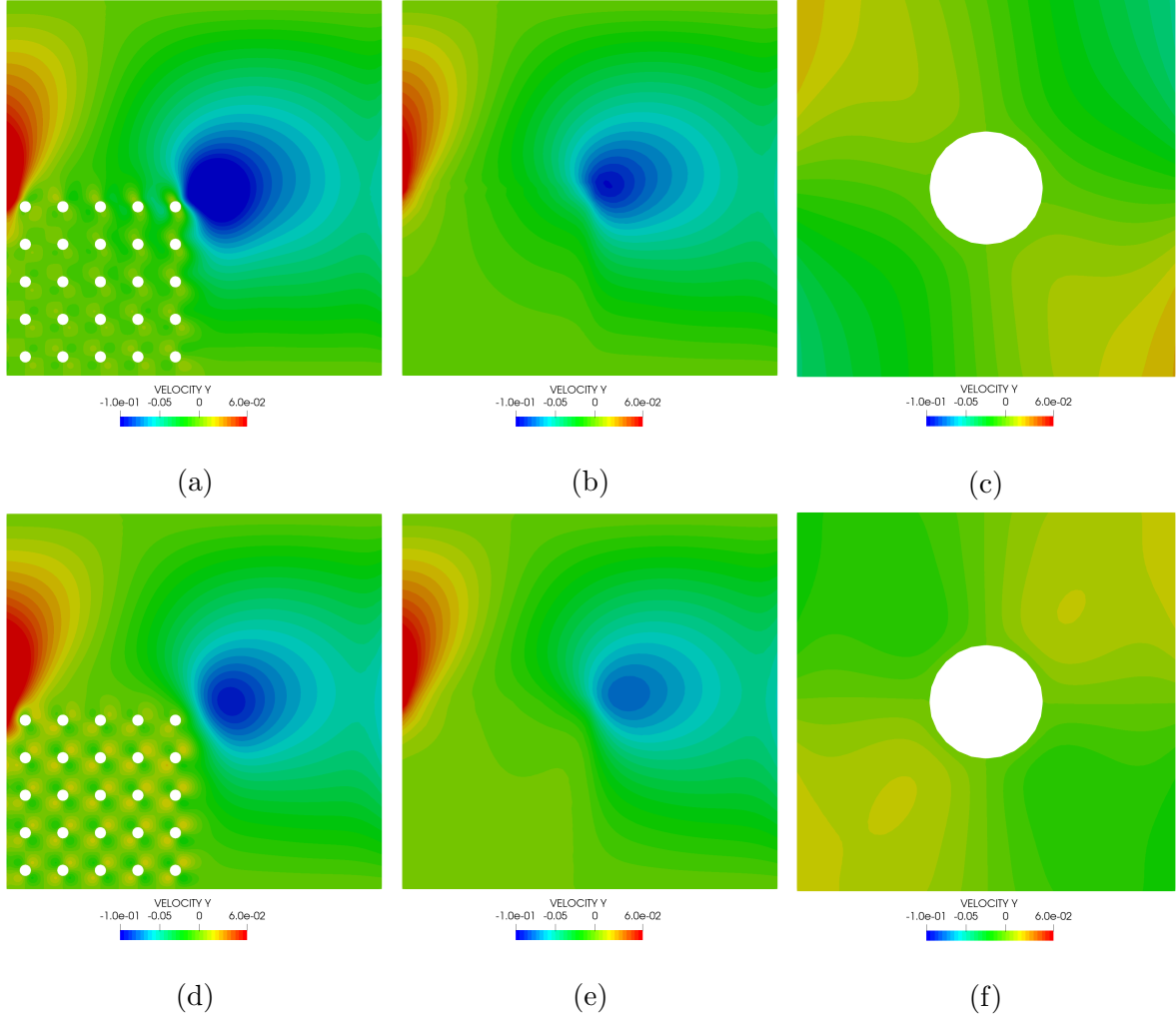


Figure 20: Velocity component  $v_y$  ( $\text{m s}^{-1}$ ) evolution compared with single-scale simulations for the L-shaped domain problem with  $Re \approx 10^1$ : (a) single-scale simulation at  $t = 2.55$  s, (b) multiscale simulation at  $t = 2.55$  s, (c) chosen fine scale domain at  $t = 2.55$  s, (d) single-scale simulation at  $t = 2.70$  s, (e) multiscale simulation at  $t = 2.70$  s, (f) chosen fine scale domain at  $t = 2.70$  s. The chosen fine scale domain is at  $x = y = 0.25$  m and is enlarged for the visualization.

We obtain an obstacle area fraction of 7.1% in the light phase and 48% in the dark phase. Both fine scale domains are discretized with a mesh size of 0.04 mm. The fluid dynamic viscosity is changed to 1  $\mu$ Pa s to obtain a Reynolds number of 50,000. As mentioned in Par. 4.2.2, if the fine scale domain size was used for the computation of the Reynolds number instead of the coarse scale domain size, a Reynolds number of only 50 would have been obtained instead.

For the dark phase in the multiscale model, there are  $2,712 \times 844 = 2,288,928$  fine scale mesh elements and  $2,712 \times 1,852 \times 2 = 10,045,248$  fine scale velocity degrees of freedom. For the light phase in the multiscale model, there are  $7,788 \times 1,524 = 11,868,912$  fine scale mesh elements and  $7,788 \times 3,712 \times 2 = 57,818,112$  fine scale velocity degrees of freedom. This amounts to a total of 14,157,840 fine scale mesh elements and 67,863,360 fine scale velocity degrees of freedom. We estimate that for this problem the number of elements in a single-scale model would be  $\frac{844 \times 1 + 1,524 \times 3}{4} \times 1,000 \times 1,000 \approx 10^9$ . The number of velocity degrees of freedom would be of the same order. There is a reduction by 100 of the computational complexity using the proposed multiscale approach.

The multiscale approach is also theoretically more relevant for parallel computing. We indeed observe in our simulations that 99% of the computation time is spent solving fine scale problems, which are independent problems that can easily be solved in parallel as mentioned in Subsec. 3.4. To demonstrate the efficiency of this parallel algorithm, we conduct strong and weak scaling analyses by solving the first three time steps of the simulation using different numbers of CPUs. We consider for these analyses only the computation time spent in the multiscale scheme presented in Subsec. 3.4, but the conclusions also apply for the total computation time since all other operations (pre-processing, post-processing, writing outputs) have a negligible cost. These simulations are run on specific nodes of a computing cluster which have an *Intel(R) Xeon(R) E5-2630 v3 @ 2.40GHz* multi-processor with 16 CPUs and 64 GB of RAM.

The results are reported in Table 2 for the strong scaling analysis. It is quite impressive that we still benefit from parallel execution when we use 64 CPUs for a coarse mesh of only 1,750 elements, as this leads to less than 30 coarse mesh elements per partition. We could expect better results, however, in relation to our comments on the independence

of the fine scale problems.

Number of CPUs	Computation time per time step (h)	Speedup	Efficiency (%)
1	45.84	1	100
2	25.29	1.81	91
4	15.16	3.02	76
8	8.24	5.56	70
16	4.78	9.58	60
32	2.50	18.34	57
64	1.28	35.81	56

Table 2: Strong scaling results. The efficiency is defined as the speedup divided by the number of CPUs. All simulations use a single computing node, except for the one with 32 CPUs which uses two, and the one with 64 which uses four.

Weak scaling analysis results are reported in Table 3. We observe that distributed computing can be exploited to solve larger problems with a limited increase of computation time. The results are again a little disappointing in relation to our comments on the independence of the fine scale problems. This can be explained by the unbalance in computational cost of the fine scale problems. In this problem for instance, there is a singularity at the corner of the dark phase which leads to larger coarse scale velocity gradients in this region. The fine scale problems, consequently, involve more Newton-Raphson iterations.

The limited parallel efficiency with more CPUs can also be explained by the difference in the number of fine scale mesh elements and velocity degrees of freedom between the two phases. With more CPUs, some partitions end up completely embedded in one phase, which increases the unbalance of the computational cost of the fine scale solves between partitions. To deal with this issue, it could be interesting to regularly and dynamically re-balance the partitions during the simulation based on the computation time

Number of CPUs	Number of coarse mesh elements	Computation time per time step (h)	Efficiency (%)
1	482	13.11	100
2	982	14.92	88
4	2,008	16.95	77
8	4,196	18.31	71
16	7,582	18.80	70
32	15,972	17.74	74
64	32,966	27.03	49

Table 3: Weak scaling results. All simulations use a single computing node, except for the one with 32 CPUs which uses four, and the one with 64 which uses eight.

spent solving fine scale problems.

For the simulation using 64 CPUs, in addition, we solve all time steps and show the results in Figure 21. We observe spurious oscillations at the boundary layer between the two phases. They can be explained by the discontinuity in obstacle area fraction between the two phases, and by the Reynolds number as well.

To reduce the oscillations, the coarse scale mesh can be refined in the boundary layer. An adapted coarse scale mesh with 3,081 elements in the dark phase and 3,149 in the light phase is proposed in Figure 22(a). Simulation results using this mesh are shown in Figure 22(b-f). The oscillations are drastically reduced. This indicates that for problems involving different distributions of obstacles, a special care should be taken for the creation of the coarse scale mesh, even though obstacles are only discretized in the fine scale domains.

Fine scale results are shown for different locations at  $t = 2.40$  s in Figure 23. The comparison with Figure 22(d) shows that the different orientations of the flow are transferred at the fine scale. As shown in Figure 23(c), in the boundary layer on the light phase

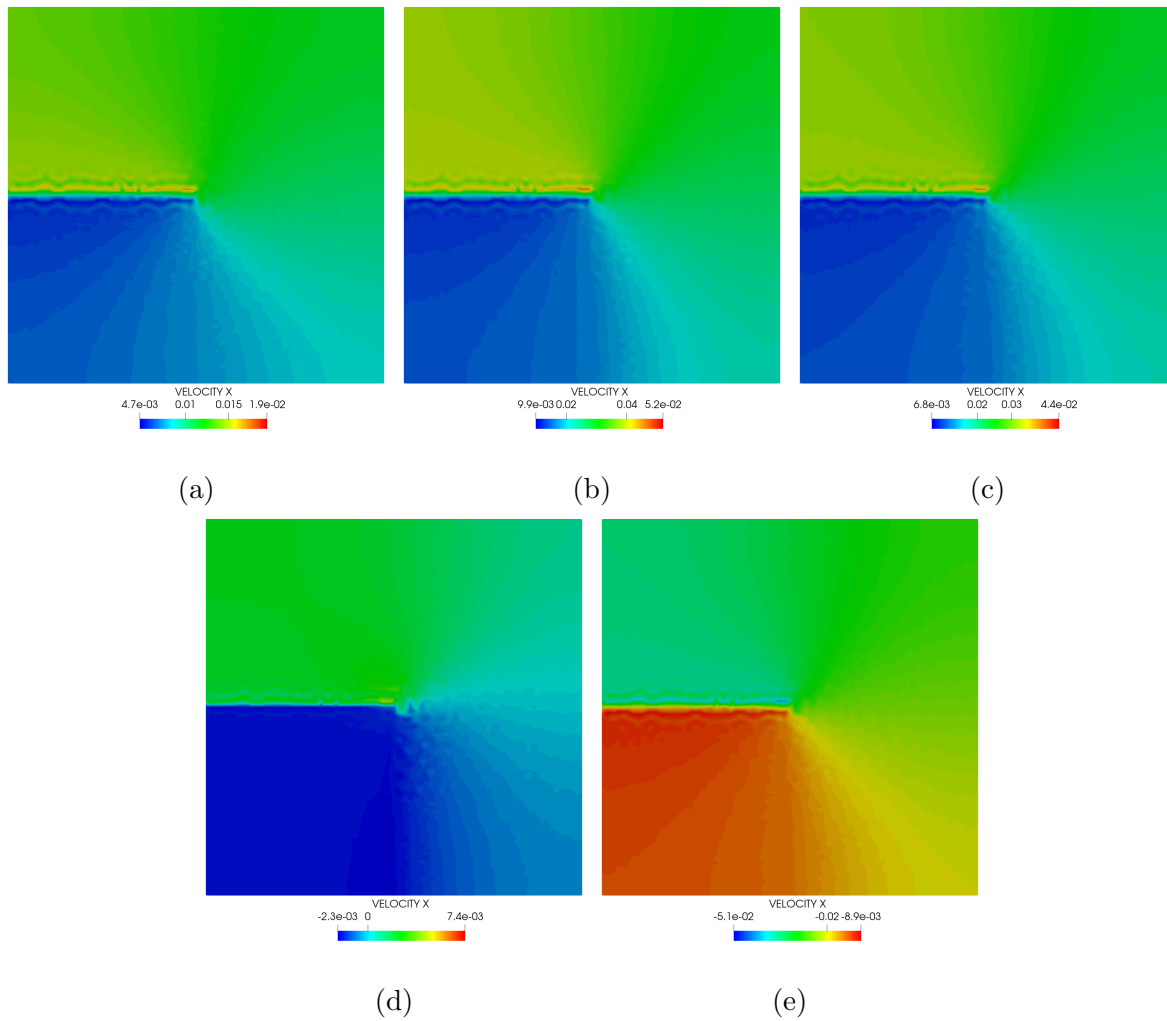


Figure 21: Velocity component  $v_x$  ( $\text{m s}^{-1}$ ) evolution for the L-shaped domain problem with  $l = 1$  mm at: (a)  $t = 2.10$  s, (b)  $t = 2.25$  s, (c)  $t = 2.40$  s, (d)  $t = 2.55$  s, (e)  $t = 2.70$  s.

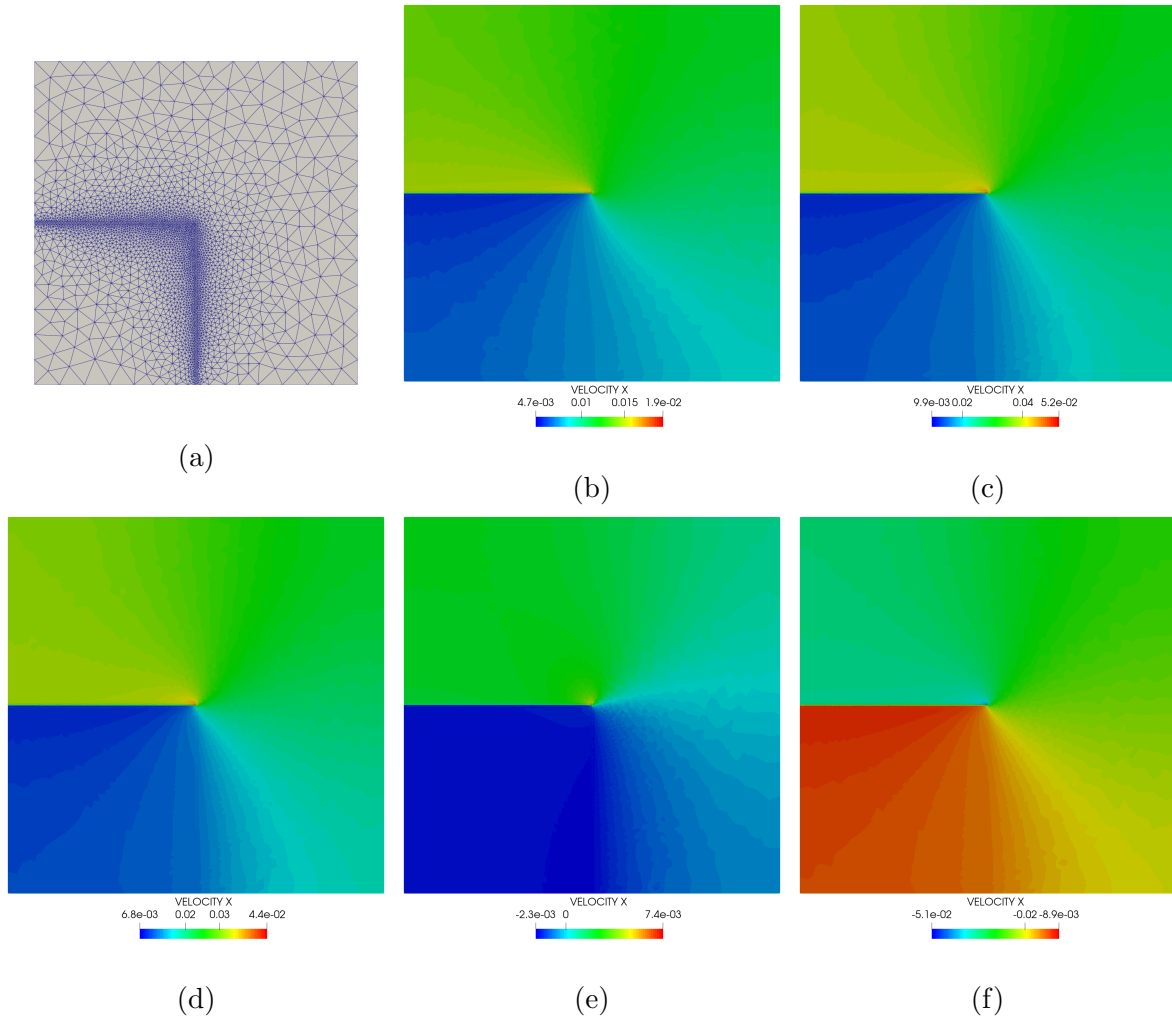


Figure 22: Adapted coarse scale mesh (a) and velocity component  $\mathbf{v}_x$  ( $\text{m s}^{-1}$ ) evolution for the L-shaped domain problem with  $l = 1$  mm at: (b)  $t = 2.10$  s, (c)  $t = 2.25$  s, (d)  $t = 2.40$  s, (e)  $t = 2.55$  s, (f)  $t = 2.70$  s.

side, in particular, we observe larger velocities at both scales. This demonstrates the capabilities of the multiscale method to capture the features of the flow simultaneously at the two scales.

## 5 Conclusions

In this paper, a new multiscale theory and, more importantly, its implementation in a Finite Element (FE) context have been proposed for unsteady incompressible flows in domains containing very small obstacles. In this theory, the obstacles are represented in fine scale domains that are attached at each integration point of the coarse scale domain. This avoids representing the obstacles directly at the coarse scale, which becomes computationally intractable for very small obstacles.

The proposed method therefore involves a coarse scale mesh, a coarse scale velocity field, and fine scale meshes and velocity fields. A principle of multiscale virtual power has been formulated to relate the coarse scale velocity to its fine scale counterpart, and to provide simultaneously the boundary conditions for the fine scale problems and the equations to solve for the coarse scale problem. The boundary conditions at the fine scale are indeed applied through Lagrange multipliers that directly provide the coarse scale force per unit volume and stress for the equations to solve at the coarse scale. The incompressibility constraint has been handled with special care by introducing an independent pressure variable at both scales.

A two-way coupling algorithm has been proposed to implement the theory. This FExFE (FE<sup>2</sup>) scheme relies on the Taylor-Hood P2/P1 pair at both scales to deal with the incompressibility constraint. A fully-implicit Newton-Raphson procedure has been proposed to solve the coarse and fine scale problems simultaneously. Automatic differentiation has been used to linearize the coarse scale force per unit volume and stress.

The multiscale simulation method has been applied to two test problems involving inertial flows. Comparisons with single-scale simulations where the obstacles are directly discretized in the coarse scale domain mesh have shown that the proposed multiscale approach is reliable. The proposed approach, in particular, captures well the influence of the obstacle size and area fraction. Its accuracy increases when the ratio between the

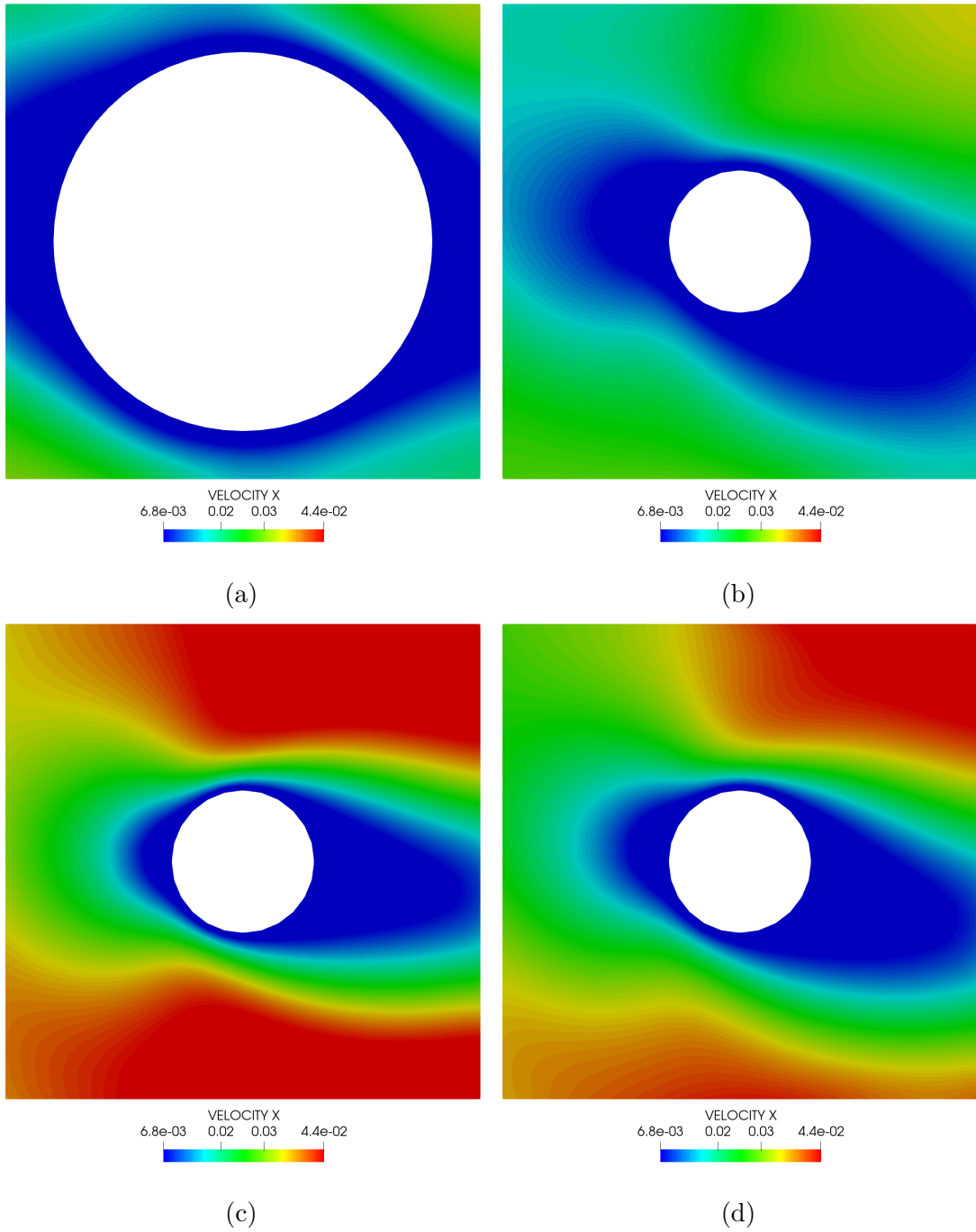


Figure 23: Velocity component  $v_x$  ( $\text{m s}^{-1}$ ) for the L-shaped domain problem with  $l = 1$  mm and an adapted mesh at  $t = 2.40$  s within a fine scale domain located at: (a)  $x = 0.45$  m,  $y = 0.45$  m, (b)  $x = 0.55$  m,  $y = 0.45$  m, (c)  $x = 0.45$  m,  $y = 0.55$  m, (d)  $x = 0.55$  m,  $y = 0.55$  m.

coarse scale domain size and the obstacle size increases, and decreases when the Reynolds number increases.

An analysis of the computational cost of the method has also been conducted. The results show that the method is not relevant when the ratio between the coarse scale domain size and the obstacle size is small, whereas it becomes interesting from the point of view of both accuracy and computational cost when this ratio is larger than 100. This method is hence relevant for a wide range of applications involving flows in porous media such as resin transfer molding or reservoir engineering.

The parallel implementation of the method in a distributed computing environment has also been presented and the efficiency of this implementation has been analyzed. The results show that the parallel efficiency is impressive with respect to the number of coarse scale mesh elements, but that it could be improved. A re-balancing strategy based on the computation time spent solving fine scale problems would in fact be worth considering. Consequently, the proposed multiscale framework for unsteady flows opens a new avenue for multiscale modeling in computational fluid dynamics. Other types of flows such as non-Newtonian flows or multiphase flows could for instance be integrated in the multiscale framework.

## References

- [1] A. Y. Beliaev and S. M. Kozlov, “Darcy equation for random porous media,” Communications on Pure and Applied Mathematics, vol. 49, pp. 1–34, jan 1996.
- [2] P. J. Blanco, P. J. Sánchez, E. A. de Souza Neto, and R. A. Feijóo, “Variational Foundations and Generalized Unified Theory of RVE-Based Multiscale Models,” Archives of Computational Methods in Engineering, vol. 23, pp. 191–253, jun 2016.
- [3] E. Skjetne and J. L. Auriault, “New insights on steady, non-linear flow in porous media,” European Journal of Mechanics - B/Fluids, vol. 18, pp. 131–145, jan 1999.
- [4] E. Skjetne and J. L. Auriault, “High-velocity laminar and turbulent flow in porous media,” Transport in Porous Media, vol. 36, no. 2, pp. 131–147, 1999.

- [5] P. J. Blanco, A. Clausse, and R. A. Feijóo, “Homogenization of the Navier-Stokes equations by means of the Multi-scale Virtual Power Principle,” Computer Methods in Applied Mechanics and Engineering, vol. 315, pp. 760–779, 2017.
- [6] M. Geers, V. Kouznetsova, and W. Brekelmans, “Multi-scale computational homogenization: Trends and challenges,” Journal of Computational and Applied Mathematics, vol. 234, pp. 2175–2182, aug 2010.
- [7] K. Matouš, M. G. Geers, V. G. Kouznetsova, and A. Gillman, “A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials,” Journal of Computational Physics, vol. 330, pp. 192–220, feb 2017.
- [8] F. Feyel, “Multiscale FE2 elastoviscoplastic analysis of composite structures,” Computational Materials Science, vol. 16, pp. 344–354, dec 1999.
- [9] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, and G. Scovazzi, “Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows,” Computer Methods in Applied Mechanics and Engineering, vol. 197, pp. 173–201, dec 2007.
- [10] S. Idelsohn, N. Nigro, A. Larreteguy, J. M. Gimenez, and P. Ryzhakov, “A pseudo-DNS method for the simulation of incompressible fluid flows with instabilities at different scales,” Computational Particle Mechanics, vol. 7, pp. 19–40, jan 2020.
- [11] M. Shakoor, “FEMS – A Mechanics-oriented Finite Element Modeling Software,” Computer Physics Communications, vol. 260, p. 107729, mar 2021.
- [12] G. Karypis and V. Kumar, “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,” SIAM Journal on Scientific Computing, vol. 20, pp. 359–392, jan 1998.
- [13] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, “PETSc Web page,” 2021. <https://petsc.org/>.

- [14] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, "A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling," SIAM Journal on Matrix Analysis and Applications, vol. 23, pp. 15–41, jan 2001.
- [15] T. A. Davis, "Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method," ACM Transactions on Mathematical Software, vol. 30, pp. 196–199, jun 2004.
- [16] M. Shakoor and C. H. Park, "A higher-order finite element method with unstructured anisotropic mesh adaption for two phase flows with surface tension," Computers & Fluids, vol. 230, p. 105154, nov 2021.