Fixed mesh methods in computational mechanics

Ramon Codina and Joan Baiges

Universitat Politècnica de Catalunya

València, September 2010

Outline

Outline

Objectives

The objectives of fixed mesh methods in CFD are:

- To solve flow problems in moving domains.
- To use a single finite element mesh that covers all the region occupied by the fluid along its evolution.
- To be able to prescribe in an easy manner boundary conditions on the moving surfaces.

The main issues to be addressed are:

- To impose boundary conditions on non-matching grids (meshes).
- To cope with the motion of the computational domain, when it exists.
- To couple fluid and solid motions in FSI applications.

Objectives

The objectives of fixed mesh methods in CFD are:

- To solve flow problems in moving domains.
- To use a single finite element mesh that covers all the region occupied by the fluid along its evolution.
- To be able to prescribe in an easy manner boundary conditions on the moving surfaces.

The main issues to be addressed are:

- To impose boundary conditions on non-matching grids (meshes).
- To cope with the motion of the computational domain, when it exists.
- To couple fluid and solid motions in FSI applications.

Objectives

The objectives of fixed mesh methods in CFD are:

- To solve flow problems in moving domains.
- To use a single finite element mesh that covers all the region occupied by the fluid along its evolution.
- To be able to prescribe in an easy manner boundary conditions on the moving surfaces.

The main issues to be addressed are:

- To impose boundary conditions on non-matching grids (meshes).
- To cope with the motion of the computational domain, when it exists.
- To couple fluid and solid motions in FSI applications.

Objectives

The objectives of fixed mesh methods in CFD are:

- To solve flow problems in moving domains.
- To use a single finite element mesh that covers all the region occupied by the fluid along its evolution.
- To be able to prescribe in an easy manner boundary conditions on the moving surfaces.

The main issues to be addressed are:

- To impose boundary conditions on non-matching grids (meshes).
- To cope with the motion of the computational domain, when it exists.
- To couple fluid and solid motions in FSI applications.

Objectives

The objectives of fixed mesh methods in CFD are:

- To solve flow problems in moving domains.
- To use a single finite element mesh that covers all the region occupied by the fluid along its evolution.
- To be able to prescribe in an easy manner boundary conditions on the moving surfaces.

The main issues to be addressed are:

- To impose boundary conditions on non-matching grids (meshes).
- To cope with the motion of the computational domain, when it exists.
- To couple fluid and solid motions in FSI applications.

Objectives

The objectives of fixed mesh methods in CFD are:

- To solve flow problems in moving domains.
- To use a single finite element mesh that covers all the region occupied by the fluid along its evolution.
- To be able to prescribe in an easy manner boundary conditions on the moving surfaces.

The main issues to be addressed are:

- To impose boundary conditions on non-matching grids (meshes).
- To cope with the motion of the computational domain, when it exists.
- To couple fluid and solid motions in FSI applications.

Objectives

The objectives of fixed mesh methods in CFD are:

- To solve flow problems in moving domains.
- To use a single finite element mesh that covers all the region occupied by the fluid along its evolution.
- To be able to prescribe in an easy manner boundary conditions on the moving surfaces.

The main issues to be addressed are:

- To impose boundary conditions on non-matching grids (meshes).
- To cope with the motion of the computational domain, when it exists.
- To couple fluid and solid motions in FSI applications.

Objectives

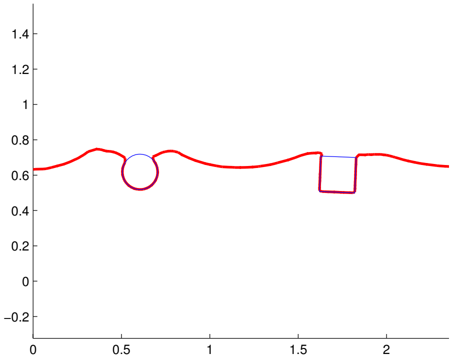
The objectives of fixed mesh methods in CFD are:

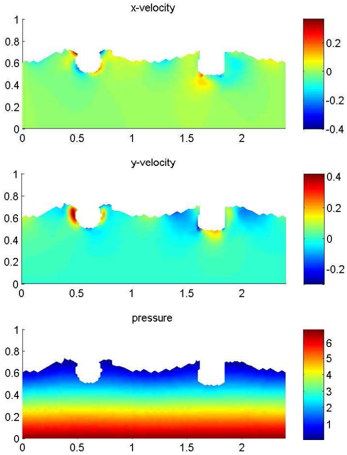
- To solve flow problems in moving domains.
- To use a single finite element mesh that covers all the region occupied by the fluid along its evolution.
- To be able to prescribe in an easy manner boundary conditions on the moving surfaces.

The main issues to be addressed are:

- To impose boundary conditions on non-matching grids (meshes).
- To cope with the motion of the computational domain, when it exists.
- To couple fluid and solid motions in FSI applications.

Falling of two bodies I





Falling of two bodies II

Falling of a ball

Contents

The contents of the talk are:

- Problem statement.
- Approximate imposition of boundary conditions. This often classifies a particular method.
- Methods to account for the motion of the computational domain.
- Summary of methods described.

Emphasis will be put on methods developed in our group.

Contents

The contents of the talk are:

- Problem statement.
- Approximate imposition of boundary conditions. This often classifies a particular method.
- Methods to account for the motion of the computational domain.
- Summary of methods described.

Emphasis will be put on methods developed in our group.

Contents

The contents of the talk are:

- Problem statement.
- Approximate imposition of boundary conditions. This often classifies a particular method.
- Methods to account for the motion of the computational domain.
- Summary of methods described.

Emphasis will be put on methods developed in our group.

Contents

The contents of the talk are:

- Problem statement.
- Approximate imposition of boundary conditions. This often classifies a particular method.
- Methods to account for the motion of the computational domain.
- Summary of methods described.

Emphasis will be put on methods developed in our group.

Contents

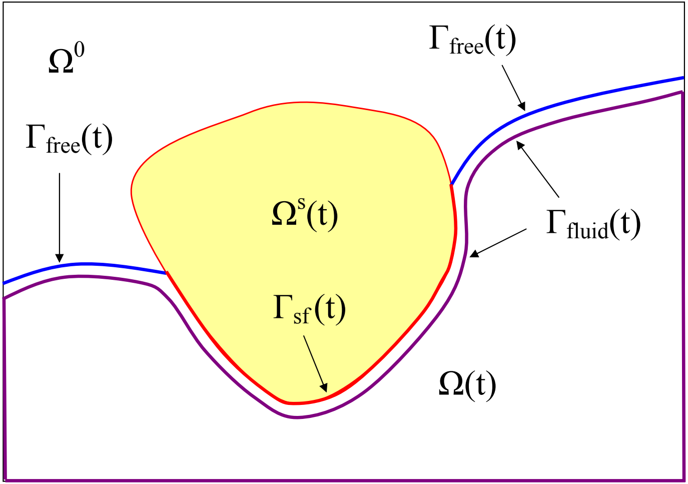
The contents of the talk are:

- Problem statement.
- Approximate imposition of boundary conditions. This often classifies a particular method.
- Methods to account for the motion of the computational domain.
- Summary of methods described.

Emphasis will be put on methods developed in our group.

Outline

Setting


Ω^0
$\Gamma_{free}(t)$
$\Gamma_{free}(t)$
$\Omega^s(t)$
$\Gamma_{fluid}(t)$
$\Gamma_{sf}(t)$
$\Omega(t)$

Flow equations

Find a velocity $\boldsymbol{u}$ and a pressure p such that

$$\rho[\partial_t \boldsymbol{u} + (\boldsymbol{u} - \boldsymbol{u}_{\mathrm{dom}}) \cdot \nabla \boldsymbol{u}] - \nabla \cdot [2\mu\nabla^S(\boldsymbol{u})] + \nabla p = \boldsymbol{f}$$
$$\nabla \cdot \boldsymbol{u} = 0$$

The boundary conditions can be:

- Free surface, $\Gamma_{\mathrm{free}}(t)$: p (or the stress) given, $\boldsymbol{u}$ unknown.
- Solid contact, $\Gamma_{\mathrm{sf}}(t)$: $\boldsymbol{u}$ given, p (or the stress) unknown.

The time-discrete problem

Notation:

$$\delta f^{n+1} = f^{n+1} - f^{n}, \delta_t f^{n+1} = \frac{f^{n+1} - f^{n}}{\delta t},$$
$$f^{n+\theta} = \theta f^{n+1} + (1-\theta) f^{n}, \quad \theta \in [1/2, 1].$$

Time discrete Navier-Stokes equations:

$$\rho \left[\delta_t \boldsymbol{u}^{n+1}\Big|_{\boldsymbol{x}^n} + (\boldsymbol{u}^{n+\theta} - \boldsymbol{u}^{n+\theta}_{\mathrm{dom}}) \cdot \nabla \boldsymbol{u}^{n+\theta} \right]$$
$$- \nabla \cdot (2\mu \nabla^S \boldsymbol{u}^{n+\theta}) + \nabla p^{n+1} = \rho \boldsymbol{f}^{n+1},$$
$$\nabla \cdot \boldsymbol{u}^{n+\theta} = 0,$$

where $\delta_t \boldsymbol{u}^{n+1}\big|_{\boldsymbol{x}^n} = (\boldsymbol{u}^{n+1}(\boldsymbol{x}) - \boldsymbol{u}^{n}(\boldsymbol{x}^n))/\delta t$, being $\boldsymbol{x}$ the spatial coordinates in $\Omega(t^{n+\theta})$, and

$$\boldsymbol{u}^{n+\theta}_{\mathrm{dom}} = \frac{1}{\theta \delta t} \left(\chi_{t^{n+\theta}, t^n}(\boldsymbol{x}^n) - \boldsymbol{x}^n \right).$$

The fully discrete problem

Find $\boldsymbol{u}_h^{n+1}$ and p_h^{n+1} such that

$$\int_\Omega \boldsymbol{v}_h \cdot \rho \; \delta_t \boldsymbol{u}_h^{n+1}\Big|_{\boldsymbol{x}^n} \,\mathrm{d}\boldsymbol{x} + \int_\Omega 2\nabla^S \boldsymbol{v}_h : \mu \nabla^S \boldsymbol{u}_h^{n+\theta} \,\mathrm{d}\boldsymbol{x}$$

$$+ \int_\Omega \boldsymbol{v}_h \cdot (\rho\,(\boldsymbol{u}_h^{n+\theta} - \boldsymbol{u}_{\mathrm{dom}}^{n+\theta}) \cdot \nabla \boldsymbol{u}_h^{n+\theta})\,\mathrm{d}\boldsymbol{x} - \int_\Omega p_h^{n+1} \nabla \cdot \boldsymbol{v}_h \,\mathrm{d}\boldsymbol{x}$$

$$= \int_\Omega \boldsymbol{v}_h \cdot \boldsymbol{f}^{n+\theta} \,\mathrm{d}\boldsymbol{x},$$

$$\int_\Omega q_h \nabla \cdot \boldsymbol{u}_h^{n+\theta} \,\mathrm{d}\boldsymbol{x} = 0.$$

Stabilization techniques might be required.

Solid body equations

Let us now consider the solid body domain $\Omega_s(t) \subset \Omega^0$. The equations of motion are:

$$\rho_s \frac{d^2 \boldsymbol{d}}{dt^2} = \nabla \cdot \boldsymbol{\sigma}_s + \rho_s \boldsymbol{b},$$
$$\rho_s J = \rho_{s0},$$

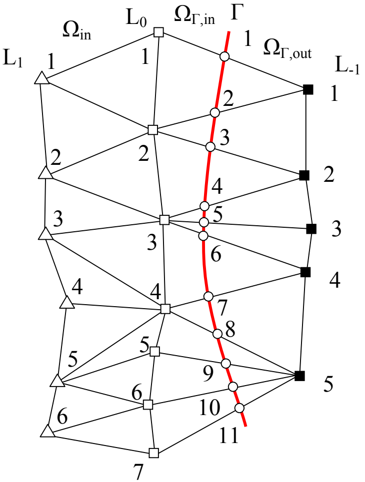
where ρ_s is the solid density, $\boldsymbol{d}$ is the displacement vector, $\boldsymbol{\sigma}_s$ is the Cauchy stress tensor, $\boldsymbol{b}$ is the vector of body forces and

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}, \quad J = \det(\boldsymbol{F}).$$

For linear elastic isotropic materials $\boldsymbol{\sigma}_s = \lambda_s (\nabla \cdot \boldsymbol{d}) \boldsymbol{I} + 2\mu_s \nabla^s \boldsymbol{d}$.

Outline

Problem setting



Immersed boundary method (IBM)

Suppose that the interface is represented by a set of points $\boldsymbol{x}_{\Gamma,i}$, $i = 1, 2, ..., P$. The BC $\boldsymbol{u} = \bar{\boldsymbol{u}}$ is prescribed by adding a force:

$$\boldsymbol{f} = \sum_{i=1}^{P} k(\boldsymbol{u} - \bar{\boldsymbol{u}})\delta(|\boldsymbol{x} - \boldsymbol{x}_{\Gamma,i}|).$$

When smoothed and introduced in the weak form of the problem yields the right-hand-side term

$$\int_\Omega \boldsymbol{v}_h \cdot \boldsymbol{f} \, \mathrm{d}\boldsymbol{x} = \int_\Omega \boldsymbol{v}_h \cdot \sum_{i=1}^{P} k(\boldsymbol{u} - \bar{\boldsymbol{u}})\bar{\delta}(|\boldsymbol{x} - \boldsymbol{x}_{\Gamma,i}|) \, \mathrm{d}\boldsymbol{x}.$$

The *elastic/penalty* constant k has to be large enough (k comes from a membrane coupling in the original Peskin's method). $\bar{\delta}$ is the smoothed Dirac-delta function.

Immersed boundary method (IBM)

Suppose that the interface is represented by a set of points $\boldsymbol{x}_{\Gamma,i}$, $i = 1, 2, ..., P$. The BC $\boldsymbol{u} = \bar{\boldsymbol{u}}$ is prescribed by adding a force:

$$\boldsymbol{f} = \sum_{i=1}^{P} k(\boldsymbol{u} - \bar{\boldsymbol{u}})\delta(|\boldsymbol{x} - \boldsymbol{x}_{\Gamma,i}|).$$

When smoothed and introduced in the weak form of the problem yields the right-hand-side term

$$\int_\Omega \boldsymbol{v}_h \cdot \boldsymbol{f} \, \mathrm{d}\boldsymbol{x} = \int_\Omega \boldsymbol{v}_h \cdot \sum_{i=1}^{P} k(\boldsymbol{u} - \bar{\boldsymbol{u}})\bar{\delta}(|\boldsymbol{x} - \boldsymbol{x}_{\Gamma,i}|) \, \mathrm{d}\boldsymbol{x}.$$

The *elastic/penalty* constant k has to be large enough (k comes from a membrane coupling in the original Peskin's method). $\bar{\delta}$ is the smoothed Dirac-delta function.

Immersed boundary method (IBM)

Suppose that the interface is represented by a set of points $\boldsymbol{x}_{\Gamma,i}$, $i = 1, 2, ..., P$. The BC $\boldsymbol{u} = \bar{\boldsymbol{u}}$ is prescribed by adding a force:

$$\boldsymbol{f} = \sum_{i=1}^{P} k(\boldsymbol{u} - \bar{\boldsymbol{u}})\delta(|\boldsymbol{x} - \boldsymbol{x}_{\Gamma,i}|).$$

When smoothed and introduced in the weak form of the problem yields the right-hand-side term

$$\int_{\Omega} \boldsymbol{v}_h \cdot \boldsymbol{f} \, \mathrm{d}\boldsymbol{x} = \int_{\Omega} \boldsymbol{v}_h \cdot \sum_{i=1}^{P} k(\boldsymbol{u} - \bar{\boldsymbol{u}})\bar{\delta}(|\boldsymbol{x} - \boldsymbol{x}_{\Gamma,i}|) \, \mathrm{d}\boldsymbol{x}.$$

The *elastic/penalty* constant k has to be large enough (k comes from a membrane coupling in the original Peskin's method). $\bar{\delta}$ is the smoothed Dirac-delta function.

Penalty methods

The difference with IBM is the definition of the force. Now it is taken as:

$$\boldsymbol{f} = \frac{\alpha}{h}(\boldsymbol{u} - \bar{\boldsymbol{u}}),$$

which introduced in the discrete weak form of the problem yields

$$\int_\Gamma \boldsymbol{v}_h \cdot \boldsymbol{f} \,\mathrm{d}\boldsymbol{x} = \int_\Gamma \boldsymbol{v}_h \cdot \frac{\alpha}{h}(\boldsymbol{u}_h - \bar{\boldsymbol{u}}) \,\mathrm{d}\boldsymbol{x},$$

where α is the penalty parameter and h is the element size.

Nitsche's method

If we define the operator

$$\boldsymbol{\sigma}(\boldsymbol{v}, q) := -2\boldsymbol{n} \cdot \mu \nabla^S \boldsymbol{v} + q\boldsymbol{n},$$

the term to be added to the right-hand-side of the discrete equations is

$$-\int_\Gamma \boldsymbol{v}_h \cdot \boldsymbol{\sigma}(\boldsymbol{u}_h, p_h) + \int_\Gamma \boldsymbol{v}_h \cdot \frac{\alpha}{h}(\boldsymbol{u}_h - \bar{\boldsymbol{u}})\, \mathrm{d}\boldsymbol{x}.$$

Terms involving $\boldsymbol{u}_h$ and p_h should be moved to the LHS. To make the problem symmetric, the Dirichlet condition $\boldsymbol{u} = \bar{\boldsymbol{u}}$ is weighted by $\boldsymbol{\sigma}(\boldsymbol{v}_h, q_h)$. The final method consists of adding

$$-\int_\Gamma \boldsymbol{v}_h \cdot \boldsymbol{\sigma}(\boldsymbol{u}_h, p_h) - \int_\Gamma \boldsymbol{u}_h \cdot \boldsymbol{\sigma}(\boldsymbol{v}_h, q_h) + \int_\Gamma \bar{\boldsymbol{u}} \cdot \boldsymbol{\sigma}(\boldsymbol{v}_h, q_h)$$
$$+ \int_\Gamma \boldsymbol{v}_h \cdot \frac{\alpha}{h}(\boldsymbol{u}_h - \bar{\boldsymbol{u}})\, \mathrm{d}\boldsymbol{x}.$$

Nitsche's method

If we define the operator

$$\boldsymbol{\sigma}(\boldsymbol{v}, q) := -2\boldsymbol{n} \cdot \mu \nabla^S \boldsymbol{v} + q\boldsymbol{n},$$

the term to be added to the right-hand-side of the discrete equations is

$$-\int_\Gamma \boldsymbol{v}_h \cdot \boldsymbol{\sigma}(\boldsymbol{u}_h, p_h) + \int_\Gamma \boldsymbol{v}_h \cdot \frac{\alpha}{h}(\boldsymbol{u}_h - \bar{\boldsymbol{u}})\, \mathrm{d}\boldsymbol{x}.$$

Terms involving $\boldsymbol{u}_h$ and p_h should be moved to the LHS. To make the problem symmetric, the Dirichlet condition $\boldsymbol{u} = \bar{\boldsymbol{u}}$ is weighted by $\boldsymbol{\sigma}(\boldsymbol{v}_h, q_h)$. The final method consists of adding

$$-\int_\Gamma \boldsymbol{v}_h \cdot \boldsymbol{\sigma}(\boldsymbol{u}_h, p_h) - \int_\Gamma \boldsymbol{u}_h \cdot \boldsymbol{\sigma}(\boldsymbol{v}_h, q_h) + \int_\Gamma \bar{\boldsymbol{u}} \cdot \boldsymbol{\sigma}(\boldsymbol{v}_h, q_h)$$
$$+ \int_\Gamma \boldsymbol{v}_h \cdot \frac{\alpha}{h}(\boldsymbol{u}_h - \bar{\boldsymbol{u}})\, \mathrm{d}\boldsymbol{x}.$$

Lagrange multiplier techniques

Using Lagrange multipliers to impose boundary conditions to the Navier-Stokes equations consists of adding the term

$$\int_\Gamma \boldsymbol{v}_h \cdot \boldsymbol{\lambda}_h \, \mathrm{d}\boldsymbol{x}$$

to the LHS of the momentum equation and to add the new equations for the Lagrange multipliers

$$\int_\Gamma \gamma_h \cdot (\boldsymbol{u}_h - \bar{\boldsymbol{u}}) \, \mathrm{d}\boldsymbol{x} = \mathbf{0},$$

where $\boldsymbol{\lambda}_h$ are the Lagrange multipliers of the finite element problem and γ_h their associated test functions.

Minimization of boundary errors with external degrees of freedom I

Suppose that the unknown u_h is interpolated as

$$\begin{aligned} u_h(\boldsymbol{x}) &= \sum_{a=1}^{n_{\rm in}} I_{\rm in}^a(\boldsymbol{x}) U_{\rm in}^a + \sum_{b=1}^{n_{\rm out}} I_{\rm out}^b(\boldsymbol{x}) U_{\rm out}^b \\ &= \boldsymbol{I}_{\rm in}(\boldsymbol{x}) \boldsymbol{U}_{\rm in} + \boldsymbol{I}_{\rm out}(\boldsymbol{x}) \boldsymbol{U}_{\rm out}, \end{aligned}$$

where $I_{\rm in}^a(\boldsymbol{x})$ and $I_{\rm out}^b(\boldsymbol{x})$ are the interpolation functions, $n_{\rm in}$ is the number of nodes in $\Omega_{\rm in}$, and $n_{\rm out}$ in layer L_{-1}.

The main idea is to compute $\boldsymbol{U}_{\rm out}$ by minimizing the functional

$$\begin{aligned} J_2(\boldsymbol{U}_{\rm in}, \boldsymbol{U}_{\rm out}) &= \int_\Gamma (u_h(\boldsymbol{x}) - \bar{u}(\boldsymbol{x}))^2 \\ &= \int_\Gamma (\boldsymbol{I}_{\rm in}(\boldsymbol{x}) \boldsymbol{U}_{\rm in} + \boldsymbol{I}_{\rm out}(\boldsymbol{x}) \boldsymbol{U}_{\rm out} - \bar{u}(\boldsymbol{x}))^2 . \end{aligned}$$

Minimization of boundary errors with external degrees of freedom I

Suppose that the unknown u_h is interpolated as

$$\begin{aligned} u_h(\boldsymbol{x}) &= \sum_{a=1}^{n_{\text{in}}} I_{\text{in}}^a(\boldsymbol{x}) U_{\text{in}}^a + \sum_{b=1}^{n_{\text{out}}} I_{\text{out}}^b(\boldsymbol{x}) U_{\text{out}}^b \\ &= \boldsymbol{I}_{\text{in}}(\boldsymbol{x}) \boldsymbol{U}_{\text{in}} + \boldsymbol{I}_{\text{out}}(\boldsymbol{x}) \boldsymbol{U}_{\text{out}}, \end{aligned}$$

where $I_{\text{in}}^a(\boldsymbol{x})$ and $I_{\text{out}}^b(\boldsymbol{x})$ are the interpolation functions, n_{in} is the number of nodes in Ω_{in}, and n_{out} in layer L_{-1}.

The main idea is to compute $\boldsymbol{U}_{\text{out}}$ by minimizing the functional

$$\begin{aligned} J_2(\boldsymbol{U}_{\text{in}}, \boldsymbol{U}_{\text{out}}) &= \int_\Gamma (u_h(\boldsymbol{x}) - \bar{u}(\boldsymbol{x}))^2 \\ &= \int_\Gamma (\boldsymbol{I}_{\text{in}}(\boldsymbol{x}) \boldsymbol{U}_{\text{in}} + \boldsymbol{I}_{\text{out}}(\boldsymbol{x}) \boldsymbol{U}_{\text{out}} - \bar{u}(\boldsymbol{x}))^2 . \end{aligned}$$

Minimization of boundary errors with external degrees of freedom II

Suppose now that the problem for u_h in $\Omega_{\rm in}$ leads to an algebraic equation of the form

$$\boldsymbol{K}_{\rm in,in}\boldsymbol{U}_{\rm in} + \boldsymbol{K}_{\rm in,out}\boldsymbol{U}_{\rm out} = \boldsymbol{F}_{\rm in}.$$

If this is supplemented with the equation resulting from the minimization of functional indicated, the system to be solved is

$$\begin{bmatrix} \boldsymbol{K}_{\rm in,in} & \boldsymbol{K}_{\rm in,out} \\ \boldsymbol{N}_\Gamma & \boldsymbol{M}_\Gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{\rm in} \\ \boldsymbol{U}_{\rm out} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_{\rm in} \\ \boldsymbol{f}_\Gamma \end{bmatrix},$$

where

$$\boldsymbol{M}_\Gamma = \int_\Gamma \boldsymbol{I}^{\rm t}_{\rm out}(\boldsymbol{x})\boldsymbol{I}_{\rm out}(\boldsymbol{x}), \quad \boldsymbol{f}_\Gamma = \int_\Gamma \boldsymbol{I}^{\rm t}_{\rm out}(\boldsymbol{x})\bar{u}(\boldsymbol{x}), \quad \boldsymbol{N}_\Gamma = \int_\Gamma \boldsymbol{I}^{\rm t}_{\rm out}(\boldsymbol{x})\boldsymbol{I}_{\rm in}(\boldsymbol{x}).$$

Minimization of boundary errors with external degrees of freedom II

Suppose now that the problem for u_h in $\Omega_{\rm in}$ leads to an algebraic equation of the form

$$\boldsymbol{K}_{\rm in,in}\boldsymbol{U}_{\rm in} + \boldsymbol{K}_{\rm in,out}\boldsymbol{U}_{\rm out} = \boldsymbol{F}_{\rm in}.$$

If this is supplemented with the equation resulting from the minimization of functional indicated, the system to be solved is

$$\begin{bmatrix} \boldsymbol{K}_{\rm in,in} & \boldsymbol{K}_{\rm in,out} \\ \boldsymbol{N}_\Gamma & \boldsymbol{M}_\Gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{\rm in} \\ \boldsymbol{U}_{\rm out} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_{\rm in} \\ \boldsymbol{f}_\Gamma \end{bmatrix},$$

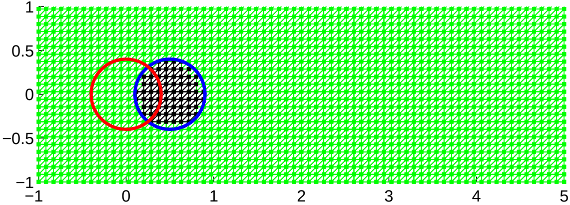
where

$$\boldsymbol{M}_\Gamma = \int_\Gamma \boldsymbol{I}^{\rm t}_{\rm out}(\boldsymbol{x})\boldsymbol{I}_{\rm out}(\boldsymbol{x}), \quad \boldsymbol{f}_\Gamma = \int_\Gamma \boldsymbol{I}^{\rm t}_{\rm out}(\boldsymbol{x})\bar{u}(\boldsymbol{x}), \quad \boldsymbol{N}_\Gamma = \int_\Gamma \boldsymbol{I}^{\rm t}_{\rm out}(\boldsymbol{x})\boldsymbol{I}_{\rm in}(\boldsymbol{x}).$$

Outline

Newly created nodes

- Key issue: calculation of temporal derivatives.
- ALE meshes: derivatives well defined.
- Fixed meshes: derivatives at newley created nodes need to be defined.



The fictitious domain method

- Equations solved on the whole background domain (including the fictitious domain).
- Boundary conditions imposed through Lagrange multipliers.
- Time derivatives at newly created nodes computed from the values of the unknowns in the fictitious domain.

The fixed-mesh ALE method I

Suppose Ω^0 is meshed with a finite element mesh M^0 and that at time t^n the domain Ω^n is meshed with a finite element mesh M^n. Let $\boldsymbol{u}^n$ be the velocity already computed on Ω^n.

1. Define $\Gamma_{\text{fluid}}^{n+1}$.
2. Deform virtually the mesh M^n to M_{virt}^{n+1} using the classical ALE concepts and compute the mesh velocity $\boldsymbol{u}_{\text{dom}}^{n+1}$.
3. Write down the ALE Navier-Stokes equations on M_{virt}^{n+1}.
4. Split the elements of M^0 cut by $\Gamma_{\text{fluid}}^{n+1}$ to define a mesh on Ω^{n+1}, M^{n+1}.
5. Project the ALE Navier-Stokes equations from M_{virt}^{n+1} to M^{n+1}.
6. Solve the equations on M^{n+1} to compute $\boldsymbol{u}^{n+1}$ and p^{n+1}.

The fixed-mesh ALE method I

Suppose Ω^0 is meshed with a finite element mesh M^0 and that at time t^n the domain Ω^n is meshed with a finite element mesh M^n. Let $\boldsymbol{u}^n$ be the velocity already computed on Ω^n.

1. Define $\Gamma_{\text{fluid}}^{n+1}$.
2. Deform virtually the mesh M^n to M_{virt}^{n+1} using the classical ALE concepts and compute the mesh velocity $\boldsymbol{u}_{\text{dom}}^{n+1}$.
3. Write down the ALE Navier-Stokes equations on M_{virt}^{n+1}.
4. Split the elements of M^0 cut by $\Gamma_{\text{fluid}}^{n+1}$ to define a mesh on Ω^{n+1}, M^{n+1}.
5. Project the ALE Navier-Stokes equations from M_{virt}^{n+1} to M^{n+1}.
6. Solve the equations on M^{n+1} to compute $\boldsymbol{u}^{n+1}$ and p^{n+1}.

The fixed-mesh ALE method I

Suppose Ω^0 is meshed with a finite element mesh M^0 and that at time t^n the domain Ω^n is meshed with a finite element mesh M^n. Let $\boldsymbol{u}^n$ be the velocity already computed on Ω^n.

1. Define $\Gamma_{\mathrm{fluid}}^{n+1}$.
2. Deform virtually the mesh M^n to M_{virt}^{n+1} using the classical ALE concepts and compute the mesh velocity $\boldsymbol{u}_{\mathrm{dom}}^{n+1}$.
3. Write down the ALE Navier-Stokes equations on M_{virt}^{n+1}.
4. Split the elements of M^0 cut by $\Gamma_{\mathrm{fluid}}^{n+1}$ to define a mesh on Ω^{n+1}, M^{n+1}.
5. Project the ALE Navier-Stokes equations from M_{virt}^{n+1} to M^{n+1}.
6. Solve the equations on M^{n+1} to compute $\boldsymbol{u}^{n+1}$ and p^{n+1}.

The fixed-mesh ALE method I

Suppose Ω^0 is meshed with a finite element mesh M^0 and that at time t^n the domain Ω^n is meshed with a finite element mesh M^n. Let $\boldsymbol{u}^n$ be the velocity already computed on Ω^n.

1. Define $\Gamma_{\text{fluid}}^{n+1}$.
2. Deform virtually the mesh M^n to M_{virt}^{n+1} using the classical ALE concepts and compute the mesh velocity $\boldsymbol{u}_{\text{dom}}^{n+1}$.
3. Write down the ALE Navier-Stokes equations on M_{virt}^{n+1}.
4. Split the elements of M^0 cut by $\Gamma_{\text{fluid}}^{n+1}$ to define a mesh on Ω^{n+1}, M^{n+1}.
5. Project the ALE Navier-Stokes equations from M_{virt}^{n+1} to M^{n+1}.
6. Solve the equations on M^{n+1} to compute $\boldsymbol{u}^{n+1}$ and p^{n+1}.

The fixed-mesh ALE method I

Suppose Ω^0 is meshed with a finite element mesh M^0 and that at time t^n the domain Ω^n is meshed with a finite element mesh M^n. Let $\boldsymbol{u}^n$ be the velocity already computed on Ω^n.

1. Define $\Gamma^{n+1}_{\text{fluid}}$.
2. Deform virtually the mesh M^n to M^{n+1}_{virt} using the classical ALE concepts and compute the mesh velocity $\boldsymbol{u}^{n+1}_{\text{dom}}$.
3. Write down the ALE Navier-Stokes equations on M^{n+1}_{virt}.
4. Split the elements of M^0 cut by $\Gamma^{n+1}_{\text{fluid}}$ to define a mesh on Ω^{n+1}, M^{n+1}.
5. Project the ALE Navier-Stokes equations from M^{n+1}_{virt} to M^{n+1}.
6. Solve the equations on M^{n+1} to compute $\boldsymbol{u}^{n+1}$ and p^{n+1}.

The fixed-mesh ALE method I

Suppose Ω^0 is meshed with a finite element mesh M^0 and that at time t^n the domain Ω^n is meshed with a finite element mesh M^n. Let $\boldsymbol{u}^n$ be the velocity already computed on Ω^n.

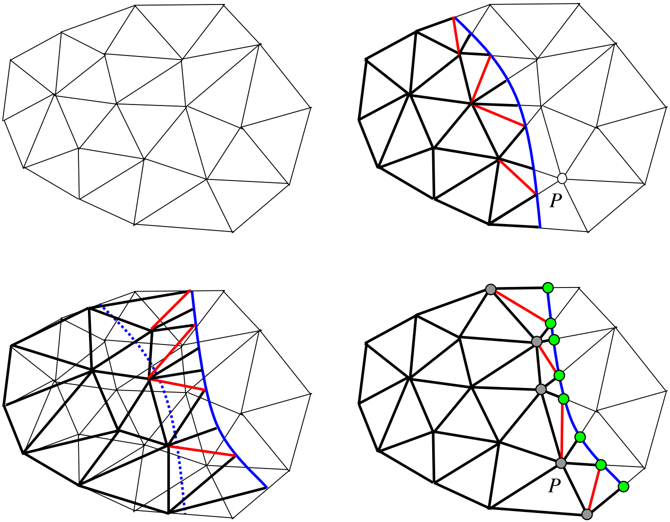
1. Define $\Gamma_{\mathrm{fluid}}^{n+1}$.
2. Deform virtually the mesh M^n to M_{virt}^{n+1} using the classical ALE concepts and compute the mesh velocity $\boldsymbol{u}_{\mathrm{dom}}^{n+1}$.
3. Write down the ALE Navier-Stokes equations on M_{virt}^{n+1}.
4. Split the elements of M^0 cut by $\Gamma_{\mathrm{fluid}}^{n+1}$ to define a mesh on Ω^{n+1}, M^{n+1}.
5. Project the ALE Navier-Stokes equations from M_{virt}^{n+1} to M^{n+1}.
6. Solve the equations on M^{n+1} to compute $\boldsymbol{u}^{n+1}$ and p^{n+1}.

The fixed-mesh ALE method I

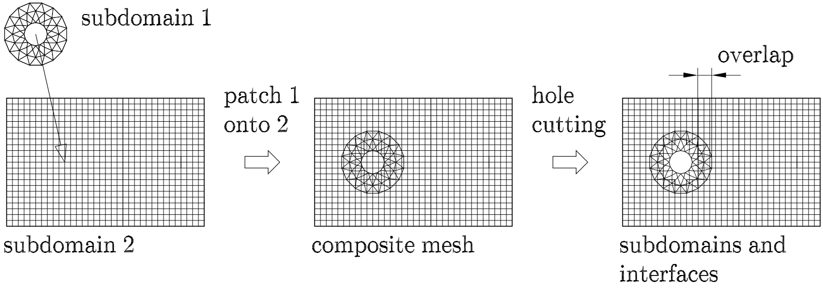
Suppose Ω^0 is meshed with a finite element mesh M^0 and that at time t^n the domain Ω^n is meshed with a finite element mesh M^n. Let $\boldsymbol{u}^n$ be the velocity already computed on Ω^n.

1. Define $\Gamma_{\text{fluid}}^{n+1}$.
2. Deform virtually the mesh M^n to M_{virt}^{n+1} using the classical ALE concepts and compute the mesh velocity $\boldsymbol{u}_{\text{dom}}^{n+1}$.
3. Write down the ALE Navier-Stokes equations on M_{virt}^{n+1}.
4. Split the elements of M^0 cut by $\Gamma_{\text{fluid}}^{n+1}$ to define a mesh on Ω^{n+1}, M^{n+1}.
5. Project the ALE Navier-Stokes equations from M_{virt}^{n+1} to M^{n+1}.
6. Solve the equations on M^{n+1} to compute $\boldsymbol{u}^{n+1}$ and p^{n+1}.

The fixed-mesh ALE method II



Chimera strategies



The problem on the composite grid is solved using a domain decomposition strategy.

Outline

Summary I

Approximate imposition of Dirichlet boundary conditions:

- Introduction of forces on the boundaries.
- Penalization of the boundary conditions, including Nitsche's method.
- Use of Lagrange multipliers to enforce boundary conditions.
- Use of inactive degrees of freedom to optimize the imposition of boundary conditions.

Summary I

Approximate imposition of Dirichlet boundary conditions:

- Introduction of forces on the boundaries.
- Penalization of the boundary conditions, including Nitsche's method.
- Use of Lagrange multipliers to enforce boundary conditions.
- Use of inactive degrees of freedom to optimize the imposition of boundary conditions.

Summary I

Approximate imposition of Dirichlet boundary conditions:

- Introduction of forces on the boundaries.
- Penalization of the boundary conditions, including Nitsche's method.
- Use of Lagrange multipliers to enforce boundary conditions.
- Use of inactive degrees of freedom to optimize the imposition of boundary conditions.

Summary I

Approximate imposition of Dirichlet boundary conditions:

- Introduction of forces on the boundaries.
- Penalization of the boundary conditions, including Nitsche's method.
- Use of Lagrange multipliers to enforce boundary conditions.
- Use of inactive degrees of freedom to optimize the imposition of boundary conditions.

Summary I

Approximate imposition of Dirichlet boundary conditions:

- Introduction of forces on the boundaries.
- Penalization of the boundary conditions, including Nitsche's method.
- Use of Lagrange multipliers to enforce boundary conditions.
- Use of inactive degrees of freedom to optimize the imposition of boundary conditions.

Summary II

Treatment of time dependent domains:

- Solving in the whole physical domain, both the solid and the fluid, with an appropriate modeling of the forces at the interface. This includes the original version of the immersed boundary method.
- Fictitious domain method, in which the solid is considered to be filled with a fictitious fluid and boundary conditions are imposed through Lagrange multipliers.
- Fixed-mesh ALE method, based on the classical ALE approach but projecting the equations always to a fixed mesh.
- Chimera strategies based on domain decomposition method, which at least allow to remove the mesh deformation due to rigid body motions of the solids inside the fluid.

Summary II

Treatment of time dependent domains:

- Solving in the whole physical domain, both the solid and the fluid, with an appropriate modeling of the forces at the interface. This includes the original version of the immersed boundary method.
- Fictitious domain method, in which the solid is considered to be filled with a fictitious fluid and boundary conditions are imposed through Lagrange multipliers.
- Fixed-mesh ALE method, based on the classical ALE approach but projecting the equations always to a fixed mesh.
- Chimera strategies based on domain decomposition method, which at least allow to remove the mesh deformation due to rigid body motions of the solids inside the fluid.

Summary II

Treatment of time dependent domains:

- Solving in the whole physical domain, both the solid and the fluid, with an appropriate modeling of the forces at the interface. This includes the original version of the immersed boundary method.
- Fictitious domain method, in which the solid is considered to be filled with a fictitious fluid and boundary conditions are imposed through Lagrange multipliers.
- Fixed-mesh ALE method, based on the classical ALE approach but projecting the equations always to a fixed mesh.
- Chimera strategies based on domain decomposition method, which at least allow to remove the mesh deformation due to rigid body motions of the solids inside the fluid.

Summary II

Treatment of time dependent domains:

- Solving in the whole physical domain, both the solid and the fluid, with an appropriate modeling of the forces at the interface. This includes the original version of the immersed boundary method.
- Fictitious domain method, in which the solid is considered to be filled with a fictitious fluid and boundary conditions are imposed through Lagrange multipliers.
- Fixed-mesh ALE method, based on the classical ALE approach but projecting the equations always to a fixed mesh.
- Chimera strategies based on domain decomposition method, which at least allow to remove the mesh deformation due to rigid body motions of the solids inside the fluid.

Summary II

Treatment of time dependent domains:

- Solving in the whole physical domain, both the solid and the fluid, with an appropriate modeling of the forces at the interface. This includes the original version of the immersed boundary method.
- Fictitious domain method, in which the solid is considered to be filled with a fictitious fluid and boundary conditions are imposed through Lagrange multipliers.
- Fixed-mesh ALE method, based on the classical ALE approach but projecting the equations always to a fixed mesh.
- Chimera strategies based on domain decomposition method, which at least allow to remove the mesh deformation due to rigid body motions of the solids inside the fluid.

THANK YOU!