Moving boundary problems with Freefem++

B. Maury

Laboratoire de Mathématiques d'Orsay, Univ. Paris-Sud 11

Jussieu, december 2011

Two main strategies

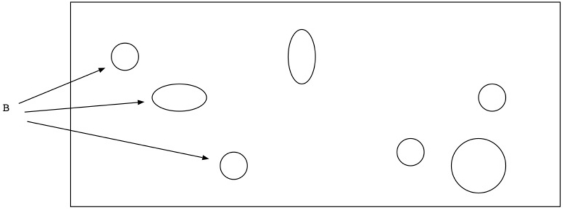
1. Fictitious domain approach.

   The moving domain is embedded in a larger, fixed one. A cartesian mesh can be used.

2. Arbitrary Lagrangian Eulerian approach.

   Moving, conforming mesh.

Rigid bodies in a fluid



$$\left\{ \begin{array}{rcll} \rho\left(\dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - \mu\Delta\mathbf{u} + \nabla p & = & 0 & \text{in} \quad \Omega\setminus\bar{B}, \\ \nabla\cdot\mathbf{u} & = & 0 & \text{in} \quad \Omega\setminus\bar{B}, \\ \mathbf{u} & = & \mathbf{v}_i + \omega_i\times(\mathbf{x}-\mathbf{x}_i) & \text{on} \quad \partial B_i, \end{array} \right.$$

$$\left\{ \begin{array}{rcl} m_i\dfrac{d\mathbf{v}_i}{dt} & = & \displaystyle\int_{B_i}\mathbf{f}_b^i - \int_{\partial B_i}\boldsymbol{\sigma}\cdot\mathbf{n} \\ J_i\dfrac{d\omega_i}{dt} & = & \displaystyle\int_{\partial B_i}(\mathbf{x}-\mathbf{x}_i)\times\boldsymbol{\sigma}\cdot\mathbf{n} \end{array} \right.$$

Stokes problem

Variational framework

$$\left\{\begin{array}{rcll} -\mu\Delta\mathbf{u} + \nabla p &=& 0 & \text{in} \quad \Omega\setminus\bar{B}, \\ \nabla\cdot\mathbf{u} &=& 0 & \text{in} \quad \Omega\setminus\bar{B}, \\ \mathbf{u} &=& \mathbf{v}_i + \omega_i\times(\mathbf{x}-\mathbf{x}_i) & \text{on} \quad \partial B_i, \end{array}\right.$$

$$\left\{\begin{array}{rcl} \mathbf{F}_i - \displaystyle\int_{\partial B_i}\boldsymbol{\sigma}\cdot\mathbf{n} &=& 0 \\ \displaystyle\int_{\partial B_i}(\mathbf{x}-\mathbf{x}_i)\times\boldsymbol{\sigma}\cdot\mathbf{n} &=& 0 \end{array}\right.$$

It amounts to minimize

$$\frac{1}{4}\int_\Omega\left|\nabla\mathbf{v} +{}^t\nabla\mathbf{v}\right|^2 - \int_\Omega \mathbf{f}\cdot\mathbf{v}$$

over

$$K = \left\{\mathbf{v}\in H^1_0\,,\ \nabla\cdot\mathbf{u} = 0\,,\ \nabla\mathbf{v} +{}^t\nabla\mathbf{v} = 0 \text{ on } B\right\}$$

$\rightarrow$ the Finite Element framework allows to bypass the direct estimation of $\boldsymbol{\sigma}\cdot\mathbf{n}$.

Penalty method

Penalty method (P. Angot, C.H. Bruneau, A. Iollo, J.P. Caltagirone, S. Vincent, P. Peyla, A. Lefebvre, B.M. ...)

Small parameter $\varepsilon > 0$ ("viscosity" $1/\varepsilon$ for the solid phase)

Unconstrained minimization of

$$\frac{\mu}{4}\int_\Omega \left|\nabla \mathbf{v} + {}^t\nabla \mathbf{v}\right|^2 - \int_\Omega \mathbf{f}\cdot\mathbf{v} + \frac{1}{2\varepsilon}\int_B \left|\nabla \mathbf{v} + {}^t\nabla \mathbf{v}\right|^2$$

over

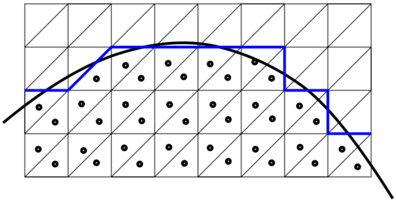
$$K = \left\{\mathbf{v} \in H^1_0\,,\ \nabla\cdot\mathbf{u} = 0\right\}$$

Implementation

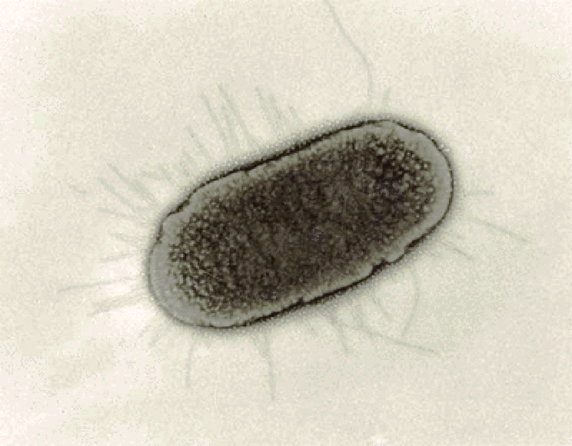
Characteristic function `chiB` defined as a P^0 function.

```
....
int2d(Th)(mu*(2*dx(u1)*dx(v1)+dy(u1)*dy(v1)+dx(u2)*dx(v2)
+2*dy(u2)*dy(v2)+dy(u1)*dx(v2)+dx(u2)*dy(v1)))
....
+int2d(Th)((2*dx(u1)*dx(v1)+dy(u1)*dy(v1)+dx(u2)*dx(v2)
+2*dy(u2)*dy(v2)+dy(u1)*dx(v2)+dx(u2)*dy(v1))*chiB/eps)
```

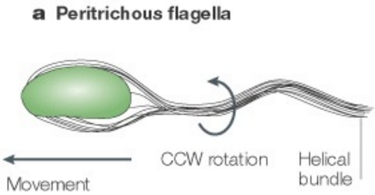
...

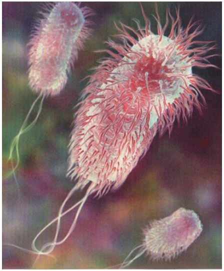


Applications : E.Coli (with A. Decoene and S. Martin)

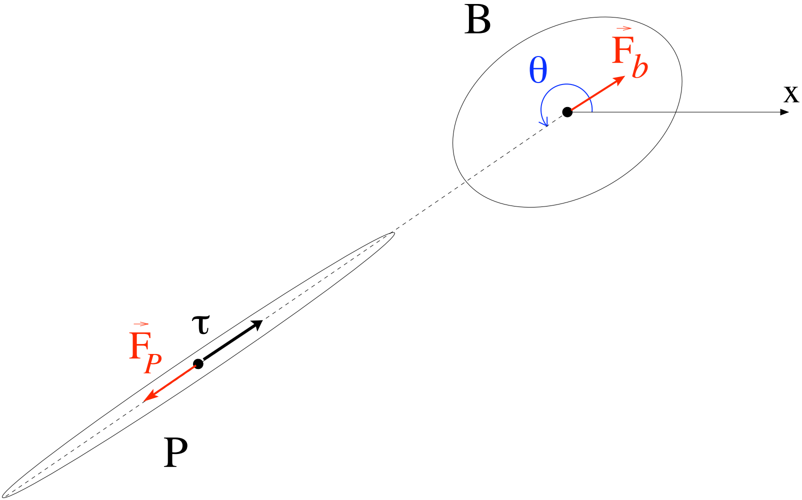


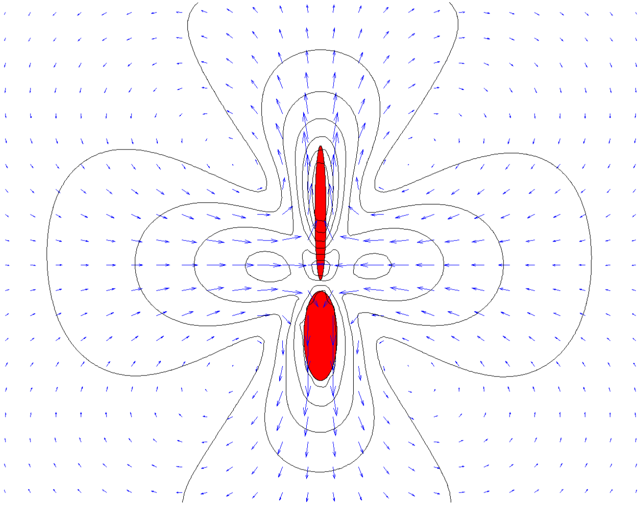
E. Coli (picture taken at 6 :47 am, before shower and breakfeast)





$\longrightarrow$ capable of swimming several times its own size per second


B
θ
$\vec{F}_b$
x
τ
$\vec{F}_P$
P



Danse of the E. Coli's

Apparent viscosity of pusher suspensions

Dense suspension, biperiodic case

625 bacteria, biperiodic, weak turbulence

Squirmers (with N. Aguilon, A. Decoene, B. Fabrêges)

Velocity field in object B :

$$\mathbf{u} = \mathbf{u}_0 + \text{ Rigid motion}$$

Penalty approach

$$+\frac{1}{2\varepsilon}\int_B \left|\nabla(\mathbf{u}-\mathbf{u}_0)+\nabla(\mathbf{u}-\mathbf{u}_0)^T\right|^2.$$

200 squirmers

Interaction Force

A little bit of theory :

$$K \subset V\ ,\ u = \arg\min_K \left(\frac{1}{2} \langle Av\ , v \rangle - \langle \varphi\ , v \rangle \right) \Longleftrightarrow Au + \xi = \varphi\ ,\ \xi \in K^\perp .$$

Penalty method :

$$u_\varepsilon = \arg\min_V \left(\frac{1}{2} \langle Av\ , v \rangle - \langle \varphi\ , v \rangle + \frac{1}{2\varepsilon} b(v,v) \right)\ , \quad \text{with } K = \ker b(\cdot,\cdot).$$

It holds

$$\frac{1}{\varepsilon} b(u_\varepsilon, \cdot) \longrightarrow \xi \quad \text{ in } V'.$$

Application : the linear functionnal

$$\mathbf{v} \longmapsto \frac{\mu}{2} \int_B \left(\nabla \mathbf{u}_\varepsilon + {}^t \nabla \mathbf{u}_\varepsilon \right) : \left(\nabla \mathbf{v} + {}^t \nabla \mathbf{v} \right)$$

can be interpreted (in the sense of distribution) as the force exerted by the fluid on the object.

Example of application : **Turbine** (joint work with S. del Pino)

Other example : fluid structure interaction

Quasi static structure (linear elasticity), Stokes or Navier Stokes fluid

$$-\Delta \mathbf{u}_\varepsilon + \nabla p + \frac{1}{\varepsilon} \mathbb{1}_B \mathbf{u}_\varepsilon = 0.$$

In the structure :

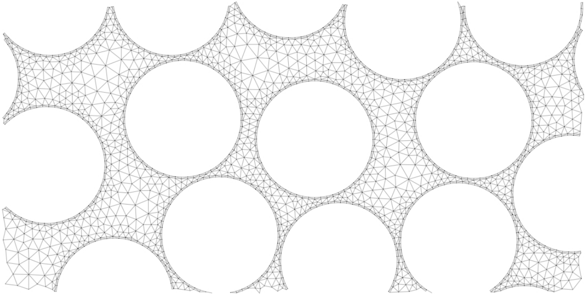
$$-\nabla \cdot \Big(\mu(\nabla \xi + \nabla \xi^T) + \lambda \, (\nabla \cdot \xi) \, I_d \Big) = \frac{1}{\varepsilon} \mathbf{u}_\varepsilon$$

Half-disc elastic obstacle

ALE approach

(T. Tezduyar, H. Hu, B.M., T. Coupez, ...)

The mesh follows the motion of the bodies or the free boundary



Principle of the method

The domain (i.e. the mesh) moves with a velocity $\mathbf{c}$.

$\Phi(x,t;\tau)$ associated flow :

$$\partial_t \Phi(x,t;\tau) = \mathbf{c}(\Phi(x,t;\tau)), \ \ \Phi(x,\tau;\tau) = x.$$

ALE velocity $\mathbf{u}_\tau$, defined in $\Omega(\tau)$ by

$$\mathbf{u}_\tau(x,t) = \mathbf{u}(\Phi(x,t;\tau),t).$$

$$\left(\frac{\partial \mathbf{u}_\tau}{\partial t} + (\mathbf{u}_\tau - \mathbf{c}_\tau)\cdot\nabla\mathbf{u}_\tau\right) - \mu\Delta\mathbf{u}_\tau \ + \nabla p_\tau = 0 \quad \text{in} \quad \Omega(\tau),$$

at $t=\tau$, valid at the first order in $t-\tau$.

Mesh velocity and Implementation issues

Mesh velocity : $\mathbf{c} \cdot \mathbf{n} = \mathbf{u} \cdot \mathbf{n}$ on the free boundary

Inside the domain : arbitrary

Example : Laplace problem on each component, possibly non isotropic

Key point : numerical push-forward.

In Freefem++, if `uh` is defined on the mesh, and the mesh is moved, the next time `uh` is called it is automatically *interpolated*

```
MeshVeloc ; // computation of mesh velocity (cx,cy)

Th = movemesh(Th,[x+dt*cx,y+dt*cy]); // The mesh is moved

tmp=ux[]; pux=0; pux[]=tmp ;  // the variable are pushed forward
```

Example

Oscillating cylinder (with A. Decoene, B. Semin)

Experiment

Velocity field

Laplacian

Non isotropic Laplacian

Piecewise constant Interpolation

Zoom

Water waves

Jet (with buckling)

Jet (Oscilating inlet)