Variational Methods for Computational Fluid Dynamics
Année 2013 - 2014, X 2011.

PC 3 (TP)

The aim of this practical session is to simulate moving domains filled of fluids. We first consider the case of a low Reynolds number flow, and then study a free surface flow with the full Navier-Stokes equations.

Exercise 1. Training.

Take a 2d meshed domain (e.g. a square). Construct a second mesh deduced from the first by deforming it. Use for that the `move mesh` FreeFem++ command, and try various deformation functions.

```
mTh = movemesh(Th,[x+vx,y+vy]);
```

constructs a new mesh `mTh` obtained from the `Th` by displacing the nodes along the vector field (here `(vx,vy)`).

Exercise 2. The melting loukoum.

We consider the Stokes problem

$$\left[\begin{array}{l} -\Delta \mathbf{u} + \nabla p = \mathbf{f}\,, \\ \operatorname{div}\, \mathbf{u} = 0\,, \end{array}\right.$$

on a domain which is initially a square. At initial time the domain of fluid is supposed to touch a flat floor. All the bottom part is assumed to be fixed while the three other parts of the boundary are free surfaces.

1. Solve the previous Stokes problem where $\mathbf{f}$ is the gravity, and assuming Neumann boundary condition on the free surfaces that can be either

   $$\mu \nabla \mathbf{u} \cdot \mathbf{n} - \mathrm{p}\mathbf{n} = -P_{ext}\mathbf{n}\,,$$

   or

   $$\mu \left(\nabla \mathbf{u} + {}^t\nabla \mathbf{u} \right) \cdot \mathbf{n} - \mathrm{p}\mathbf{n} = -P_{ext}\mathbf{n}\,.$$

   (Notice that without restriction, one can take $P_{ext} = 0$.) Do both a P^1 bubble / P^1, and a P^2/P^1 approximation.

2. Make the mesh move according to the velocity found in the preceding step. Use for this the `movemesh` FreeFem++command and the deformation vector `(dt*ux,dt*uy)`.

3. Notice the difference of the results of the simulation according to the Neumann boundary condition used.

4. Notice also that eventually the simulation breaks down after a while due to a poor mesh and badly oriented triangles.

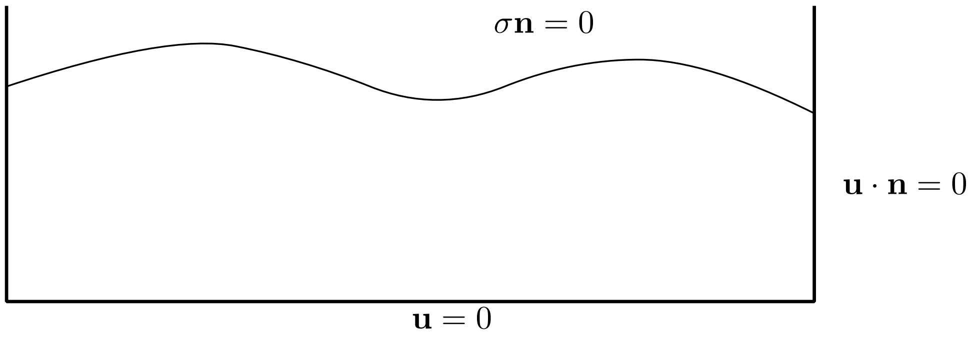

Figure 1: Fluid in a container.

Exercise 3. ALE for wave motion.

We consider a fluid filling a glass. The situation depicted in the following picture implies that the fluid stays in the domain

$$\Omega = \{(x, y) \in \mathbb{R}^2 \,, |\, 0 < x < L \,,\, 0 < y\} \,.$$

1. Write a Navier-Stokes solver using the method of characteristics for the convection. Use for this the FreeFem++command `convect` and adding in the variational formulation the following line:

```
+int2d(Th)(-convect([pux,puy],-dt,pux)*tux -convect([pux,puy],-dt,puy)*tuy)
```

   where `[pux,puy]` is the previous velocity. The argument `-dt` means that the field is integrated backward in time during δt. Use also the boundary conditions:

   - $\mathbf{u} = 0$ at the bottom $y = 0$ part of the fluid domain ;
   - $u_1 = 0$ and $\sigma_{21} = 0$ at the left and right parts of the boundary($x = 0$ and $x = L$). (The fluid will be allowed to slip on the boundary, but only vertically. Notice that σ_{21} is also $\boldsymbol{\sigma}\mathbf{n}\cdot\mathbf{t}$ on these boundaries.
   - $\boldsymbol{\sigma}\mathbf{n} = -P_{ext}\mathbf{n} = 0$ on the free surface.

   Use also the parameters: $\mu = 0.001$, $\rho = 1$, $g = 5$, $L = 6$, $\delta t = 0.02$ and the initial profile is given by

$$h(x) = 1 + \frac{1}{2}\exp\left(-\left(\frac{x-3}{0.7}\right)^2\right) .$$

2. Write a *Lagrangian* solver for the free surface. That means move the mesh according to the computed velocity $\mathbf{u}$ as for the previous exercise. Show that the computation does not last for a long time.

3. Implement an ALE approach. In that aim, one can move the mesh only vertically according to a vertical displacement $\mathbf{c} = (0, c_y)$. The only constraint for c_y is that it must be consistent (which means one has

$$\mathbf{c}\cdot\mathbf{n} = \mathbf{u}\cdot\mathbf{n} \text{ on the boundary.}$$

A possible choice (though not the unique one) is to solve

$$\begin{cases} \Delta c_y = 0\,, \\ c_y = \dfrac{\mathbf{u}\cdot\mathbf{n}}{n_y} \text{ on the free surface and on the bottom}\,, \\ \dfrac{\partial c_y}{\partial \mathbf{n}} = 0 \text{ everywhere else.} \end{cases}$$

4. Move the mesh according to the vector field $(0, c_y)$ and transport all quantities on the new mesh before solving again Navier-Stokes equations. This is done using the following FreeFem++ syntax

```
real[int] tmp(ux[].n); //tmp is an array of the same size than ux

tmp=ux[]; pux=0; pux[]=tmp ;
tmp=uy[]; puy=0; puy[]=tmp ;
tmp=cx[]; cx=0; cx[]=tmp ;
tmp=cy[]; cy=0; cy[]=tmp ;
```

These complicated commands are meant to transport the old variables on the new mesh. Strictly speaking they do nothing (copy the variable into a temporary array and then copy it back). However if one does not do this kind of command, FreeFem++would by default interpolate the old variable on the new mesh which is not what is desired.