Nečas Center for Mathematical Modeling

Numerical solution of continuum physics problems with FEniCS or how to use FEM and not vary (too much) about programming ...

Jaroslav Hron

Mathematical Institute, Charles University in Prague

CHARLES UNIVERSITY PRAGUE

faculty of mathematics and physics

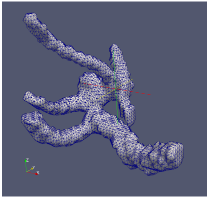
- Lectures and example downloadable at
  `http://www.karlin.mff.cuni.cz/~hron/warsaw_2014/`

ex1 - Fluid-structure interaction (JH)

ex2 - Fluid-structure interaction (JH, M. Mádlík

ex3 - Cahn-Hilliard 3-phase flow (M. Řehoř)

ex4 - Blood flow in aneurysm (H. Švihlová)



Numerical solution of continuum physics problems with FEniCS or how to use FEM and not vary (too much) about programming ...

- ✓ **lecture 1** introduction to FEniCS/python, FEM and how to solve laplace equation
- ✓ **lecture 2** boundary conditions, time discretization, convection-diffusion equation, (stabilization)
- ✓ **lecture 3** Stokes, incompressible Navier-Stokes equations
- ✓ **lecture 4** linear and non-linear elasticity
- ☞ **lecture 5** level-set method, etc.

Multiphase flow - levelset method

$$\begin{aligned}
\varrho(\frac{\partial \mathbf{v}}{\partial t} + [\nabla \mathbf{v}]\mathbf{v}) - \operatorname{div}(\boldsymbol{\sigma}) &= \varrho \mathbf{f} && \text{in } \Omega \\
\operatorname{div} \mathbf{v} &= 0 && \text{in } \Omega \\
\mathbf{v} &= v_D && \text{on } \Gamma_D \\
[\nabla \mathbf{v}]\mathbf{n} &= \mathbf{0} && \text{on } \Gamma_N
\end{aligned}$$

where $\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{D}$ is the stress tensor and $\mathbf{D} = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)$.

- add levelset function ϕ to track material interface

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \mathbf{v} = 0$$

  such that $\phi > 0$ in material one and $\phi < 0$ for material two.
- material properties

$$\mu = \begin{cases} \mu_0 & if \phi > 0 \\ \mu_1 & if \phi < 0 \end{cases} \qquad \varrho = \begin{cases} \varrho_0 & if \phi > 0 \\ \varrho_1 & if \phi < 0 \end{cases}$$

```
def rho(l):
   return(rho0 * 0.5* (1.0+ sign(l)) + rho1 * 0.5*(1.0 - sign(l)))

def nu(l):
   return(nu0 * 0.5* (1.0+ sign(l)) + nu1 * 0.5*(1.0 - sign(l)))
```

Multiphase flow - levelset method

$$
\begin{aligned}
\varrho\left(\frac{\partial \mathbf{v}}{\partial t} + [\nabla \mathbf{v}]\mathbf{v}\right) - \operatorname{div}(\boldsymbol{\sigma}) &= \varrho \mathbf{f} && \text{in } \Omega \\
\operatorname{div} \mathbf{v} &= 0 && \text{in } \Omega \\
\mathbf{v} &= v_D && \text{on } \Gamma_D \\
[\nabla \mathbf{v}]\mathbf{n} &= \mathbf{0} && \text{on } \Gamma_N
\end{aligned}
$$

where $\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{D}$ is the stress tensor and $\mathbf{D} = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)$.

- levelset function ϕ to track material interface

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \mathbf{v} = 0$$

- if the function ϕ is the "distance" function, i.e. $|\nabla \phi| = 1$ then unit normal to the interface is exactly $\mathbf{n} = \nabla \phi$, curvature $\kappa = -\operatorname{div} \mathbf{n}$
- the property of "distance" is lost in the evolution - need reinitialization
- surface tension force:

$$\mathbf{f} = c\kappa\mathbf{n}\delta(\phi) = c\operatorname{div}(\mathbf{I} - \mathbf{n}\otimes\mathbf{n})\delta(\phi)$$

```
def Delta(eps,q):
        tmp=(1.0/eps)*0.5*(1.0+cos(3.14159*q/eps))
    return conditional(lt(abs(q),eps),tmp,Constant(0.0))
```

Multiphase flow - levelset method

- levelset function ϕ to track material interface

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \mathbf{v} = 0$$

- Pure transport equation for the levelset - in FEM needs some stabilization (IP - interior penalty, SUPG - streamline upwind Petrov Galerkin), for IP add:

$$\int_{\text{all facets}} \alpha h^2 \, \mathrm{jump}(\nabla \phi \cdot \mathbf{n}) \, \mathrm{jump}(\nabla \bar{\phi} \cdot \mathbf{n}) \, \mathrm{d}S$$

```
n = FacetNormal(mesh)
h = CellSize(mesh)
h_avg = (h('+') + h('-'))/2.0
alpha=Constant(0.1)
IP = alpha('+')*h_avg*h_avg*inner(jump(grad(l),n), jump(grad(l_),n))*dS
```

Task no.5 - Rising bubble benchmark

1. start with the example code `lecture5/bubble_ex.py`
2. add reinitialization, see `http://link.springer.com/article/10.1007/s00607-012-0259-z/fulltext.html`
3. add surface tension, see for example `http://www.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2006/rapporter06/lindbo_dag_06153.pdf`
4. compare result with benchmark of 2D rising bubble `http://www.featflow.de/en/benchmarks/cfdbenchmarking/bubble.html`

Final report

1. write down the mathematical formulation of your problem
   - the PDE, boundary conditions, inital conditions
   - write down a well posed weak form of your equation
2. use FEM and FEniCS to implement some solution algorithm, describe the main steps
   - mesh specification
   - finite element spaces used, time scheme used
   - weak form of your problem
   - type of nonlinear and linear solvers use, with precise stopping criterias
3. specify the value of interest X, i.e. some quantity which is computed and can be investigated with respect to precision
   - some solution norm or physical quantity like force, heat flux
4. test the numerical algorithm for discretization robustness, i.e.
   - solve the problem for set of succesively refined meshes with h,$2h$,$4h$,...
   - compare different time discetizations (1st vs 2nd order)
   - solve the problem for a set of timesteps dt,$2dt$,$4dt$,.. (at least 3)
   - report the change of the value of interest $X_{h,dt}$ with respect to the discretization parameters (mesh h, dt)
5. estimate the order of convergence, i.e. α, β such that $X(h) = O(h^{\alpha})$ and/or $X(dt) = O(h^{\beta})$