# FEniCS Course

## Lecture 9: Incompressible Navier–Stokes equations

*Contributors*
Anders Logg
André Massing

# The incompressible Navier–Stokes equations

$$
\begin{aligned}
\dot{u} + u \cdot \nabla u - \nu \Delta u + \nabla p &= f && \text{in } \Omega \times (0, T] \\
\nabla \cdot u &= 0 && \text{in } \Omega \times (0, T] \\
u &= g_{\mathrm{D}} && \text{on } \Gamma_{\mathrm{D}} \times (0, T] \\
\nu \frac{\partial u}{\partial n} - pn &= g_{\mathrm{N}} && \text{on } \Gamma_{\mathrm{N}} \times (0, T] \\
u(\cdot, 0) &= u_0 && \text{in } \Omega
\end{aligned}
$$

- $u$ is the fluid velocity and $p$ is the pressure divided by the density $\rho$
- $\nu = \mu/\rho$ is the kinematic viscosity, $\mu$ dynamic viscosity
- $f$ is a given body force per unit mass
- $g_{\mathrm{D}}$ is a given boundary velocity
- $g_{\mathrm{N}}$ is a given boundary function for the natural boundary condition
- $u_0$ is a given initial velocity

# Variational problem

Multiply the momentum equation by a test function $v$ and integrate by parts:

$$\int_\Omega (\dot{u} + u \cdot \nabla u) \cdot v \, \mathrm{d}x + \nu \int_\Omega \nabla u : \nabla u \, \mathrm{d}x - \int_\Omega p \nabla \cdot v \, \mathrm{d}x$$
$$= \int_\Omega f \cdot v \, \mathrm{d}x + \int_{\Gamma_{\mathrm{N}}} g_{\mathrm{N}} \cdot v \, \mathrm{d}s$$

Short-hand notation:

$$(\dot{u} + u \cdot \nabla u, v) + \nu(\nabla u, \nabla v) - (p, \nabla \cdot v) = (f, v) + (g_{\mathrm{N}}, v)_{\Gamma_{\mathrm{N}}}$$

Multiply the continuity equation by a test function $q$ and sum up: find $(u, p) \in V$ such that

$$(\dot{u} + u \cdot \nabla u, v) + \nu(\nabla u, \nabla v) - (p, \nabla \cdot v) - (q, \nabla \cdot u)$$
$$= (f, v) + (g_{\mathrm{N}}, v)_{\Gamma_{\mathrm{N}}}$$

for all $(v, q) \in \hat{V}$

# Discrete mixed variational form of Navier–Stokes

Time-discretization leads to a *saddle-point* problem on each time step:

$$\begin{bmatrix} M + \Delta t A + \Delta t N(U) & \Delta t B \\ \Delta t B^\top & 0 \end{bmatrix} \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

- Efficient solution of the saddle-point problem relies on the efficiency of special-purpose preconditioners (Uzawa iteration, Schur complement preconditioners, ...)
- We will use another approach (simpler and often more efficient)

# The classical Chorin-Teman projection method

Step 1: Compute *tentative velocity* $u^\star$ solving

$$
\begin{aligned}
&\frac{u^\star - u^n}{\Delta t} - \nu \Delta u^\star + (u^* \cdot \nabla) u^{**} = f^{n+1} && \text{in } \Omega \\
&u^\star = g_D && \text{on } \Omega_D \\
&\frac{\partial u^\star}{\partial n} = 0 && \text{on } \Omega_N
\end{aligned}
$$

Step 2: Compute a *corrected* velocity $u^{n+1}$ and a *new* pressure $p^{n+1}$ solving

$$
\begin{aligned}
&\frac{u^{n+1} - u^\star}{\Delta t} + \nabla p^{n+1} = 0 && \text{in } \Omega \\
&\nabla \cdot u^{n+1} = 0 && \text{in } \Omega \\
&u^{n+1} \cdot n = 0 && \text{on } \partial\Omega
\end{aligned}
$$

# Computing the tentative velocity

In principle, the term $(u^* \cdot \nabla)u^{**}$ can be approximated in several ways

- Explicit: $u^* = u^{**} = u^n \Rightarrow$ diffusion-reaction equation
- Semi-implicit $u^* = u^n$ and $u^{**} = u^{n+1} \Rightarrow$ convection-diffusion-reaction equation
- Fully-implicit $u^* = u^{**} = u^{n+1}$ retaining the basic non-linearity in the Navier-Stokes equations

The natural outflow condition $\nu\partial_n u - pn = 0$ is artificially enforced by requiring

- $\partial_n u^\star = 0 \quad$ on $\partial\Omega_N$ in step 1
- $p^{n+1} = 0 \quad$ on $\partial\Omega_N$ in step 2

# Solving the projection step

Applying $\nabla\cdot$ to $\dfrac{u^{n+1} - u^\star}{\Delta t} + \nabla p^{n+1} = 0$ and using requirement $\nabla \cdot u^{n+1} = 0$ yields

$$\Delta p^{n+1} = \frac{1}{\Delta t} \nabla \cdot u^\star \quad \text{in } \Omega$$

We already required

$$p = 0 \quad \text{on } \partial\Omega_N$$

Multiplying $\dfrac{u^{n+1} - u^\star}{\Delta t} + \nabla p^{n+1} = 0$ with $n$ and restricting to $\partial\Omega_D$ gives

$$\frac{\partial p^{n+1}}{\partial n} = 0 \quad \text{on } \partial\Omega_D$$

Compute $u^{n+1}$ by

$$u^{n+1} = u^\star - \Delta t \nabla p^{n+1}$$

including boundary conditions for $u$ at $t = t^{n+1}$

# Chorin-Teman projection method – Summary

1. Compute tentative velocity $u^\bigstar$ by

$$(\frac{u^\bigstar - u^n}{\Delta t}, v) + ((u^* \cdot \nabla)u^{**}, v) + \nu(\nabla u^\bigstar, \nabla v) - (f, v) = 0$$

   including boundary conditions for the velocity.

2. Compute new pressure $p^{n+1}$ by

$$(\nabla p^{n+1}, \nabla q) + \frac{1}{\Delta t}(\nabla \cdot u^\bigstar, q) = 0$$

   including boundary conditions for the pressure.

3. Compute corrected velocity by

$$(u^{n+1} - u^\bigstar, v) + \Delta t(\nabla p^{n+1}, v) = 0$$

   including boundary conditions for the velocity.

# Useful FEniCS tools (I)

Note grad vs. $\nabla$:

*Python code*

```
dot(grad(u), u)
dot(u, nabla_grad(u))
```

Solving linear systems:

*Python code*

```
solve(A, x, b)
solve(A, x, b, "gmres", "ilu")
solve(A, x, b, "cg", "amg")
```

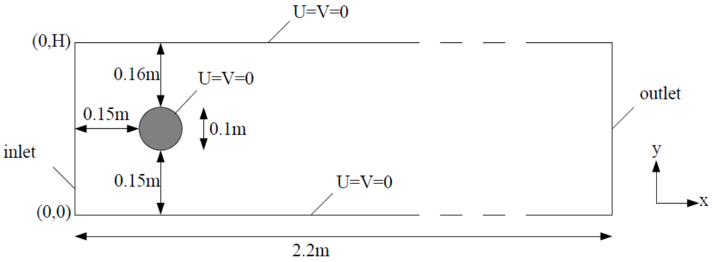
Defining $a$ and $L$ based on residual formulation:

*Python code*

```
F1 = ( (1/k)*inner(u - u0,v) + inner(grad(u0)*u0,v)
+ nu*inner(grad(u),grad(v)) - inner(f,v) )*dx
a1 = lhs(F1)
L1 = rhs(F1)
```

# The FEniCS challenge!

Implement a famous benchmark simulating a laminar flow around a cylinder. The geometry is described by



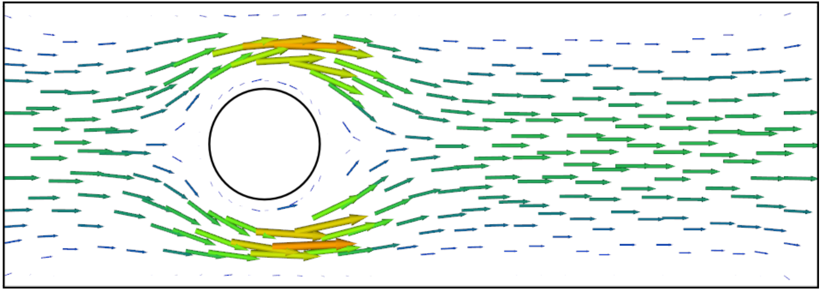

Set the kinematic viscosity $\nu = 0.001\,\mathrm{m}^2/s$ and $\rho = 1.0\,\mathrm{kg/m}^3$. A "do-nothing" boundary condition is assumed at the outlet. Defining $U_m = 1.5\,\mathrm{m/s}$, the time-dependent inflow condition is given by

$$U = 4U_m y(H-y)\sin(\pi t/8)/H^2, \qquad V = 0.$$

Schäfer/Turek, *Benchmark Computations of Laminar Flow Around a Cylinder* (1996)

# The FEniCS challenge!

The inflow boundary lies at $x = -0.2$ and the outflow boundary at $x = 2.0$. Compute the flow on the time interval $[0, 8]$ with time-step $dt = 0.001$. Test your implementation first for a larger time-step $dt = 0.01$ and the same channel problem but with the cylinder removed. If everything goes fine you should get something like



Happy coding!

Schäfer/Turek, *Benchmark Computations of Laminar Flow Around a Cylinder* (1996)