14 – NAVIER-STOKES EQUATIONS

These equations describes the motion of fluids. We work with **steady** NS equations for **incompressible** flows.
(the property of the fluid in a fixed point do not change in the time; the density of the fluid is constant in every point at in every time)

The problem in strong formulation is: (non linear term)

- find $(u,p) \in \mathcal{U} \times Q$ s.t.

$$\begin{cases} -\mu \Delta u + (u\cdot\nabla)u + \nabla p = f & \text{in } \Omega \subset \mathbb{R}^d \\ \operatorname{div}(u) = 0 & \text{in } \Omega \subset \mathbb{R}^d \\ u = g & \text{in } \Gamma_D \subset \partial\Omega \\ -p\vec{n} + \mu(\nabla u)\vec{n} = h & \text{in } \Gamma_N \subset \partial\Omega \end{cases} \qquad \begin{cases} \Gamma_D \cap \Gamma_N = \phi \\ \Gamma_D \cup \Gamma_N = \partial\Omega \end{cases}$$

(u: velocity $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$, p: pressure)

with d is the dimension of the problem
f is the forcing term $f \in [L^2(\Omega)]^d$
$\mu \in P$ is the parameter
g, h have proper regularity ⟶ for semplicity, $g = h = 0$ we can study the homogeneous case

We can pass to **weak formulation**

- find $(u,p) \in \tilde{\mathcal{U}} \times \tilde{Q}$ s.t.

$$\begin{cases} \mu\, a(u,v) + c(u,w,v) + b(u,p) = f(v) & \forall v \in \tilde{\mathcal{U}} \\ b(u,q) = 0 & \forall q \in \tilde{Q} \end{cases}$$

(non linear part)

with $\tilde{\mathcal{U}} := [H^1_{\Gamma_0}(\Omega)]^d$
$\tilde{Q} := L^2(\Omega)$ if $\Gamma_N = \phi$, $\tilde{Q} := L^2_0(\Omega)$ to fix an added condition to have unique solution $\int_\Omega p\,dx = 0$

$a(u,v) := +\int_\Omega \nabla u : \nabla v \, dx$ ⟶ Laplacian wrt x of u + Laplacian wrt y of u, $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$

$b(u,p) := -\int_\Omega p \operatorname{div}(v)\, dx$

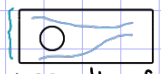
$c(u,w,v) := \int (u\cdot\nabla) w \cdot v \, dx$ ⟶ Scalar product between u & the gradient operator that produces two equations

Equation 1 $\int_\Omega \left(u_1 \frac{\partial}{\partial x} w_1 v_1 \right) dx + \int_\Omega \left(u_2 \frac{\partial}{\partial y} w_1 v_1 \right) dx$

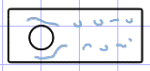
Equation 2 $\int_\Omega \left(u_1 \frac{\partial}{\partial x} w_2 v_2 \right) dx + \int_\Omega \left(u_2 \frac{\partial}{\partial y} w_2 v_2 \right) dx$

14.1 Flow Regime

We have two different kind of flow regime



- **Laminar**: the flows goes around the obstacle and nothing happen
- **Turbulent**: the flows creates a lot of vortices after obstacle



It is determined by the **Reynolds Number** $Re := \frac{LU}{\mu}$ where

- $\mu :=$ Kinematic viscosity
- $L :=$ characteristic lenght of Ω
- $U :=$ characteristic velocity of Ω ⟶ related to boundary condition

How can I solve this problem numerically?

14.2 General formulation of nonlinear problems

The problem becomes

find $u \in \mathcal{U}$ s.t. $g(u(\mu), v; \mu) = 0 \quad \forall v \in \mathcal{U}$

At the FOM level I introduce $\mathcal{U}_\delta \subset \mathcal{U}$ and the problem now is

find $u_\delta \in \mathcal{U}_\delta$ s.t. $g(u_\delta(\mu), v_\delta; \mu) = 0 \quad \forall v_\delta \in \mathcal{U}_\delta$

The problem is well posed if the Frechet derivative of g at $u_\delta(\mu)$ is $dg(u_\delta(\mu))$ (applied to 2 variable) — continuous, inf-sup stable

We solve the problem with Newton Method: given $u^0_\delta(\mu) \in \mathcal{U}_\delta$ we iterate over k

$$dg\left[u^k_\delta(\mu)\right](\delta^k u_\delta, v_\delta; \mu) = -g(u^k_\delta(\mu), v_\delta; \mu)$$

Newton step — Jacobian Matrix — Unkown — Residual Vector

$$u^{k+1}_\delta(\mu) = u^k_\delta(\mu) + \delta^k_{u_\delta}$$

More precisely, for NSE:

find $(\delta u^k, \delta p^k) \in \tilde{u}_\delta \times \tilde{Q}_\delta$ s.t.

$$\begin{cases} (1)\ a(\delta u^k_\delta(\mu), u_\delta;\mu) + c(\underbrace{u^k_\delta}_{\text{unknown}}, \delta u^k_\delta, v_\delta) + c(\delta u^k_\delta, \underbrace{u^k_\delta}_{\text{unknown}}, v_\delta) + b(v_\delta, \delta p^k_\delta) = f(u_\delta) - a(u^k_\delta, v_\delta) - c(u^k_\delta, u^k_\delta, v_\delta) - b(v_\delta, p^k_\delta) \\ (2)\ b(\delta u^k_\delta, q_\delta) = -b(u^k_\delta, q_\delta) \end{cases}$$

with

$$(u^{k+1}_\delta, p^{k+1}_\delta) = (u^k_\delta, p^k_\delta) + (\delta u^k_\delta, \delta p^k_\delta) \longrightarrow X^{k+1} = X^k + \delta^k X \qquad X = (u_\delta, p_\delta)$$

Algebraically

$J_\delta(X^k;\mu)\,\delta^k X = -G_\delta(X^k_\delta;\mu)$, where the jacobian matrix for NSE is $J_f = \begin{pmatrix} A + C_1(u^k_\delta) + C_2(u^k_\delta) & B^T \\ B & 0 \end{pmatrix}$

where $A :=$ stiffness matrix

$B_{ij} := -\int_\Omega \psi_i \operatorname{div}(\varphi_j)\,dx \longrightarrow \{\psi_i\}_{i=1}^{N^p_\delta}$ span $\tilde{Q}$ & $\{\varphi_j\}_{j=1}^{N^u_\delta}$ span $\tilde{u}$ different from Vianni's notes

$C_1(u^k_\delta)_{ij} = c(u^k_\delta, \varphi_j, \varphi_i)$

$C_2(u^k_\delta)_{ij} = c(\varphi_j, u^k_\delta, \varphi_i)$

What about the ROM level? We have to create the basis, i.e. with POD and compute the supremizer

To compute the basis we use POD and

1. We proceed standard for pressure to obtain $V_p \in \mathbb{R}^{N^p_\delta, N}$
2. We enrich the velocity space with supremizer obtaining $V_u \in \mathbb{R}^{N^u_\delta, 2N}$ $V_u = [V_{\text{standard } u} \mid V_S]$ a global stabilized basis $V = \begin{bmatrix} V_u & 0 \\ 0 & V_p \end{bmatrix} \in \mathbb{R}^{N^u_\delta + N^p_\delta, 3N}$

We want to apply a Reduced Newton

Algo

For every k

$J_N(X^k_N;\mu)\,\delta X_N = -G_N(X^k_N;\mu)$, where $J_N(X_N;\mu) = \begin{pmatrix} A_N + C_{1N}(u^k_N) + C_{2N}(u^k_N) & B_N^T \\ B_N & 0 \end{pmatrix}$

$A_N = V_u^T A V_u$

$B_N = V_p^T B V_u$

$C_{1N} = V_u^T C_1(V_u^T u^k_N;\mu) V_u$ $\longrightarrow$ depends on FOM level

$C_{2N} = V_u^T C_2(V_u^T u^k_N;\mu) V_u$

↓

PROBLEM! No separation of the variable, at each step at the ROM level you have to compute FOM level

How to go from ROM to FOM during the iteration?

Considering k-th iteration

- I want to solve $J_N(X^k_N;\mu)\,\delta X_N = -G_N(X^k_N;\mu)$
- Namely at each iteration of the reduced solver $V^T J_\delta(V X^k_N;\mu) V\,\delta X^k_N = -V^T G_\delta(V X^k_N;\mu)$

1. Compute the FOM representation of X^k_N
2. Assemble $J_\delta(V X^k_N;\mu)$ and $G_\delta(V X^k_N;\mu)$
3. Project jacobian and residual vector
4. Solve reduced system obtaining X^{k+1}_N

But, all these steps are so consuming! To avoid this you can apply some ML strategy that we already study