Week 1

Computational

- Numerical methods
- Algorithms
- Programming
- Computer architecture
- Parallel computing

Fluid Dynamics

- Governing equations
- Turbulence models: LES, RANS *etc.*
- Forces and interface models
- Multiphysics coupling
- Interpreting flow solution

Fluid dynamics

$$\nabla \cdot \boldsymbol{u} = 0$$

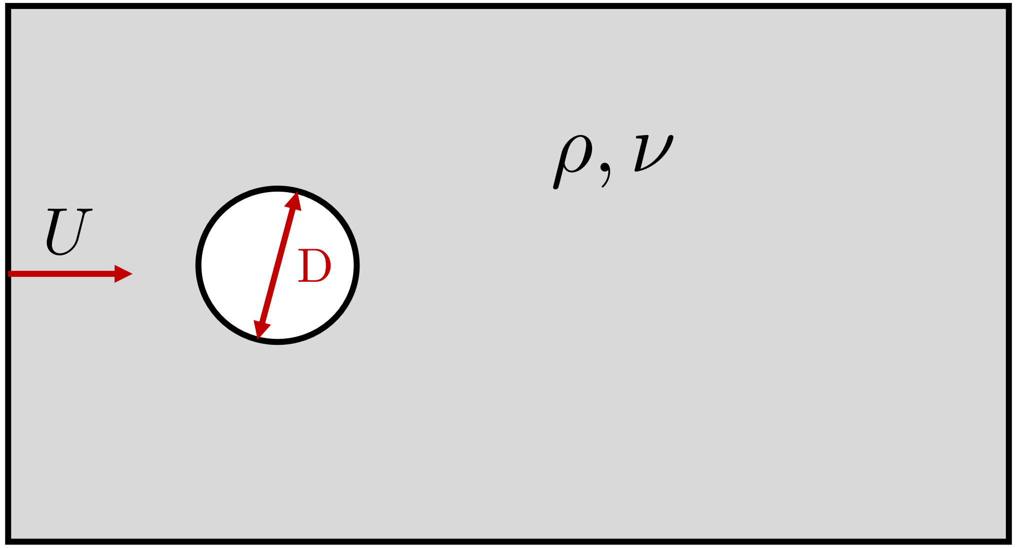
$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla \left(\frac{p}{\rho}\right) + \nu \nabla^2 \boldsymbol{u}$$

$$x' = \frac{x}{D} \qquad t' = \frac{tU}{D} \qquad u' = \frac{u}{U}$$

$$P = \frac{p}{\rho U^2} \qquad Re = \frac{UD}{\nu}$$

$$\nabla \cdot \boldsymbol{u} = 0$$

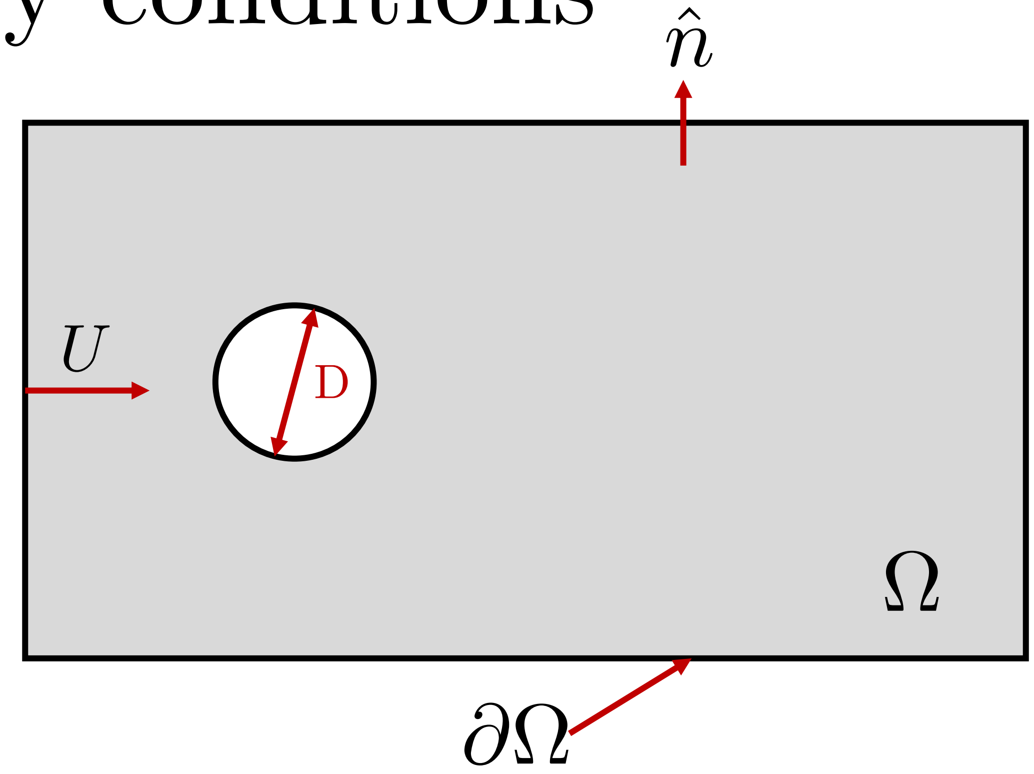
$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla \left(\frac{p}{\rho}\right) + \frac{1}{Re} \nabla^2 \boldsymbol{u}$$



$$[U, D, \nu] \rightarrow \{\boldsymbol{u}, p/p\}$$

$$[Re] \rightarrow \{\boldsymbol{u}', P\}$$

Initial and boundary conditions



Solution: $\boldsymbol{u}(\boldsymbol{x}, t), \boldsymbol{P}(\boldsymbol{x}, t)$

Initial condition: $\boldsymbol{u}(\boldsymbol{x}, 0), \boldsymbol{P}(\boldsymbol{x}, 0)$

Boundary conditions:

$\boldsymbol{u}(\boldsymbol{x} \in \partial\Omega, t), \boldsymbol{P}(\boldsymbol{x} \in \partial\Omega, t)$

Dirichlet BC:

$\boldsymbol{u} = \alpha, P = \beta$

Neumann BC:

$\frac{\partial \boldsymbol{u}}{\partial n} = \alpha, \frac{\partial P}{\partial n} = \beta$

$$\nabla \cdot \boldsymbol{u} = 0$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla\left(\frac{p}{\rho}\right) + \frac{1}{Re}\nabla^2 \boldsymbol{u}$$

Initial and boundary conditions

Inlet: fixed-velocity condition

$$\boldsymbol{u} = U, \frac{\partial P}{\partial n} = 0$$

Outlet: zero-shear flow at far-field

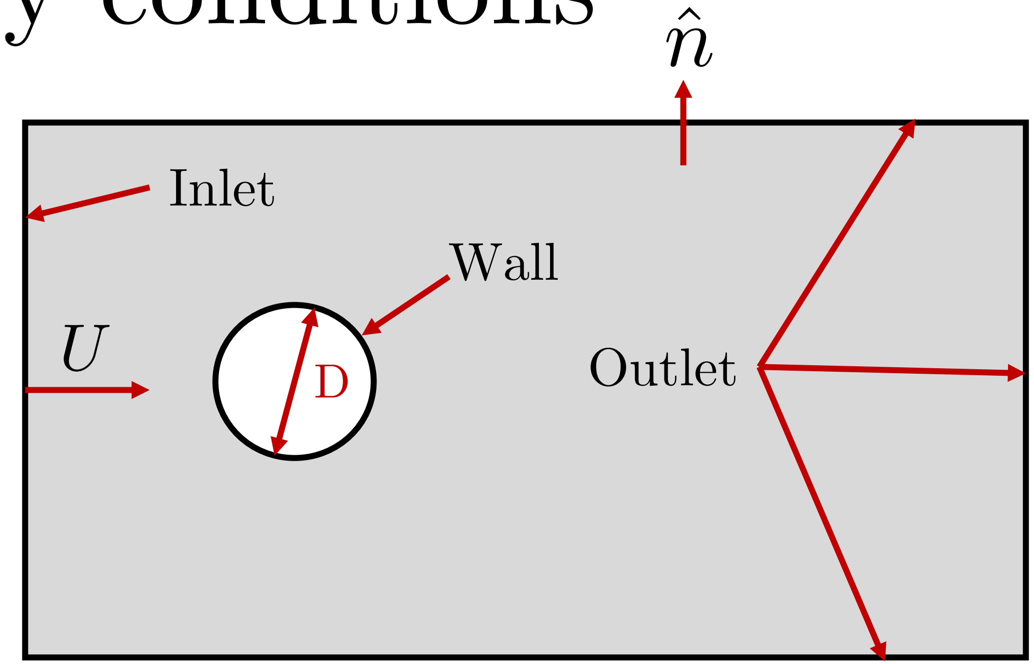
$$\mu \frac{\partial u}{\partial n} = 0, \quad \nabla P = \nabla^2 P = 0$$

$$\implies \frac{\partial u}{\partial n} = 0, \quad P = P_{\text{gauge}} = 0$$

Wall: No-slip condition &

thin boundary layer approximation

$$\boldsymbol{u} = 0, \frac{\partial P}{\partial n} = 0$$



$$\nabla \cdot \boldsymbol{u} = 0$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla \left(\frac{p}{\rho}\right) + \frac{1}{Re}\nabla^2 \boldsymbol{u}$$

Numerical discretization

Let $u(t)$ satisfies ODE: $\dfrac{du}{dt} = f(u), u(t=0) = u0$

$$u(t \pm \Delta t) \;=\; u(t) \;\pm \frac{du}{dt}\Delta t \;+ \frac{d^2u}{dt^2}\frac{\Delta t^2}{2!} \;\pm \frac{d^3u}{dt^3}\frac{\Delta t^3}{3!} \;+ \ldots$$

Taylor series expansion

$$\frac{du}{dt} = \frac{u(t+\Delta t) \;\; - u(t)}{\Delta t} - \underbrace{\frac{d^2u}{dt^2}\frac{\Delta t}{2!}}_{\text{T.E.}=\mathcal{O}[\Delta t]} - \cdots$$

Forward Euler – diffusive but stable

$$\frac{du}{dt} = \frac{u(t) \;\; - u(t-\Delta t)}{\Delta t} + \underbrace{\frac{d^2u}{dt^2}\frac{\Delta t}{2!}}_{\text{T.E.}=\mathcal{O}[\Delta t]} + \cdots$$

Backward Euler – Anti-diffusive, unstable

$$\frac{du}{dt} = \frac{u(t+\Delta t) \;\; - u(t-\Delta t)}{2\Delta t} - \underbrace{\frac{d^3u}{dt^3}\frac{\Delta t^2}{3!}}_{\text{T.E.}=\mathcal{O}[\Delta t^2]} - \cdots$$

Central – Dispersion error

Numerical discretization

Let $u(t)$ satisfies ODE: $\dfrac{du}{dt} = f(u), u(t=0) = u0$

$$u(t \pm \Delta t) = u(t) \pm \frac{du}{dt}\Delta t + \frac{d^2u}{dt^2}\frac{\Delta t^2}{2!} \pm \frac{d^3u}{dt^3}\frac{\Delta t^3}{3!} + \dots$$

Taylor series expansion

$$\frac{du}{dt} = \frac{u(t+\Delta t) - u(t)}{\Delta t} - \underbrace{\frac{d^2u}{dt^2}\frac{\Delta t}{2!}}_{\text{T.E.}=\mathcal{O}[\Delta t]} - \dots$$

Forward Euler – diffusive but stable

$$\frac{du}{dt} = \frac{u(t+\Delta t) - u(t-\Delta t)}{2\Delta t} - \underbrace{\frac{d^3u}{dt^3}\frac{\Delta t^2}{3!}}_{\text{T.E.}=\mathcal{O}[\Delta t^2]} - \dots$$

Central – Dispersion error

$$\frac{du}{dt} = \frac{3u(t+\Delta t) - 4u(t) + u(t-\Delta t)}{2\Delta t} + \underbrace{2\frac{d^3u}{dt^3}\frac{\Delta t^2}{3!}}_{\text{T.E.}=\mathcal{O}[\Delta t^2]} + \dots$$

2$^{\text{nd}}$ order backward, dispersion error

Integration

Let $u(t)$ satisfies ODE: $\dfrac{du}{dt} = f(u), u(t=0) = u0$

$$\frac{du}{dt} = \frac{u(t+\Delta t) \;\; - u(t)}{\Delta t} - \underbrace{\frac{d^2u}{dt^2}\frac{\Delta t}{2!}}_{\text{T.E.}=\mathcal{O}[\Delta t]} - \dots$$

Forward Euler – diffusive but stable

$$f(u(t+\Delta t)) = f(u(t)) \; + \frac{df}{du}(u(t+\Delta t) - u(t)) \; + \dots$$

Taylor series expansion

$$\frac{u^{n+1} - u^n}{\Delta t} = f(u^{n+1})$$

Implicit discretization, stable, possible for only linear f

$$\frac{u^{n+1} - u^n}{\Delta t} = f(u^n)$$

Explicit discretization, conditional stability

Example

$$\frac{\partial u}{\partial t} + c\,\frac{\partial u}{\partial x} = 0, \quad u(x,\ t=0) = u_0\ e^{\alpha(x-x_0)^2}$$

Implicit discretization, Forward in time, central in space (FTCS)

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\,\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} \;=\; 0$$

$$\Longrightarrow\ -\frac{N_c}{2}\,u_{i-1}^{n+1} \;+\; u_i^{n+1} \;+\; \frac{N_c}{2}\,u_{i+1}^{n+1} \;=\; u_i^n, \quad N_c = \frac{c\Delta t}{\Delta x}$$

Solution always stable

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}.$$

System of linear equations.

Solve using

- Tridiagonal matrix algorithm
- Iterative solution

Example

$$\frac{\partial u}{\partial t} + c\,\frac{\partial u}{\partial x} = 0, \quad u(x,\ t=0) = u_0\ e^{\alpha(x-x_0)^2}$$

Explicit discretization, Forward in time, central in space (FTCS)

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\,\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \; = \; 0$$

$$\Longrightarrow u_i^{n+1} \; = \; u_i^n \; - \; \frac{N_c}{2}\,(u_{i+1}^n \; - \; u_{i-1}^n), \quad N_c = \frac{c\Delta t}{\Delta x}$$

No matrix solution required.

Solution conditionally stable for:

$$N_c < 1$$

Homework: Implement this solution algorithm with different discretization methods in any programming language and compare solutions.