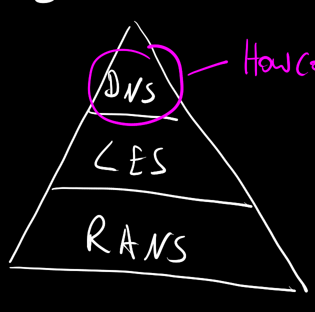
# Why is DNS of turbulence so expensive?



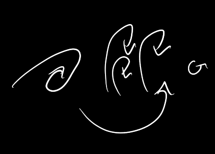

How costly?

Cost

computational cost

resolved physics

Spoiler: $\mathcal{O}(Re^3)$

Turbulence: 

most engineering applications are turbulent

- chaotic
- lives in multiple scales
- small scales affect larger scales

## When is a flow turbulent?

incompressible dimensionless Navier-Stokes equations

non-linear contribution

$$\frac{\partial \underline{u}}{\partial t} + (\underline{u}\cdot\underline{\nabla})\,\underline{u} = -\nabla p + \frac{1}{Re}\nabla^2 \underline{u}$$

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

the most important dimensionless number in CFD

→ Reynolds Number

$$Re = \frac{\text{convective transport}}{\text{diffusive transport}}$$

convection is associated with the non-linearity (→ turbulence)

⇒ large Re: convection dominated flow

small Re: diffusion dominated flow

[ $Re \to \infty$ : Euler Equations

$Re \to 0$ : Stokes Equations ]

pipe flow: $Re > 2300$ transition to turbulence

$Re > 4000$ fully turbulent flow

## How to compute the Re?

$$Re = \frac{\text{characteristic length} \cdot \text{characteristic velocity}}{\text{viscosity}}$$

i.e. we have a higher Re if:

1. the geometry is large
2. the fluid moves fast
3. the fluid is rather inviscid

typical engineering CFD fluids are air & water

$\nu_{air} = 1.46\cdot 10^{-5}\,\frac{m^2}{s}$ $\nu_{water} = 1.14\cdot 10^{-6}\,\frac{m^2}{s}$

even if our geometry is only 1m and the fluid flows at $1\frac{m}{s}$, we already $Re > 10^5$

typical airplanes have velocities $> 100\,m/s$

↳ aerospace $Re = 10^8$

## But how does Re affect the scales of turbulence?

→ Kolmogorov scales

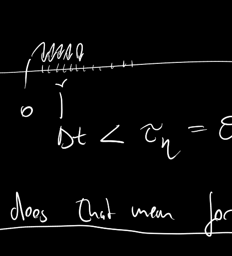
length scale: $\eta = \mathcal{O}(Re^{-3/4})$

time scale: $\tau_\eta = \mathcal{O}(Re^{-1/2})$

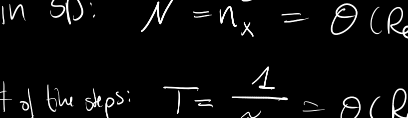
velocity scale: $u_\eta = \mathcal{O}(Re^{-1/4})$

[ see Pope: "Turbulent Flows"

if you use a discretization that is larger than these scales, you do not resolve them



$\Delta x < \eta = \mathcal{O}(Re^{-3/4})$ for $Re^8$ → $\Delta x \approx 10^{-6}$




$\Delta t < \tau_\eta = \mathcal{O}(Re^{-1/2})$ for $Re^8$ → $\Delta t \approx 10^{-4}$

## What does that mean for the computational cost?

dof per axis: $n_x = \frac{1}{\eta} = \mathcal{O}(Re^{3/4})$

in 3D: $N = n_x^3 = \mathcal{O}(Re^{9/4})$

# of time steps: $T = \frac{1}{\tau_\eta} = \mathcal{O}(Re^{1/2})$

## How high is the cost per time step?

**explicit**

$\mathcal{O}(N)$

are we stable?

CFL: $\frac{\Delta t \cdot u}{\Delta x} < c = \mathcal{O}(1)$

$$\leadsto \Delta t < \frac{\Delta x}{u}\cdot\mathcal{O}(1) = \frac{\mathcal{O}(Re^{-3/4})}{\mathcal{O}(Re^{-1/4})}$$

$$= \mathcal{O}(Re^{-3/4+1/4})$$

$$= \mathcal{O}(Re^{-1/2})$$

That's exactly our time scale

**implicit** (solving a linear system or non-linear)

$\mathcal{O}(N)$ multigrid

$\mathcal{O}(N\log N)$ FFT

$\mathcal{O}(K\cdot N)$ Krylov solver

$\mathcal{O}(N^3)$ LU / QR

in other words: DNS allows for explicit time-stepping methods because the step length required to be numerically stable is also the step length required to capture all scales

## What is the total cost?

total = cost per time step · # of time steps

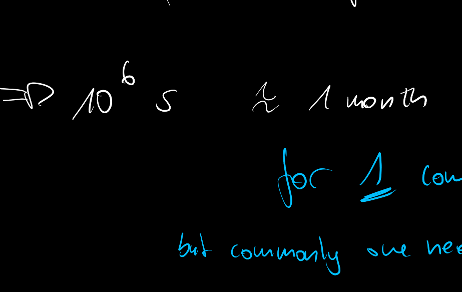
$= \mathcal{O}(N) \cdot \mathcal{O}(Re^{1/2})$

$= \mathcal{O}(Re^{9/4}) \cdot \mathcal{O}(Re^{1/2})$ DNS

$= \mathcal{O}(Re^{11/4}) \approx \mathcal{O}(Re^{12/4}) = \mathcal{O}(Re^3)$

aerospace example: $Re = 10^8$ → $\mathcal{O}(10^{24})$ floating point operations

Top500



| Rank | System | Cores | Rmax (TFlop/s) | Rpeak (TFlop/s) | Power (kW) |
|---|---|---|---|---|---|
| 1 | Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan | 7,630,848 | 442,010.0 | 537,212.0 | 29,899 |
| 2 | Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States | 2,414,592 | 148,600.0 | 200,794.9 | 10,096 |
| 3 | Sierra - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States | 1,572,480 | 94,640.0 | 125,712.0 | 7,438 |

$10^{15}$

$500\cdot 10^{15}\,\frac{flop}{s}$

let's assume an exascale computer $10^{18}\,\frac{flop}{s}$

⇒ $10^6\,s$ ≈ 1 month

for 1 computation

but commonly one needs

> 1000 computations

for design optimization & exploration

[ Moore's law: doubled power/strength every 18 months ]

astrophysics: $Re = 10^{12}$