Numerical approaches to turbulence modelling + simulation

Direct Numerical Simulation (DNS)

3D Navier Stokes fully describes turbulence, therefore we 'just' need to solve them in 3D, and time dependent, and resolve all the scales including the dissipation range – Taylor microscale, down to near-Kolmogorov.

However, this is computationally expensive (and in Engineering we develop other methods based around filtering to split the flow into resolved and sub-grid components, → LES and RANS, see later).

DNS concept: fully resolved Navier Stokes, all turbulent information solved for ⇒ complete knowledge of the flow (lots of data, computationally expensive)

We solve in the time domain; Method is:

1. initialise flow at t=0 with (a) mean flow to match roughly what is expected, (b) fluctuating velocity is chosen to facilitate rapid convergence – we usually seed the flow with low level random noise to promote rapid development to turbulence.

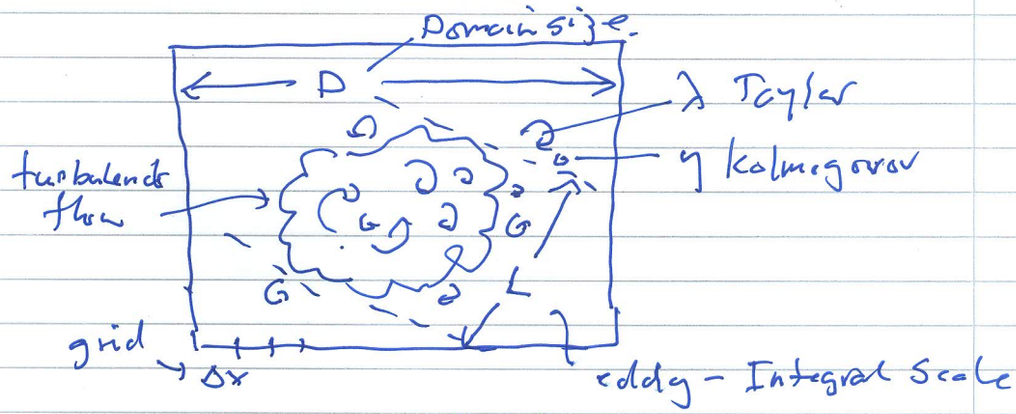
2. step in time, conditioning – this develops turbulence (so we end up with statistically steady turbulence) and allows memory of initial conditions to fade away.

3. Collect statistics sample time series, calculate Reynolds stress, etc. This is just like in a wind tunnel but with data collection at every point and every time step.

4. Data Analysis – process the data, calculate turbulence quantities, plot graphs, explore the data to understand the flow.

DNS costs - computational complexity

Suppose we are solving turbulent flow in some domain of size D and with large eddy scales of size L. We need $D \gg L$ to avoid influence of domain boundaries (don't want to over-constrain the turbulence). We also need to resolve the dissipation range, so need at least λ and extend down to near (about 10η) Kolmogorov η.



We need

$$D \gg L \gg \lambda \gg \eta$$

box — D; large eddy — L; Taylor micro- — λ; Kolmogorov scale — η

outer scales; inner scales.

Suppose we use N grid points along a dimension of order D, then $N \sim D/\Delta x$, and we ideally need $\Delta x \sim \eta$ (or near).

$$D/L \sim 10 \text{ (say)}$$

$$\lambda/L \sim Re^{-1/2}$$

$$\eta/L \sim Re^{-3/4}$$

where $Re \sim \frac{U_0 L}{\nu}$. Number of grid points scales as $N \sim O(D/\eta) \sim Re^{+3/4}$ along an edge.

Computational cost scales as:

time to process the grid N^3
time steps @ constant CFL N
$\}$ $\sim$ CPU $\sim N^4$. cost

Memory $\sim N^3 \sim Re^{9/4}$

CPU $\sim N^4 \sim Re^3$

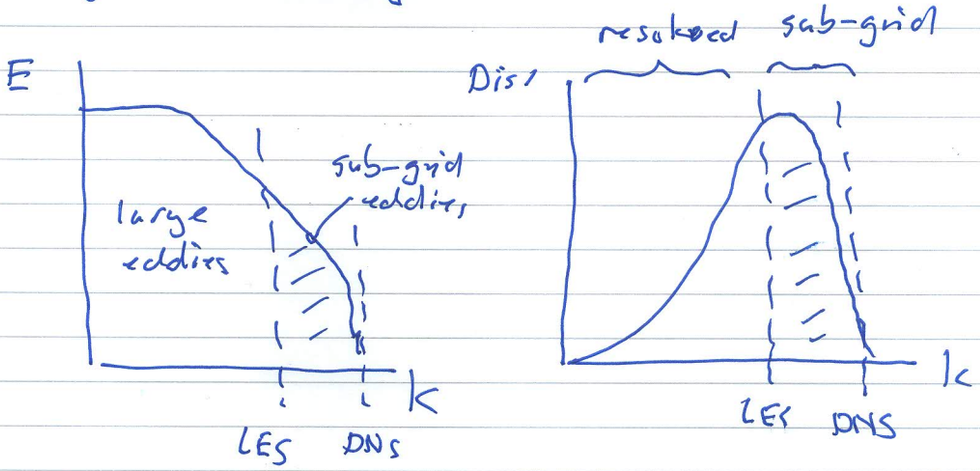
Each decade of Re increase $\Rightarrow \times 10^3$ CPU cost.

BUT, computers get faster each year.

LES (Large eddy simulation)

Aim is to resolve only the large eddies on the grid and to model the small (sub-grid) ones.

[concept: if DNS costs scale rapidly with resolution, then methods to reduce resolution will deliver large cost savings.]



Idea is to use a low-pass filter (at grid scale, or near) to decompose the flow into (a) resolvable large eddies which we retain on the grid, and (b) sub-grid eddies which we model.

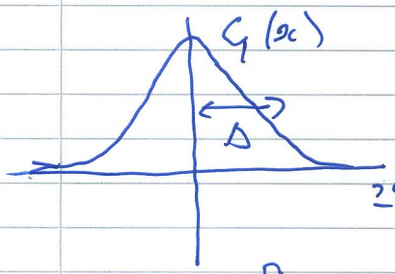
⇒ Modelling process is easier because
- SG energy is relatively small (modelling errors have less impact)
- SG scales are more locally defined, so gradient diffusion approx should be better.
- SG eddies are more isotropic, so simpler models should scale well.

How do we implement the filter?
The simplest is to use the natural tendency of Navier-Stokes to filter the scales, ⇒ note that the viscosity places a natural cut-off on the smallness of scales at near-laminar.
⇒ We can implement a filter by using an eddy viscosity ν_{SG}, and choosing it so that the natural extent of small scales is just at the limit of the computational grid. - This is the basis of most SG-models.

Mathematically, filtering is defined by - [$\hat{(\,)} \equiv$ filter]

$$\hat{u}(\underline{x},t) = \int_{\text{Volume}} G(\underline{x}-\underline{x}')\, u(\underline{x}',t)\, d^3\underline{x}'$$

i.e. a convolution of the flow with a filter kernel of width Δ related to the grid scale (Δx, etc). The precise shape of the filter is not too important provided it decays to zero fast and approx width is $\sim \Delta$.



unit area $\int_{\text{Vol.}} G(x)\, d^3x = 1$.

We apply this to the NS equations, & rewrite in filtered variables.

$$\frac{\partial}{\partial t}\hat{u}_i + \frac{\partial}{\partial x_j}\left(\underbrace{\hat{u}_i\hat{u}_j - \hat{u}_i\hat{u}_j}_{-T_{ij}} + \frac{2}{3}\delta_{ij}E_{SG} + \hat{u}_i\hat{u}_j \right)$$

combine with T_{ij} (see below)

combine with P

$$= -\frac{\partial}{\partial x_i}(\hat{p}/\rho) + \frac{\partial}{\partial x_j}\left(\frac{2}{3}\delta_{ij}E_{SG}\right)$$

we've done two tricks here: (a) added & subtracted $(\hat{u}_i\hat{u}_j)_{,j}$ so that we have a conventional convective term in filtered velocity, and (b) added $(\frac{2}{3}\delta_{ij}E_{SG})_{,j}$ to each side in anticipation of resolving a modelling issue below -, here E_{SG} = sub grid energy.

Now we can define the subgrid stress τ_{ij}

$$\tau_{ij} \equiv -(\widehat{u_i u_j} - \hat{u}_i\hat{u}_j)$$

and the SG energy (~~sim~~ SG energy, similar to

$$E_{SG} \equiv -\frac{1}{2}\tau_{kk} \geq 0$$

trace of Reynolds stress for TICE) Now we can use a gradient diffusion (eddy visc) model for τ_{ij}

$$\left(\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij}\right) = 2\nu_{SG}\hat{S}_{ij}$$

where $\hat{S}_{ij} = \frac{1}{2}\left(\frac{\partial\hat{u}_i}{\partial x_j} + \frac{\partial\hat{u}_j}{\partial x_i}\right)$. ν_{SG} = sub-grid eddy visc.

Here we've modelled the deviatoric part of τ_{ij} so that when we take the trace of each side it remains consistent, since trace of S_{ij} must be zero by continuity.

The $\frac{1}{3}\tau_{kk}S_{ij}$ term on the RHS is combined with the pressure,

'modified pressure' defined as

$$P \equiv (\hat{P}/\rho) - \frac{1}{3}\tau_{kk}$$