Overview

1. Recap of N-S equations
2. Approximating the equations of motion
3. Pros and cons of each method
4. Scale separation
Recap

• The Navier-Stokes equations describe the motion of fluids, through the conservation of mass, momentum, and energy.
• The equations are nonlinear, which complicates our ability to analyze and simulate fluid flows.
• Why? The nonlinearity creates a continuous spectrum of different flow features.
Recap of N-S equations

- This spectrum contains very large integral scales and very small dissipation scales.
- The simultaneous representation of both large and small scales makes for a very large computational problem.
- Current computational resources limit the amount of small features that can be accurately simulated.
Recap

• The complexity of a flow can be reduced to alleviate this computation bottleneck.

• This technique is aimed at capturing the primary features of a flow with sufficient detail and accepting that the full turbulent solution may not be obtained perfectly.

• This sets the stage for LES as a tool to solve for the “reduced” flow.

• Before diving into LES, we will go over the description of the DNS, LES, and RANS techniques.
Approximating the equations of motion

- Numerical studies require that the equations of motion for a (compressible, incompressible, Boussinesq) fluid must be approximated on a computational grid using discrete physical points or basis functions.
- Three basic methodologies are prevalent in turbulence application and research:
  - Direct Numerical Simulation (DNS)
  - Large-Eddy Simulation (LES)
  - Reynolds-Averaged Navier-Stokes (RANS)
Approximating the equations of motion

Direct Numerical Simulation

- The DNS approach focuses on finding a numerically-accurate solution to the N-S equations (i.e., resolve all eddies).
- As we saw last class, it is an expensive operation.
Large-Eddy Simulation

- The LES approach emphasizes capturing those primary flow features that are larger than a prescribed filter width ($\Delta$) \textit{(i.e., resolve larger-eddies and model smaller “universal” ones)}.
- Since $\Delta$ is prescribed, one has control over the required resolution and computational effort.
- The LES approach introduces the closure problem and reduces the information of the resolved flow.
- Has primarily trended toward engineering applications, but its use in atmospheric science is increasing.
Approximating the equations of motion

Reynolds-Averaged Navier-Stokes

- The RANS approach focuses on a statistical description of the basic fluctuation-correlations (i.e., model ensemble statistics).
- Can be used to study flows with realistic complexity.
Approximating the equations of motion
Direct Numerical Simulation

- Pros
  - Does not require the use of a turbulence model
  - Accuracy is only limited by computational capabilities since errors are generally only due to sensitivity to perturbations or accumulated round-off errors.
  - All aspects of the flow in time and space are available, which is not possible for experiments (i.e., 3D velocity and scalar fields).
Pros and cons of each method

Direct Numerical Simulation

- Cons
  - Restricted to low-Re flows with relatively simple geometries.
  - Very high memory and computational time costs.
  - Typically the “largest possible” number of grid points is used without proper convergence evaluation (i.e., does not allow for systematic variation of numerical and physical parameters)
Pros and cons of each method

Large-Eddy Simulation

• Pros
  • Only small scales require modeling. Since the small scales are likely insensitive to specifics of the flow, these models can be rather simple and “universal”.
  • Much cheaper computational cost than DNS.
  • Unsteady predictions of the flow are made, which implies that information about extreme events are available over some period of time.
  • Properly designed LES should allow for \( Re \rightarrow \infty \).
  • In principle, we can gain as much accuracy as desired by refining our numerical grid.
Pros and cons of each method

Large-Eddy Simulation

• Cons
  • Basic assumption (small scales are universal) requires independence of small (unresolved) scales from boundary conditions (especially important for flow geometry). This is a problem for boundary layers, where proximity to the wall defines some of the smallest scales of the flow – which necessitates explicit modeling of the region.
  • Still very costly for many practical engineering applications.
  • Filtering and turbulence theory of small scales still needs development for complex geometry and highly anisotropic flows.
Reynolds-Averaged Navier-Stokes

- **Pros**
  - Low computational cost (can obtain mean statistics in a short time).
  - Can be used for highly complex flow geometries.
  - When combined with empirical information, can be highly useful for engineering applications and to parameterize near-wall behavior.
Pros and cons of each method

Reynolds-Averaged Navier-Stokes

• **Cons**
  
  • Only steady flow phenomena can be explored with full advantage of computational reduction.
  
  • Models are not universal since dynamic consequences of all scales must be represented. Usually a pragmatic “tuning” of model parameters is required for specific applications.
  
  • Most accurate turbulence models give rise to highly complex equations sets, which can lead to numerical formulation and convergence issues.
• DNS delivers the most accurate data (in general).
• DNS can be used to validate and analyze aspects of LES, such as numerical methods and subgrid model.
• LES provides a more complete picture of turbulent flow than RANS.
• RANS can be validated against LES data (where LES is used to obtain statistical information about the flow).
Relationship between each method

Resolution

Computational Boundary

Validation

Extrapolation

DNS

LES

LES

RANS

RANS

Complexity (Reynolds number, geometry ... )
So far, we have discussed LES very generically:

- Resolve only the largest energy-containing scales.
- Model the small “universal” scales.

We will now formalize the idea of scale separation mathematically to show how to deal with the equations of motion and derive subgrid models.
• How do we accomplish scale separation?
  • A low-pass filter

• What is meant by low-pass?
  • A low-pass filter passes over signals with a frequency (wavenumber) lower than a certain cutoff frequency (wavenumber) and only smooths signals with a frequency (wavenumber) higher than the cutoff value.

• Our goal for the low-pass filter:
  • Attenuate (smooth) high frequency (high wavenumber/small scale) turbulence features that are smaller than a prescribed characteristic scale $\Delta$ while leaving low-frequency (low wavenumber/large scale) motions unchanged.
Filtering (Sagaut Chapter 2, Pope Chapter 13.2)

- The formal (mathematical) LES filter is a convolution filter defined for a quantity $\phi(\vec{x}, t)$ in physical space as:

$$\tilde{\phi}(\vec{x}, t) = \int_{-\infty}^{\infty} \phi(\vec{x} - \vec{\zeta}, t) G(\vec{\zeta}) d\vec{\zeta}$$

- $G \equiv$ the convolution kernel of the chosen filter.
- $G$ is associated with the characteristic cutoff scale $\Delta$ (also called the filter width).
So, we have the convolution filter

\[ \tilde{\phi}(\vec{x}, t) = \int_{-\infty}^{\infty} \phi(\vec{x} - \vec{\zeta}, t) G(\vec{\zeta}) d\vec{\zeta} \]

Here we will use Pope’s notation for the Fourier transform:

\[ F\{\phi(x)\} = \int_{-\infty}^{\infty} e^{-ikx} \phi(x) dx \]
Convolution

- Take the Fourier transform of $\tilde{\phi}(\vec{x})$ (dropping $t$ for simplicity):

$$F\{\tilde{\phi}(\vec{x})\} = \int_{-\infty}^{\infty} e^{-i\vec{k}\vec{x}} \int_{-\infty}^{\infty} \phi(\vec{x} - \vec{\zeta}, t) G(\vec{\zeta}) d\vec{\zeta} d\vec{x}$$

- We can define a new variable $\vec{r} = \vec{x} - \vec{\zeta}$:

$$F\{\tilde{\phi}(\vec{x})\} = \int_{-\infty}^{\infty} e^{-i\vec{k}(\vec{r} + \vec{\zeta})} \int_{-\infty}^{\infty} \phi(\vec{r}) G(\vec{\zeta}) d\vec{\zeta} d\vec{r}$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i\vec{k}(\vec{r} + \vec{\zeta})} \phi(\vec{r}) G(\vec{\zeta}) d\vec{\zeta} d\vec{r}$$

Note: $d\vec{r} = d\vec{x}$ because $\vec{\zeta} \neq f(\vec{x})$ and we changed the order of integration
Convolution

• We left off with:

\[
F\{\tilde{\phi}(\vec{x})\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i\vec{k}(\vec{r}+\vec{\zeta})} \phi(\vec{r}) G(\vec{\zeta}) \, d\vec{\zeta} \, d\vec{r} 
\]

• Recall that \(e^{a+b} = e^a e^b\):

\[
F\{\tilde{\phi}(\vec{x})\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i\vec{k}\vec{r}} e^{-i\vec{k}\vec{\zeta}} \phi(\vec{r}) G(\vec{\zeta}) \, d\vec{r} \, d\vec{\zeta}
\]

\[
= \int_{-\infty}^{\infty} e^{-i\vec{k}\vec{r}} \phi(\vec{r}) d\vec{r} \int_{-\infty}^{\infty} e^{-i\vec{k}\vec{\zeta}} G(\vec{\zeta}) d\vec{\zeta}
\]

\[
= F\{\phi(\vec{x})\} F\{G(\vec{\zeta})\}
\]

where we changed the order for integration
• We found this convolution relationship:

\[
F\{\tilde{\phi}(x)\} = F\{\phi(x)\}F\{G(\zeta)\}
\]

• Segaut writes this as:

\[
\hat{\phi}(k, \omega) = \phi(k, \omega)\hat{G}(k\omega)
\]

where \((^\wedge)\) denotes a Fourier coefficient.
Convolution

- \( F\{\tilde{\phi}(\vec{x})\} = F\{\phi(\vec{x})\}F\{G(\vec{\zeta})\} \) or \( \hat{\phi}(\vec{k},\omega) = \hat{\phi}(\vec{k},\omega)\hat{G}(\vec{k}\omega) \)
- \( \hat{G} \) is the transfer function associated with the filter kernel \( G \).
- Recall, that a transfer function is the wavespace (Fourier) relationship between the input and output of a linear system.
- A convolution is an integral that expresses the amount of overlap of one function \( G \) as it is shifted over another function \( \phi \) (i.e., it blends one function with another).
Just as $G$ is associated with a filter scale $\Delta$ (filter width), $\hat{G}$ is associated with a cutoff wavenumber $k_c$.

In a similar manner to Reynolds decomposition, we can use the filter function to decompose the velocity field into resolved and unresolved (or subfilter) components:

\[
\phi(\vec{x}, t) = \tilde{\phi}(\vec{x}, t) + \phi'(\vec{x}, t)
\]

- total
- resolved
- subfilter
Fundamental properties of “proper” LES filters

- The filter should not change the value of a constant $a$

\[
\int_{-\infty}^{\infty} G(\vec{x}) d\vec{x} = 1 \Rightarrow \tilde{a} = a
\]

- Linearity

\[
\tilde{\phi} + \tilde{\zeta} = \tilde{\phi} + \tilde{\zeta}
\]

(this is automatically satisfied for a convolution filter)

- Commutation with differentiation

\[
\frac{\partial \tilde{\phi}}{\partial \vec{x}} = \frac{\partial \tilde{\phi}}{\partial \vec{x}}
\]
• In the general case, LES filters that verify these properties are not Reynolds operators.
• Recall for a Reynolds operator (average) defined by \( \langle \rangle \)
  • \( \langle a\phi \rangle = a\langle \phi \rangle \)
  • \( \langle \phi' \rangle = 0 \)
  • \( \langle \phi + \zeta \rangle = \langle \phi \rangle + \langle \zeta \rangle \)
  • \( \langle\langle \phi \rangle \rangle = \langle \phi \rangle \)
  • \( \langle \frac{\partial \phi}{\partial \vec{x}} \rangle = \frac{\partial \langle \phi \rangle}{\partial \vec{x}} \)
For our LES filter, in general using Sagaut’s shorthand
\[ \int_{-\infty}^{\infty} \phi(\vec{r}, t) G(\vec{\zeta}) d\vec{\zeta} = G \ast \phi: \]

- \( \tilde{\phi} = G \ast G \ast \phi = G^2 \ast \phi \neq \tilde{\phi} = G \ast \phi \)
- \( \tilde{\phi}' = G \ast (\phi - G \ast \phi) \neq 0 \)

For an LES filter, a twice filtered variable is not equal to a single filtered variable – unlike it is for a Reynolds average.

Likewise, the filtered subfilter scale component is not equal to zero as it is for a Reynolds average.
Common (or classic) LES filters

- Box or “top-hat” filter (equivalent to a local average):
  \[
  G(r) = \begin{cases} \frac{1}{\Delta} & \text{if } r \leq \frac{\Delta}{2} \\ 0 & \text{otherwise} \end{cases} \\
  \hat{G}(k) = \frac{\sin \left( k \Delta / 2 \right)}{k \Delta / 2}
  \]

- Gaussian filter ($\gamma$ typically 6):
  \[
  G(r) = \left( \frac{\gamma}{\pi \Delta^2} \right)^{\frac{1}{2}} \exp \left( -\frac{\gamma |r|^2}{\Delta^2} \right) \\
  \hat{G}(k) = \exp \left( -\frac{\Delta^2 k^2}{4 \gamma} \right)
  \]

- Spectral or sharp-cutoff filter:
  \[
  G(r) = \frac{\sin(k_c r)}{k_c r} \\
  \hat{G}(k) = \begin{cases} 1 & \text{if } |k| \leq k_c \\ 0 & \text{otherwise} \end{cases}
  \]

recall that $k_c$ is the characteristic wavenumber cutoff.
Only the Gaussian filter is local in both real and wave space.
• Where we apply a filter is important.
• The Fourier transform of a box filter is a wave, and the inverse transform of a spectral cutoff filter is a wave.
• This means we will get different results for these two filters depending on where they are applied.
Filters: local vs non-local

• Thus, we say a box filter is local in physical space and non-local in wavespace.
• Conversely, a cutoff filter is local in wavespace and non-local in physical space.
• When a filter is non-local, think about it in terms of adding “wiggles” everywhere.
• As opposed to the box and cutoff filters, the Gaussian filter is (semi) local in both physical space and wavespace (semi because it differs by constants).
• This is because the Fourier transform of a Gaussian is also a Gaussian.
We can tie the notion of filters and local/non-local behavior to numerical models and resolution.

The notion of “spectral” or “effective” resolution arises because the spectra from LES often does not fall at $\Delta$, but rather at some larger scale that is a multiple of $\Delta$.

For instance, a finite difference scheme (perhaps 2nd-order central difference) is local in physical space, but non-local in wavespace.

This impacts smaller wavenumbers (larger scales).
Spectral resolution

From Gibbs and Fedorovich (2014).
Spectral resolution

![Graph showing w-velocity spectral density vs wavenumber with various markers indicating different scales]
• We defined the convolution of two functions as:

\[ \tilde{\phi}(\vec{x}, t) = \int_{-\infty}^{\infty} \phi(\vec{x} - \vec{\zeta}, t) G(\vec{\zeta}) d\vec{\zeta} \]

• How can we interpret this relation? \( G \), our filter kernel, ‘moves’ along our other function \( \phi \) and smooths it out (provided it is a low-pass filter).
Convolutions example

- One example is using a box filter applied in real space.
- See convolution_example.m (and associated iso_vel128.mat data file) on Canvas or website.
Filtering turbulence (real space, cutoff filter)

Note: here (and throughout the presentation) we are using DNS data from Lu et al. (International Journal of Modern Physics C, 2008).
Filtering turbulence (real space, Gaussian filter)

Note: here (and throughout the presentation) we are using DNS data from Lu et al. (International Journal of Modern Physics C, 2008).
Filtering turbulence (real space, box filter)

Note: here (and throughout the presentation) we are using DNS data from Lu et al. (International Journal of Modern Physics C, 2008).
Filtering turbulence (real space, box filter)

Another example (Fedorovich and Gibbs, submitted)
Note: here (and throughout the presentation) we are using DNS data from Lu et al. (International Journal of Modern Physics C, 2008).
Filtering turbulence (wave space)
Filtering turbulence (wave space)
The LES filter can be used to decompose the velocity field into resolved and subfilter scale (SFS) components:

\[
\phi(\vec{x}, t) = \tilde{\phi}(\vec{x}, t) + \phi'(\vec{x}, t)
\]

We can use our filtered DNS fields to look at how the choice of our filter kernel affects this separation in wavespace.
Decomposition of turbulence for real filters

- The Gaussian (or box) filter does not have as compact of support in wavespace as the cutoff filter.
- This results in attenuation of energy at scales larger than the filter scale.
- The scales affected by the attenuation are referred to as *resolved SFSs*. 

\[ k = (k_1^2 + k_2^2 + k_3^2)^{1/2} \]