AMATH 590 - Project Proposal: Flow Reconstruction, Time-Stepping, and Parameterization of sub-grid scales using Neural Networks

Deepak Venkatanarasimhan(1940141)

May 14, 2021

Abstract

This project proposal discusses ML methods covered in my prior reviews and concepts learned in the course. Performing numerical simulations on Navier Stokes at scale and high resolution is both time and computational-resource-intensive. Current research shows the application of neural networks to close smoothed/ averaged versions of Navier Stokes such as RANS or coarse-grid resolved versions like LES. In this project, I plan to use the Johns Hopkins forced isotropic turbulence dataset. The dataset provides velocity and other variables at multiple time, and space scales [6]. To explore the data, I start with Fluid-flow reconstruction [5] and RNN-based time steppers [3]. Finally, I propose to build an NN-based parameterization of the relationship between sub-grid scales and integral scales in LES [4].

1 Project Proposal

I have discussed using ML's ability to capture fluid structure in space or over time in previous reviews. One could use ML to achieve efficient reconstruction from a sparse state, or understand parameterization of a model such as Smagorinsky [7]. Such Neural networks are not universal and are tied to the specific fluid one uses to train them. Albeit, they have significant practical applications in understanding, sensing, and control of fluids. I use the project as an opportunity to explore a large turbulence dataset [6] and study areas of work stated below. The dataset is explained in the next section.

- **Fluid Reconstruction** Use a shallow decoder as described in [5], to be able to reconstruct a high-resolution version of a fluid snapshot, from a coarse version. Doing the same leads to efficiency gain in computation, given the costs of collecting/sensing, setting up, and storing a high-resolution version.
- **Time-stepping/ Flow maps** Apply the technique discussed in [3] to capture the turbulent fluid's evolution with time. With such a Flow map, one could predict evolution in time and use it for control purposes.
- **Study Sub-grid Scales** Study interaction between Integral and sub-grid scales as defined at [2]. Compare learned models with closed-form models such as Smagorinsky. The ability to model fluids at sub-grid scales that are computationally hard to resolve is an active area of research [1].

Working on the above ML experiments help me explore the application of Neural Networks to Turbulence Data and get a sense of possibilities and limitations.

2 Data

The simulation parameters used to setup the database of forced isotropic turbulence are shared below:

1. The simulation is performed on a domain of $(0, 2\pi)$. The domain is divided into 1024 points across the 3 axes. So one has DNS output on 1024^3 spatial points.

2. 5028 time samples are available between $t = 0$ and $t = 10.056$ with $\delta_t = 0.002$.

3. Statistical characteristics of this data is available at [2]

One can download the data from a jupyter notebook using a web service as shown at get_data.ipynb. To extract the data, one needs to specify the beginning and ending space coordinates and the point in time. Downloading data at high resolution takes longer (especially if one uses interpolation), as expected. I have extracted data at low(8*8) and high(128*128) resolutions and shared them below for reference in Figure 1 and 2. Code to make these plots is at ExploreTurbulenceData.ipynb. I have used a spatial domain of $(0, 1)$ on all the three axes and evenly divided the domain by 8 for low-resolution, and 128 for high-resolution. The original spatial domain from $(0, 2\pi)$ is divided into 1024 parts. Extracting any resolution beyond this requires interpolation. I have used $(3, 4)$ for the temporal dimension, with a time-step of 0.2. Query time for extracting data at a fine discretization is high and must be considered during experiment set-up.

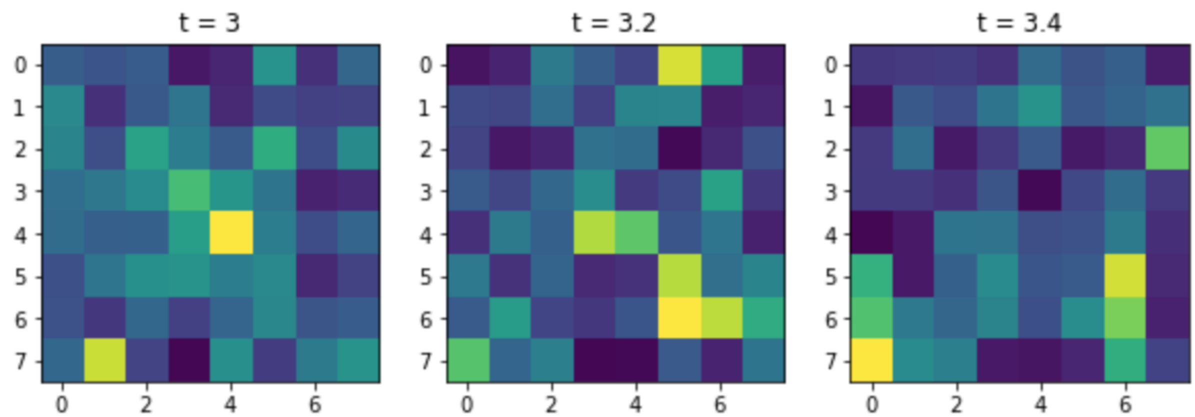

Figure 1: Low Resolution(8*8) Snapshot of the Fluid at a given z, and three points in time

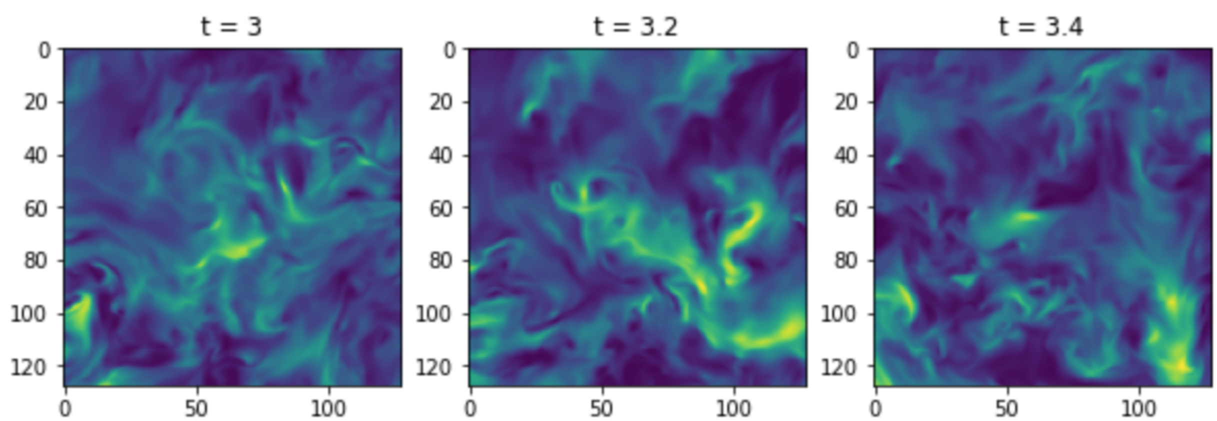

Figure 2: High Resolution(128*128) Snapshot of the Fluid at a given z, and three points in time

3 Expected Results and Learning

I expect flow-reconstruction performance to be good for cases within the training data(no extrapolation), as shared in the paper [5]. The flow map construction [3] may be harder to accomplish given that the turbulent data is not stationary; however, it is worth exploring to understand if setting up a multiscale structure helps. I am attempting parameterizing the flow at Kolmogorov scales to understand the data and parse out required variables from the simulation, as discussed in [7]. The ability to connect the unresolved sub-grid scales to a coarser grid leads to significant performance gains in overall modeling [1].

References

[1] Dmitrii Kochkov et. al. *Machine learning accelerated computational fluid dynamics*. 2021.

[2] E. Perlman et. al. *The Johns Hopkins Turbulence Databases (JHTDB) - Readme File*. 2008.

[3] Nathan Kutz et. al. *Hierarchical Deep Learning of Multiscale Differential Equation Time-Steppers*. 2020.

[4] Petros et. al. *Automating Turbulence Modeling by Multi-Agent Reinforcement Learning*. 2020.

[5] Steven Brunton et. al. *Shallow Neural Networks for Fluid Flow Reconstruction with Limited Sensors*. 2020.

[6] Yi Li et. al. *A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence*. 2008.

[7] Anikesh Pal. *Deep Learning parameterization of subgrid scales in wall-bounded turbulent flows*. 2019.