**Production Pipeline**

1. Generate Navier-Stokes flow snapshots: inputfile\_dns2d\_matlab.m

* Tunable parameters: Re, [force\_new.m] force\_amp, kappa\_max, kappa\_min
* Look for:
  + Range of Du/Dt and u
  + Vortex movements (bubbles tend to get trapped in vortices; pay attention to vortex behaviors for the sake of getting nontrivial statistics in the long run)
* Set record rate dwrite, taking into account:
  + Particle trajectory time step using RK4 is 2\*dwrite

1. Construct interpolant datafile

* Run collapse\_data.m and set the number of snapshots to include; output: single .mat file containing 3D matrices of flow velocities and material derivatives
* Run create\_interpolant.ipynb; set load file path and save file path

1. Generate trajectory data using interpolant created in the previous step: particle\_data\_gen.ipynb

* Set up interpolant file path in block 3
* Set up ep, R and dt; run a single particle trajectory, watch
  + Stability plots – slow manifold must be stable always (leave some margin)
  + Velocity comparison: flow vs. particle vs. 1st order slow manifold approximation. Ideally, there needs to be separation between flow and particle velocity, as well as between particle velocity and slow manifold approximation. This will make it more advantageous to use RNN-based method to improve the approximation.
* Use the same parameters as the single trajectory test. Set pts and traj\_len to get the official testing data set.
  + Save position data and examine scatter point animation using MATLAB: scripts/Get\_Particle\_Movie.m
  + Examine PDFs of material derivatives and local flow velocity experienced by particles: checkPDF.ipynb. Need to set data file path and sample frequency for calculating PDFs. May want to save figures for later reference.

1. Generate training data file (using cellular flow): train\_data\_gen.ipynb

* Set parameters: same ep, R and dt as in the previous step; also set number of samples and total time steps
* Set cellular flow parameters: A, B and w. Verify for a single trajectory
  + The stability criterion λmin MUST STAY ABOVE 0 AT ALL TIMES (may want to check a few different trajectories)
  + The trajectories have some nontrivial behaviors; also make sure there is room for improving the 1st order slow manifold approximation
* Check PDFs for generated data set
  + Make sure stability eigenvalue has no density for values smaller than 0
  + Compare PDFs for flow velocity U, material derivative DU/Dt and model target V-Vm with those for the testing data set obtained in the last point of step 3. Make sure that the support of each quantity for the training set has a wider range and is associated with higher densities.

1. Perform training: model\_training.ipynb

* Set up: model architecture, training file path, time step and (optionally) pre-trained weights
* Run model.fit for a specified number of batch size and epochs
  + Make sure that validation results are really good – otherwise there is no chance that the model performs well for a turbulent flow

1. Run *a priori* tests: model\_training.ipynb

* Set up: file path testfile, time index spacing ss
* Plot a priori (non-cumulative) prediction results and compare with targets; iterate through a few test cases to make sure that model performance is consistent
* Calculate and plot a priori RMSE for all test cases

1. Run *a posteriori* tests: ap\_predict.ipynb

* Make sure that the model structure is consistent with that defined in step 5
* Set up training file path or load the shift and scale of the input (assuming model input is standardized) directly

1. s