Supplementary Material

In this document, we discuss the neural network training process, and how to implement the model using the supplied model files.

1 Training the neural network model

The response variable for the generalized neural network model is scaled drift flux ($\frac{\overline{\phi}_s \tilde{v}_{d,z}}{\phi_{max} u_t}$) using five features:

- particle Reynolds number, Re_p
- dimensionless filter size, $1/\Delta_f^*$, where filter size Δ_f is the length of filter in meter, and is made dimensionless using $d_p\, Fr_p^{1/3}$ to get Δ_f^*
- scaled filtered solid volume fraction, $\overline{\phi}_s^*$, which is $\overline{\phi}_s$ scaled by $\phi_{s,max}$
- dimensionless filtered slip velocity $\tilde{u}_{slip}^*$, which is slip velocity made dimensionless with terminal settling velocity, u_t
- dimensionless axial direction gas-phase pressure gradient $\frac{\partial p_g^*}{\partial z}$, which is the pressure gradient made dimensionless with $\rho_s g$

In summary, the model reads

$$\frac{\overline{\phi}_s \tilde{v}_{d,z}}{\phi_{max} u_t} = f\left(Re_p,\ \frac{d_p\, Fr_p^{1/3}}{\Delta_f},\ \frac{\overline{\phi}_s}{\phi_{s,max}},\ \frac{1}{\rho_s g}\frac{\partial \overline{p}_g}{\partial z},\ \frac{\tilde{u}_{slip,z}}{u_t}\right) \tag{1}$$

We used one million data points which are randomly sampled from the twenty one fine-grid simulations. The data set is split into 0.7 million training examples, 60,000 validating examples, and 0.24 million testing examples. We used three fully connected hidden layers with 64, 32, and 16 nodes, respectively. We used ReLU activation functions for all three layers, Adam optimizer for the training, and mean absolute error as the cost function. The model was trained using mini batch with a batch size of 1024, and using 25 epochs.

2 Implementing the neural network model

During coarse-grid simulation, we use local flow quantities computed with Wen & Yu drag model at each grid as features to compute drift flux and drag correction. When features are fed into the model for prediction, they need to be in the order as shown in Eq.(1). Implementing the model requires two pre-processing steps.

1. Scale the flow variables using the parameters listed in Section 1
2. Standardize the scaled variables using the normalizing factors in the provided `.csv` files. $mean_i$ is stored in `mean.csv`, and std_i is in `std.csv`.

$$x_{i,norm} = \frac{x_i - mean_i}{std_i} \tag{2}$$

The neural network model is saved in the following two files, and are attached with this supplementary material:

- `nn_model.hdf5` contains model weights, which are the model parameters after training.
- `nn_model.json` contains model structure as mentioned above

We incorporated this model into OpenFOAM using a third party package named "keras2cpp" (https://github.com/pplonski/keras2cpp). Readers are referred to the source for implementation source code. This code converts the neural network architecture and weights in the two model files (listed above) to a single data file, and provideds C++ implementation for generating predictions based on the data file. We extended the OpenFOAM twoPhaseEulerFoam solver according to keras2cpp, so that during simulation, it reads the constructed neural network data file, collects input features from the scaled and normalized flow quantities, and then evaluates neural network model to predict scaled drift flux $\frac{\phi_s v_{d,z}}{\phi_{max} u_t}$. We also developed a small C++ package where we have a main program that calls prediction subroutine and provides a prediction of dimensional drift flux based on an input file that contains input variables: system Reynolds number, filter size, solid volume fraction, pressure gradient, and slip velocities. Within the package, we created a README file to explain how these subroutines work and provide a guideline for an inexperienced end-user. The package is attached to the supplementary material, and is available at https://github.com/yundij/ANN-sub-grid-Drag.

When solid volume fraction is less than 0.01 or greater than 0.55 ($\phi_s < 0.01$ or $\phi_s > 0.55$), the predicted drift flux is zero. These limits are placed to ensure no corrections at extremely dilute and dense regions. After the drift flux prediction, a drag correction term, H, can be computed as

$$H = 1 + \frac{\phi_s v_{d,z}\, u_{slip,z}}{\phi_s \|\boldsymbol{u}_{slip}\|^2} \tag{3}$$

We then use clipping to avoid drag correction predictions out of the physical range between 0.001 and 1.5. This range is determined empirically based on the distribution, where outliers due to prediction inaccuracy and numerical error are clipped to avoid simulation crash. Finally, we scale the Wen & Yu predicted drag force, $F_{Wen\&Yu}$ using H as

$$F_d = F_{Wen\&Yu} * H \tag{4}$$