Implementation of Neural Network Drag Model

1 Model details and utilization

1.1 Data preprocessing

- At each time step and in each cell in a coarse-grid simulation, we use the current values of particle volume fraction, ϕ, gas phase pressure gradient in the vertical direction, $\frac{\partial P}{\partial z}$, gas- and particle-phase velocity vectors, ($\mathbf{u}_{gas}$,$\mathbf{u}_{solid}$), and the cell Volume, V_{cell} to determine the drag force.
- First find the Wen-Yu drag force using the above particle volume fraction, the velocity vectors, particle size, and the gas phase physical properties. Note that for Euler-Euler F_{Wen-Yu} is computed per unit volume of the mixture, whereas for Euler-Lagrange it is computed per particle. As a part of this calculation, one would also evaluate the particle Reynolds number $Re_g = \frac{\rho_g|\mathbf{u}_{gas}-\mathbf{u}_{solid}|d_p(1-\phi)}{\mu_g}$. Refer to this force as F_{Wen-Yu}.
- The actual drag force is given by $F_D = F_{Wen-Yu} \cdot H$. The neural network model is used to find H as described below.
- For the purpose of drag correction calculation, we consider the slip velocity in the vertical direction. $u_{slip} = u_{gas,z} - u_{solid,z}$.
- Reference length in the simulation is defined as: $f_r = d_p \left(\frac{u_t^2}{gd_p}\right)^{\frac{1}{3}}$. Here u_t is the terminal settling velocity of the particle. For simulating the flow of monosized particles, this reference length is the same in all the cells and at all times..
- Then compute the following quantities:
  - Particle Reynolds number: $Re_p = \frac{\rho_g u_t d_p}{\mu_g}$. This is variable x_1 in the notes below. When monosized particles are taken into account, this quantity will be the same in all the cells and at all times.
  - The ANN uses the inverse of the cell size as input. So, one should compute the quantity $\chi = \frac{1}{\Delta^*} = \frac{f_r}{V_{cell}^{1/3}}$. This is variable x_2 in the notes below.
  - Scaled particle volume fraction, $\phi^* = \frac{\phi}{\phi_{max}}$, where $\phi_{max} = 0.64$. This is variable x_1 in the notes below.
  - Scaled pressure gradient, $\frac{\partial P^*}{\partial z} = \frac{1}{\rho_s g}\frac{\partial P}{\partial z}$. This is variable x_3 in the notes below.
  - Scaled slip velocity, $u_{slip}^* = \frac{u_{slip}}{u_t}$. This is variable x_4 in the notes below.
- These five quantities, x_j, j=1,5 are supplied to the ANN code after the following normalization.

$$\widetilde{x}_j = \frac{x_j - \mu_j}{\sigma_j}, \tag{1}$$

1.2 Neural Network model

The neural network model information is saved in two files, which can be found in *NNetModel* folder:

- `DF.json`: in this file, the neural network's architecture is contained.
- `DF.HDF5`: in this file, the weights for the considered architecture are stored.

Variable	Minimum value allowed	Maximum value allowed
Solid density, ρ_s [kg/m^3]	960	2100
Gas density, ρ_g [kg/m^3]	0.13	10
Gas viscosity, μ_g [Pa $\cdot$ s]	$1.84{\cdot}10^{-5}$	$5.23{\cdot}10^{-5}$
Particles diameter, d_p [μm]	75	300
Terminal velocity, u_t [m/s]	0.14	1.92
Particles Re, Re_p [-]	0.75	40.57

Table 1: Ranges of phase properties and characteristic quantities for the simulations considered in the training data which can be used in the data-driven numerical simulations.

To implement this neural network model into `OpenFOAM`, an open-source third party package was used: `keras2cpp` (freely available at: *https://github.com/pplonski/keras2cpp*). This code converts the neural network architecture and weights which are found in the two aforementioned files into a single data file, called *DFnet*. When incorporated into the CFD solver, *DFnet* reads the constructed neural network data file, collects the input features from the scaled and normalized flow quantities, and finally predicts the scaled drift flux, which is referred to in this notes as $Y = \phi_s^* u_d^*$ (where u_d is the drift velocity as in Eq. 28 of [1]).

$$H = 1 + \frac{Y}{\phi^* u^*_{slip}}, \quad 0.01 < \phi < 0.55 \tag{2}$$

When implementing the neural network, we adopted limits for drift flux corresponding to dense and dilute regions, i.e., when solid volume fraction is lower than 0.01 or higher than 0.55, drift flux is imposed to be equal to zero. This is mainly due to the fact that the network has shown difficulty in learning the physics at the extreme solid volume fractions. Thus, when the system is in a very dilute or dense state, i.e., the status of complete void or close packing is observed, an homogeneous drag model should be applied and therefore no correction is needed. The zero drift flux condition is enforced in this conditions to ensure this result. Finally, this drag correction term is used to scale the drag force that was predicted by the initial Wen&Yu model (F_{WenYu}) as follows

$$F_d = F_{WenYu} \cdot H \tag{3}$$

1.3 Model utilization

The neural network was trained by means of a dataset consisting of 10,000,000 samples of the five nondimensional variables described in Sec. 1.1: these samples were collected from different two-fluid simulations carried out with different boundary conditions. In light of this, the network is capable to make accurate predictions when different types of gas, particles, and velocities are taken into account. The ranges for the physical properties the network can deal with (in terms of minimum and maximum value allowed for each variable, respectively) are summarized in Table 1.

References

[1] Jiang, Y., Chen, X., Kolehmainen, J., Kevrekidis, I. G., Ozel, A., and Sundaresan, S. Development of data-driven filtered drag model for industrial-scale fluidized beds. *Chemical Engineering Science 230* (2021), 116235.