MSBM Project Documentation

Michael K. Davis

September 8, 2025

1 PINN Architecture

We are solving the 2D MSBM equations at steady state along the x-direction, so the solution is varies only along the y-direction. Furthermore, the y-direction velocity is zero at steady state. Thus, our only input is y, and our solution is u and ϕ, where u is the x-direction velocity and ϕ is the particle volume fraction.

We use two separate PINNs for u and ϕ, respectively, which are denoted in the style of [1] as

$$u(\theta_u, y) = \Gamma_u(\mathcal{M}_u(\theta_u, I_u(y))),$$
$$\phi(\theta_\phi, y) = \Gamma_\phi(\mathcal{M}_\phi(\theta_\phi, I_\phi(y))).$$

The input transformations are $I_u(y)$ and $I_\phi(y)$ for each PINN, where

$$I_u(y) = \frac{2y}{H} - 1$$

normalizes the input such that $I_u(y) \in [-1, 1]$; this range is beneficial for the choice of activation function, which are discussed later.

As for $I_\phi(y)$, we have

$$I_\phi(y) = \alpha_i \cdot \exp\left(-\beta_i \left(\kappa_i|\frac{2y}{H} - 1|\right)^{(\gamma_i+1)}\right),$$

where the input is still similarly normalized, but then passed through a modified Gaussian expansion for learnable parameters $\alpha_i,\ \beta_i,\ \gamma_i, \kappa_i > 0,\ i \in [1, N_\phi]$, where N_ϕ is the number of neurons for ϕ's PINN. The modified Gaussian expansion was chosen because its solution for y is similar in shape to the expected solution of ϕ, and it is modified by including $(\gamma_i + 1)$ rather than a simple square; this allows for a high-frequency solution for ϕ, but only where high-frequencies are expected. The $+1$ is included to prevent singularities, and can be any non-negative real number. The modified Gaussian expansion is more problem-specific and stable than a Fourier expansion.

As for the core neural networks themselves, $\mathcal{M}_u(\theta_u, I_u(y))$ and $\mathcal{M}_\phi(\theta_\phi, I_\phi(y))$ are

$$\mathcal{M}(\theta_u, I_u(y)) = \sigma\big(W_u^{(L_u)}\,\sigma\big(W_u^{(L_u-1)} \cdots \sigma\big(W_u^{(1)} \cdot I_u(y) + b_u^{(1)}\big) \cdots + b_u^{(L_u-1)}\big) + b_u^{(L_u)}\big),$$
$$\mathcal{M}(\theta_\phi, I_\phi(y)) = \sigma\big(W_\phi^{(L_\phi)}\,\sigma\big(W_\phi^{(L_\phi-1)} \cdots \sigma\big(W_\phi^{(1)} \cdot I_\phi(y) + b_\phi^{(1)}\big) \cdots + b_\phi^{(L_\phi-1)}\big) + b_\phi^{(L_\phi)}\big).$$

where the number of layers for each PINN is denoted as L_u and L_ϕ, and the weights and biases are collectively denoted as $\theta_u = \{W_u^{(l)}, b_u^{(l)}\}_{l=1}^{L_u}$ and $\theta_\phi = \{W_\phi^{(l)}, b_\phi^{(l)}\}_{l=1}^{L_\phi}$. The weights and biases are in this case matrices with sizes that depend on the number of neurons N_u and N_ϕ, respectively. Neurons contribute to PINN width, while layers contribute to PINN depth. As for the activation function, σ, it is the same hyperbolic tangent function for both PINNs.

$$\sigma = \tanh(\cdot) = \frac{e^{(\cdot)} - e^{-(\cdot)}}{e^{(\cdot)} + e^{-(\cdot)}}.$$

Lastly, the output transformations are $\Gamma_u(\cdot)$ and $\Gamma_\phi(\cdot)$, where

$$\Gamma_u(\cdot) = (\cdot) \cdot (1 + I_\phi(y)) \cdot (1 - I_\phi(y)),$$

$$\Gamma_\phi(\cdot) = \frac{\phi_m}{1 + e^{-(\cdot)}} \cdot (1 + I_\phi(y)) \cdot (1 - I_\phi(y)),$$

which serve to enforce the zero Dirichlet boundary conditions as hard constraints for both u and ϕ, and also bound the solution for ϕ such that $\phi \in [0, \phi_m]$.