Korean J. Chem. Eng., 40(1), 57-66 (2023)
DOI: 10.1007/s11814-022-1340-8

pISSN: 0256-1115
eISSN: 1975-7220

A deep learning approach using temporal-spatial data of computational fluid dynamics for fast property prediction of gas-solid fluidized bed

Pengfei Qin*,, Zhaojie Xia*,**, and Li Guo*,**,***,†**

*State Key Laboratory of Multiphase Complex Systems, Institute of Process Engineering, Chinese Academy of Sciences, Beijing 100190, P. R. China
**School of Chemical Engineering, University of Chinese Academy of Sciences, Beijing 100049, P. R. China
***Innovation Academy for Green Manufacture, Chinese Academy of Sciences, Beijing 100190, P. R. China
(*Received 11 June 2022 • Revised 29 September 2022 • Accepted 6 November 2022*)

Abstract−To deal with the critical issue of long computational time in practical application of computational fluid dynamics (CFD), this paper presents a new approach of deep learning for voidage prediction (DeepVP) that couples short time CFD simulations (limited CFD iterations) with the deep learning method to accelerate the 2D voidage distribution prediction for a gas-solid fluidized bed at steady state. Short time CFD simulations are first performed to obtain a sequence of voidage distribution images containing the temporal-spatial property of a gas-solid fluidized bed of the early period. A deep learning model is built to predict the voidage distribution at steady state, which is achieved by implementing multi-scale convolutional neural networks based on the sequence of voidage images. The case study results for a bubbling bed show that the voidage distribution at steady state for the bubbling bed can be predicted with comparable accuracy of conventional CFD simulations at about 1/30th computational cost. Moreover, the DeepVP method exhibits better extrapolation capability than the deep learning approach merely based on CFD condition parameters.

Keywords: Gas-solid Fluidized Bed, CFD Surrogate Model, Deep Learning, Multi-scale Convolutional Neural Network, Voidage Distribution Prediction

INTRODUCTION

The gas-solid fluidized bed is widely used in many chemical industrial processes of due to its simple structure, good gas-solid contact, and high efficiency of heat and mass transfer [1-3]. The voidage distribution at steady state under different conditions is an important property of such beds. Conventional experimental techniques for characterizing the performance of these beds are usually time-consuming and expensive [4]. The numerical simulation method of computational fluid dynamics (CFD) to gas-solid fluidized beds significantly reduce the reliance on experiments in their design [5]. CFD also provides a detailed analysis of flow patterns, which is not available in experiments. Therefore, CFD simulations have become an essential computational method complementing experiments [6,7]. However, a critical issue is their long computational time, which is required in numerically solving the Navier-Stokes equations that govern the motion of fluids [8-11].

With the successful application of the deep learning method in computer vision [12,13] and natural language process [14,15], recent developments in machine learning have found potential for new alternative methods in simulation of complex fluid motion for speeding up numerical simulation without compromising accuracy [16]. There is increasing interest in the development of a surrogate model of CFD with deep learning method. Some attempts have appeared recently to combine deep learning with CFD simulation to reduce the computation cost and speed up the simulation process [17,18]. Some researchers take the straightforward approach of using the CFD simulation condition parameters as input to predict the simulation results directly. Li et al. [19], Masoumi et al. [20], and Bakhtiari et al. [21] adopted a scheme of directly predicting the final state of a CFD simulation with input conditions, which is referenced as parameter prediction method (PPM) in this work. Li et al. [19] constructed a deep neural network to predict the peak overpressure inside a tank from 11 parameters, including liquid temperature, liquid filling percentage, tank size, etc. Masoumi et al. [20] employed the CFD method to simulate the operating state of the solar collector under different parameter conditions. Then they built a deep learning model to predict the collector's outlet water temperature from the shape and operating parameters. Bakhtiari et al. [21] took advantage of the CFD method to simulate cycloid thrusters under different conditions. The kinetic performance of the cycloid thrusters was predicted by training a neural network based on the CFD data. Specifically, the thrust and torque are predicted from parameters of blade number, pitch, propulsion coefficient, etc.

Other attempts using deep learning algorithms to accelerate CFD simulations take the limited amount of CFD simulation results as input to perform prediction [22,23], of which the prediction model contains rich evolution property data of the process simulated. Bazai et al. [22] and An et al. [23] used the so called rolling prediction method. Bazai et al. [22] combined CFD with deep learning to predict the behavior of fluidized beds without solving transfer equations. They take the first ten frames of the CFD simulation as input

†To whom correspondence should be addressed.
E-mail: lguo@ipe.ac.cn

to predict the next frame and then add the newly predicted frame to the input to predict the next frame. By repeating this process, all the desired frames can be generated. An et al. [23] proposed a new method for turbulent combustion simulation based on the deep learning method. The method takes the flow field data at time t_0 as input to predict the frame at the next time step, which is then used as the input recursively to continue predicting the next frame. These works employ rolling prediction methods to accelerate CFD simulation.

These earlier attempts that combine the deep learning method with CFD simulation show the feasibility and potential of deep learning in accelerating CFD simulation. However, the model trained by the PPM scheme [19-21] has good performance within the CFD condition parameter range of the training dataset but has poor extrapolation performance. The rolling prediction method [22,23] takes advantage of information from CFD simulation; it suffers, however, not only long prediction time but also accumulated error, which hardly makes it applicable for accurate property prediction at steady state.

This paper proposes a new approach for predicting the voidage distribution of a gas-solid fluidized bed at steady state using a limited amount of CFD simulation results as the input of a deep learning-based prediction model. The framework is called deep learning for voidage prediction with CFD for gas-solid fluidized bed (DeepVP). Short-time CFD simulations (limited CFD iterations) are first performed to obtain a sequence of voidage distribution images in a gas-solid fluidized bed. A deep learning model is built to directly predict the voidage distribution at steady state using the sequence of voidage images obtained. The application of DeepVP to a bubble-bed shows 30 times faster prediction of voidage distribution than the conventional CFD simulation method.

The rest of the paper is organized as follows. In section 2, the scheme of the DeepVP method is overviewed. The network structure of DeepVP is described in detail. The model building process and assessment for the voidage prediction of a bubbling bed is given as a case study of the DeepVP method in section 3. The CFD simulation of bubbling gas-solid fluidized bed, dataset preparation, and performance metrics are described first. Then, the model validation of DeepVP, acceleration performance of DeepVP, generalization capability of the DeepVP method, and the prediction accuracy dependency on the number of CFD simulations are presented and discussed. The results and discussion show the effectiveness and potential of the proposed DeepVP method. The conclusion is given in the last section.

METHOD

The DeepVP method in this work combines the domain knowledge of a gas-solid fluidized bed with deep learning through building a prediction model to speed up the acquisition of the voidage distribution at steady state. The essence of the method is to use a set of temporal-spatial properties representing the gas-solid motion of a fluidized bed in early state to predict the state property at final steady state. The temporal-spatial property set described with CFD simulation is mapped and converted into an image sequence, while the prediction target of the spatial property is an image. Therefore, the spatial property prediction becomes a problem of image prediction, which can be properly described with deep learning method. A multi-scale convolutional neural network is adopted, where a low-resolution scale is used to predict the macroscopic structure of the image, and a high-resolution scale is to fill in the microscopic details. The neural networks of multiple scales cooperate to obtain good prediction accuracy. The method implementation is tested with voidage distribution for a bubbling bed.

1. The Scheme of the DeepVP Method

As shown in Fig. 1, the DeepVP method couples CFD simulation and deep learning. The purpose of incorporating domain knowledge into the deep learning prediction model of voidage distribution is achieved by short time CFD simulation. K (K≪N) steps of CFD simulation were performed to get the sequence of voidage images. Since each iteration of CFD simulation is based on the physical laws of mass conservation, momentum conservation, and energy conservation, the sequence of voidage images extracted from the K steps simulation contains domain knowledge of the gas-solid fluidized bed, provided K iterations are sufficient for characterizing the early period operation state of gas-solid fluidized bed.

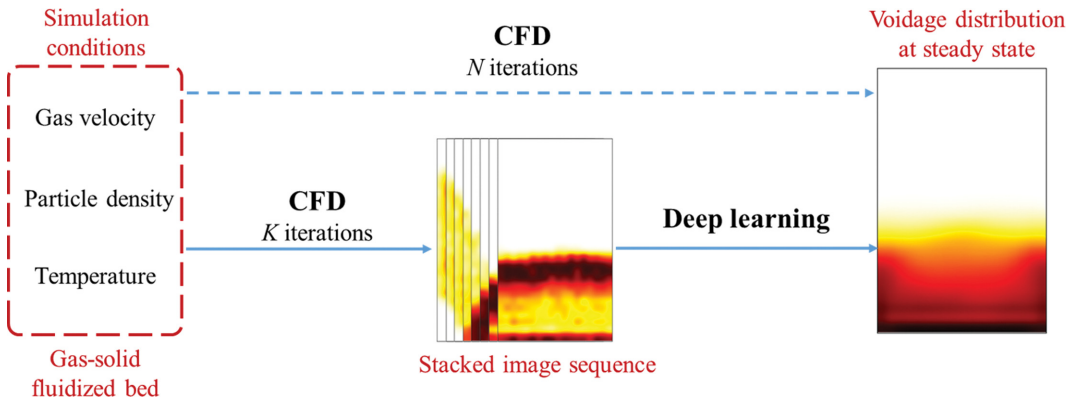

Fig. 1. Scheme overview of the proposed framework of DeepVP for voidage prediction of gas-solid fluidized bed with CFD and deep learning, where the short time CFD simulation results (K iterations and K≪N) are as input and the voidage distribution at steady state are as the output of DeepVP model. The conventional CFD simulations (blue dashed line) should have N iterations to obtain the final steady state, which is drawn as a reference.

Deep learning prediction is aimed at creating a prediction model for the voidage distribution prediction at steady state quickly. The image sequence is used as the information transmission medium so that rich information in two dimensions of time and space are integrated in the model-building process. The images carry spatial information, while their sequences carry temporal information. Such image sequences correspond implicitly to the temporal-spatial evolution of the physical quantities in the CFD simulation for pressure, concentration distribution, acceleration and velocity of gas and solid particle. Therefore, the rich physical knowledge of the gas-solid fluidized bed can be carried into the deep learning model by the image sequence.

Now the problem of voidage prediction in the gas-solid fluidized bed is mapped into a problem of image prediction that a deep learning scheme is applicable. In DeepVP, multi-scale convolutional neural networks are employed. Following the DeepVP, the targeted voidage distribution at steady state obtained from N iterations of CFD simulations is expected to be predicted efficiently with a deep learning prediction model on the basis of much fewer K iterations of CFD simulation (K≪N) for the same initial simulation condition parameters.

2. Structure and Implementation of Multi-scale Convolutional Neural Network

The network structure employed in DeepVP method is a multi-scale convolutional neural network, which is implemented using a three scale structure as show in Fig. 2. The convolutional neural network (CNN) is a feedforward neural network with a deep structure that contains convolutional computations [24]. In contrast to the fully connected neurons between each layer of the ordinary neural network, the convolutional neural network connects the neurons between each layer through convolution calculation. Thus, the convolutional computation of CNN is characterized with three notable features of parameter sharing, sparse connections, and equivariant representation [25]. The convolutional neural network requires fewer parameters and is more capable of processing image data.

As shown in Fig. 2, the neural network of this work is a convolutional neural network with three scales. The input of the first scale is an image sequence of one-fourth the size of the original image. Subsequently, the image sequence goes through four convolution layers, and the channel after each convolution layer is 128, 256, 128, and 1, respectively. Finally, an image of $\frac{1}{4}w \times \frac{1}{4}h$ is generated. The input of the second scale is a sequence of images which is half the size of the original image. Then, the image sequence goes through five convolution layers, and the channel after each convolution layer is 128, 256, 256, 128, and 1, respectively. Finally, an image of $\frac{1}{2}w \times \frac{1}{2}h$ is generated. The input of the third scale is the image se-

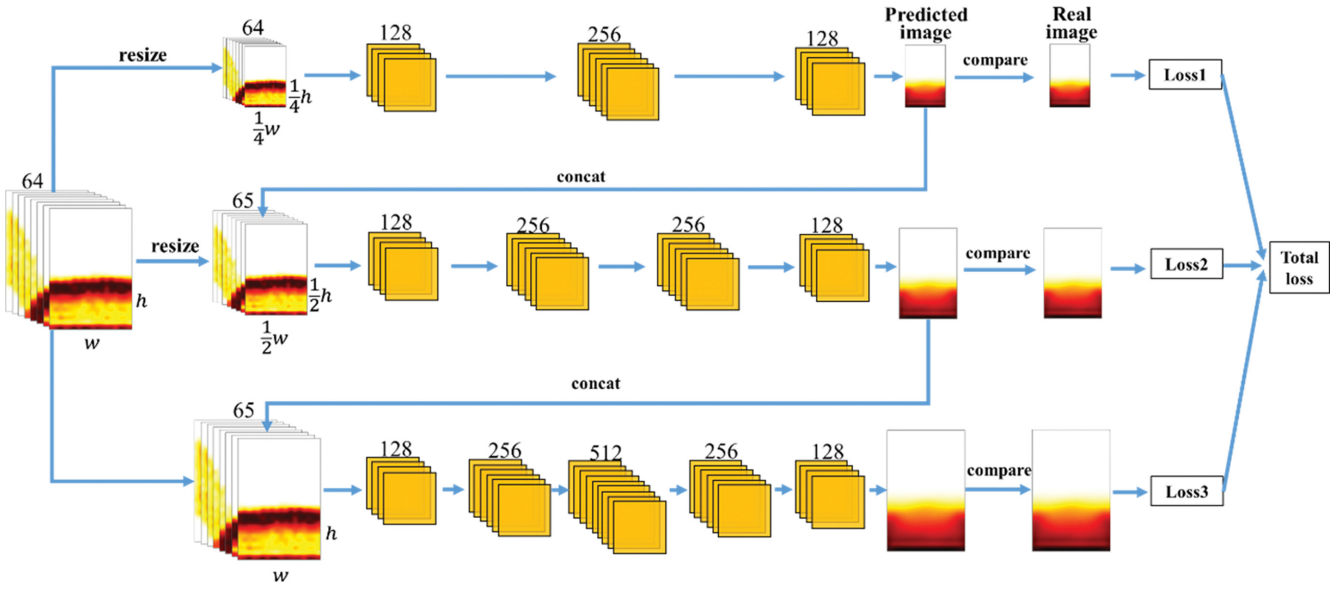

Fig. 2. Neural network structure in DeepVP for predicting the voidage image of gas-solid fluidized beds. The network contains three scales, and the input of each scale is an image sequence resized from the original image sequence. In each scale network, the main structure is a convolutional neural network. The final loss function is the sum of the loss functions of the three scale networks.

Table 1. Details of convolutional neural networks for building the voidage prediction model

Network attributes	Scales		
	S1	S2	S3
Layer number	4	5	6
Feature maps	128, 256, 128, 1	128, 256, 256, 128, 1	128, 256, 512, 256, 128, 1
Kernel size	3, 3, 3, 3	5, 3, 3, 3, 5	5, 3, 3, 3, 3, 5

quence of the original image size. Similarly, the image sequence goes through six convolution layers, and the channel after each convolution layer is 128, 256, 512, 256, 128, and 1, respectively. Finally, an image of $w \times h$ is generated. Table 1 shows the convolution layer's output channel and convolution kernel size in each scale, where S1, S2, and S3 represent three scales, respectively. The three-scale neural network cooperates in obtaining higher prediction accuracy. Similar ideas have been utilized in works such as Mathieu [26], Aigner [27], Zhang [28], and achieved good results.

To improve the accuracy and stability of the CNN model prediction, the image sequence for each pixel was normalized to $[-1, 1]$. In addition, the input to the CNN model is an image sequence that is achieved by stacking the images of different time steps into the channel dimension in order. The image sequence will be convoluted for multi-times, and the ReLU function is adopted as the activation function for all convolutional layers, except for the last convolutional layer where the tanh function is used to control the output range within $[-1, 1]$. After the image is generated, the loss function of this scale is acquired by comparing it with the real image.

Two loss functions are used for the CNNs training in this work: L_1 and GDL (Gradient Difference Loss) [26] loss function. The loss function L_1 is used to calculate the absolute value error between pixels and is defined as follows:

$$L_1 = \frac{1}{n}\sum \left| V_{i,j} - \tilde{V}_{i,j} \right| \tag{1}$$

where V_{ij} represents the true value of row i and column j in the voidage image, and $\tilde{V}_{ij}$ represents the predicted value of row i and column j in the voidage image. n is the total number of pixels in the voidage image.

GDL is employed to calculate the deviation between pixel gradients, defined as follows.

$$L_{gdl} = \frac{1}{n}\sum \left\| V_{i,j} - V_{i-1,j} \right| - \left| \tilde{V}_{i,j} - \tilde{V}_{i-1,j} \right| + \left\| V_{i,j} - V_{i,j-1} \right| - \left| \tilde{V}_{i,j} - \tilde{V}_{i,j-1} \right\| \tag{2}$$

The loss function of each scale L_{scale} in the neural network is the sum of the L_1 loss function and the GDL loss function of the layer.

$$L_{scale} = L_1 + L_{gdl} \tag{3}$$

The network structure of this work has three scales, and each scale calculates a loss function value. The total loss function L_{total} of the entire neural network is the sum of the three scale loss function values.

$$L_{total} = L_{scale1} + L_{scale2} + L_{scale3} \tag{4}$$

The neural network of this work was implemented by TensorFlow 1.14 [29]. The neural network was trained using the gradient descent algorithm [30]. Specifically, the Adam optimization algorithm was chosen [31]. The gradient descent algorithm is the most commonly used optimization algorithm in neural networks, and the Adam algorithm has good performance and stability in most cases.

In neural network design, the setting of hyper-parameters is very important. The learning rate in this work is set to 0.0001, the batch size is 32, and the maximum iteration number is 120,000. A batch sample is randomly selected from the training set and fed to the neural network for training in each iteration. The NVIDIA Tesla V100 GPU with 32 GB video memory was used for neural network training to accelerate the training process. The typical time for training is about up to two hours in total.

3. Dataset Preparation

In DeepVP, the CFD simulation results are the input for the deep learning model. The widely used bubbling bed is adopted for implementation of the DeepVP model. The shape of the bubbling bed used in this work is a cylinder with a diameter of 0.044 m and a height of 0.12 m. The solid particles in the cylinder are poppy seeds [32]. The gas is blown into the cylinder from the distribution plate at the bottom, driving the movement of the solid particles. The GPU-enabled CFD simulation code of Zhao Peng [33] for a soft sphere CFD-DEM method implementation was used to simulate the process of the bubbling bed from the initial state to the steady state. A summary of main equations and the detailed basic information of the CFD simulations are provided as supporting information. The method details can be found in Zhao's work [33].

The raw dataset consists of a set of results of CFD simulations under different initial conditions. Many factors affect the voidage image at steady state in a bubbling bed. Generally speaking, the most important influencing factors are fluid gas velocity, particle density, and temperature. To ensure the normal fluidization of the bubbling bed, the gas velocity ranges from 0.4 m/s to 0.85 m/s, the particle density is 800 kg/m^3 to 1,000 kg/m^3, and the temperature is 0 °C to 100 °C. For preparing the dataset with a wide coverage of different operating conditions of the bubbling bed, a detailed scheme for the conditions of the CFD simulation was made. The gas velocity takes values at intervals of 0.05 m/s, with a total of ten optional values: 0.4 m/s, 0.45 m/s, …, 0.85 m/s. The particle density has optional values of 800 kg/m^3, 900 kg/m^3, and 1,000 kg/m^3. The temperature has ten optional values: 0 °C, 10 °C, …, 90 °C. The CFD simulations were performed for all the 300 possible combinations of these conditions, constituting 300 sets of initial conditions for the simulation. For each simulation, K=5,120 (t=0.64 s), N=160,800 (t=20.10 s), the data of the targeted voidage distribution of steady-state were extracted during 5-20 s. The input data of the training and testing sets were extracted when K=5,120. The complete 300 simulations were divided in a ratio of 8 : 2, where the training set contains 240 simulations, and the test set contains 60 simulations (referred to as Test-dataset 1). All the CFD simulation conditions of the bubbling bed are summarized in Table 2.

To verify the interpolation and extrapolation generalization capability of the model, an interpolation test set (referred to as Test-dataset 2) and an extrapolation test set (referred to as Test-dataset 3) are also constructed in addition to the 60 simulations of Test-dataset 1. The particle density of the interpolated test set is 850 kg/m^3 and 950 kg/m^3. And the particle density of the extrapolated test set is 700 kg/m^3 and 1,100 kg/m^3, which is beyond the particle density range of 800-1,000 kg/m^3 for the training data set. To control the computational cost for these two test sets, the intervals of gas velocity and temperature are doubled, so there are only five optional values to be selected. 50 simulations were performed for both the interpolation test set (Test-dataset 2) and the extrapolation test set (Test-dataset 3). It takes about 2 hours of program running time to complete a single simulation, which simulates the evolution pro-

Table 2. Condition parameters of the CFD simulations for preparing data samples of the DeepVP voidage distribution prediction model for a bubbling bed

Dataset	Gas velocity (m/s)	Particle density (kg/m^3)	Temperature[a] (°C)	Total sample number
Training set	0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85	800, 900, 1,000	0, 10, 20, 30, 40, 50, 60, 70, 80, 90	30,720
Test-dataset 1	0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85	800, 900, 1,000	0, 10, 20, 30, 40, 50, 60, 70, 80, 90	7,680
Test-dataset 2	0.40, 0.50, 0.60, 0.70, 0.80	850, 950	0, 20, 40, 60, 80	6,400
Test-dataset 3	0.40, 0.50, 0.60, 0.70, 0.80	700, 1,100	0, 20, 40, 60, 80	6,400
Total				51,200

[a]The temperature is set indirectly by changing both the gas density and gas viscosity simultaneously.

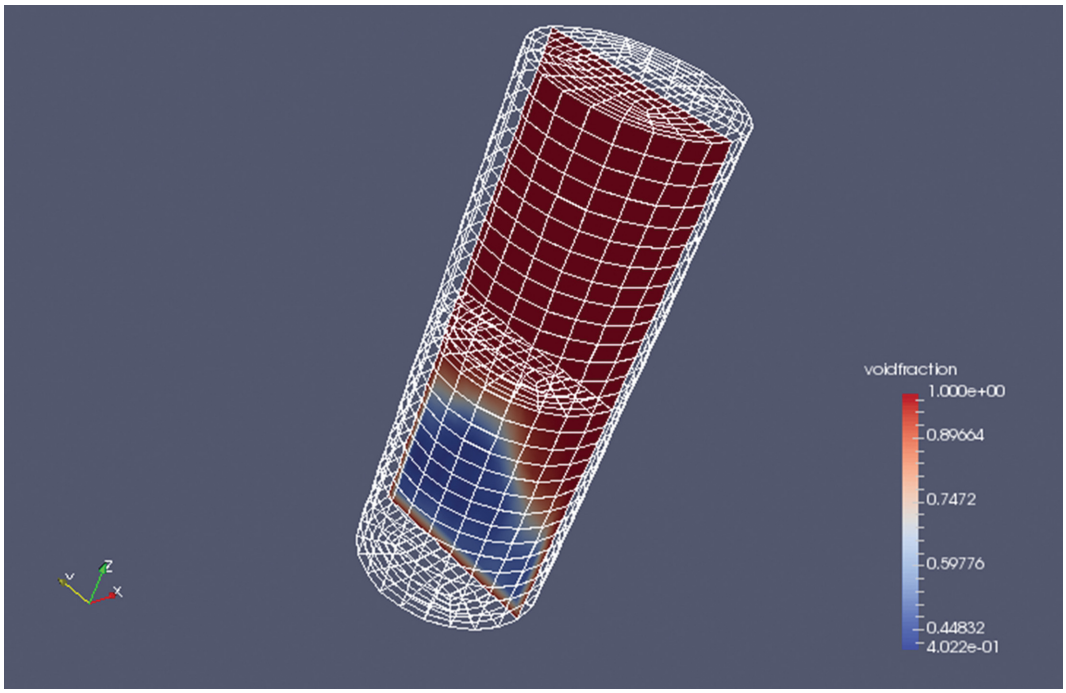

Fig. 3. An example of an axially sliced grid at 237 steps along the diameter during the CFD simulation for a bubbling bed, at U_g=0.80 m/s, ρ_s=800 kg/m^3, T=50 °C.

cess of 20 s of physical time.

After the raw data of the simulation were prepared, preprocessing was performed to convert the raw data into the image sequence for the deep learning model. Firstly, spatial preprocessing is performed on the simulation results. The cylindrical bubbling bed is sliced axially along the diameter through the axle center to obtain a maximal two-dimensional voidage image of the cylinder. Fig. 3 shows an example of an axially sliced grid along the diameter during the CFD simulation. The width and height of this 2D section are 18 and 28, respectively. The width and height of the section are utilized directly as the size of the input image. Secondly, the data is preprocessed in the time dimension. The image sequence data with a length of 64 were extracted from 320 frames output of 5120 CFD iterations within the total 160,800 iterations of the 20 s CFD simulation, where the simulation time step is 1.25e-05 s, and the output interval is 16 time steps. Besides, the targeted voidage image at steady state was averaged from the voidage images of the 5-20 s frames in the CFD simulation.

Data augmentation is performed finally on the preprocessed data to best use the 320 images for the default input length of our neural network of 64. Among the 320 images, any sequence segment with a length of 64 corresponds to the voidage image of the simulation at steady state. Data augmentation is performed based on this idea by continuously extracting 64 frames from the total 320 frames only by variation of the starting point with an interval of 2. Thus, each simulation yields 128 samples. There are 51,200 image data samples extracted from the total 400 simulations. The total sample numbers of the training set and the three testing sets are shown in Table 2.

4. Performance Metrics

Three metrics were adopted to evaluate the prediction accuracy of the neural network built in this work: Mean absolute error (MAE),

peak signal to noise ratio (PSNR) [34], and sharpness difference (Sharpdiff) [26].

MAE is defined as follows:

$$MAE=\frac{1}{n}\sum\left|V_{i,j}-\tilde{V}_{i,j}\right| \quad (5)$$

where V_{ij} represents the true value of row i and column j in the voidage image, $\tilde{V}_{ij}$ represents the predicted value of row i and column j in the voidage image. n is the total number of pixels in the voidage image. MAE measures the absolute value error between the real and generated images' corresponding pixels. The smaller the MAE value, the higher the model's prediction accuracy.

PSNR is defined as follows:

$$PSNR=10\log_{10}\frac{MAX_V^2}{\frac{1}{n}\sum(V_{i,j}-\tilde{V}_{i,j})} \quad (6)$$

where the definitions of $V_{i,j}$, $\tilde{V}_{i,j}$ and n are the same as above. MAX_V^2 represents the square of the maximum possible value of voidage, which is 1.0 in our work. The unit of PSNR is dB. The larger the PSNR value, the higher the model's prediction accuracy.

Sharpdiff is defined as follows:

$$Sharpdiff=10\log_{10}\frac{MAX_V^2}{\frac{1}{n}(\sum_i\sum_j\left|(\nabla_i V+\nabla_j V)-(\nabla_i\tilde{V}+\nabla_j\tilde{V})\right|)} \quad (7)$$

$$\nabla_i V=\left|V_{i,j}-V_{i-1,j}\right| \quad (8)$$

$$\nabla_j V=\left|V_{i,j}-V_{i,j-1}\right| \quad (9)$$

$$\nabla_i\tilde{V}=\left|\tilde{V}_{i,j}-\tilde{V}_{i-1,j}\right| \quad (10)$$

$$\nabla_j\tilde{V}=\left|\tilde{V}_{i,j}-\tilde{V}_{i,j-1}\right| \quad (11)$$

where the definitions of MAX_V^2 and n are the same as above. Sharpdiff is used to measure the gradient error between real and generated images. The larger the Sharpdiff value, the higher the model's prediction accuracy.

RESULTS AND DISCUSSION

1. Model Validation of the DeepVP Method

The DeepVP method was validated using CFD simulation of a bubbling bed under three different conditions of U_g=0.45 m/s, ρ_s=1,000 kg/m^3, T=20 °C; U_g=0.60 m/s, ρ_s=900 kg/m^3, T=80 °C; U_g=0.80 m/s, ρ_s=800 kg/m^3, T=50 °C. To validate the DeepVP model, the voidage images predicted by the DeepVP method and the CFD simulation results were compared.

To test the CNN parameter choice, a set of new models were built by adding or reducing network scales, layers and nodes. The testing dataset and training time of new models with varied scale, number of layers and number of nodes of each layer were tested and the results are listed in Tables 3-5, using the parameters of this work as baseline highlighted in bold. As shown in Tables 3-5, the choice of this work is good, that is, a tradeoff of performance of testing dataset and training time.

To validate the prediction model further, the DeepVP model was compared with the PPM method using Test-dataset 1. As mentioned in section 1, PPM is a method of directly predicting the final state of a CFD simulation with input condition parameters. For the purpose of validating the DeepVP method, the PPM method was adopted as a reference method. The PPM method for the voidage prediction of the bubbling bed was implemented in this

Table 3. Performance of testing dataset and training time of DeepVP for varied CNN network scale

Scale	Layers[a]	MAE (%)	PSNR	SharpDiff	Training time (min)
1	4	1.07	34.1	20.1	23
2	4, 5	0.97	34.9	20.7	30
3 (this work)	**4, 5, 6**	**0.9**	**35.8**	**21.1**	**55**
4	4, 5, 6, 6	0.9	35.7	21	67

[a]For the structure of each layer, please refer to Table 1.

Table 4. Performance of testing dataset and training time of DeepVP for varied CNN network scale, layer number and nodes of each layer

Layer increment	MAE (%)	PSNR	SharpDiff	Training time (min)
−2	1.025	34.2	20.1	26
−1	0.935	35.3	20.7	33
0 (this work)	**0.9**	**35.8**	**20.95**	**55**
1	0.915	35.6	20.9	71

Table 5. Performance of testing dataset and training time of DeepVP for varied CNN network scale, layer number and nodes of each layer

Nodes number	MAE (%)	PSNR	SharpDiff	Training time (min)
/2	0.935	35.4	20.7	30
***1 (this work)**	**0.9**	**35.8**	**20.95**	**55**
*2	0.895	35.7	20.95	100

Table 6. Network structure for PPM

Network attributes	Generator	Discriminator
Layer number	6	5
Feature maps	128, 256, 512, 256, 128, 1	128, 256, 512, 1

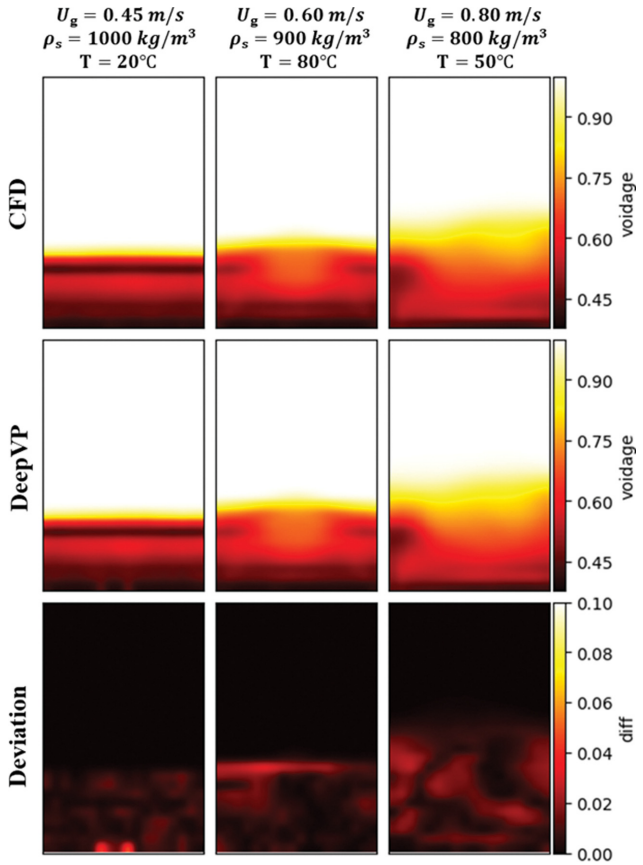

Fig. 4. Predicted voidage images of the DeepVP method within Test-dataset 1 under three different conditions for the bubbling bed and comparison against the CFD simulation results.

work using PyTorch [35]. The input parameters to the PPM method are gas velocity, particle density, and temperature. The generative adversarial network (GAN) [36] was used in the implemented PPM method of this work, where the generator and discriminator are all convolutional neural networks. The network structures of the generator and discriminator for PPM are listed in Table 6.

As shown in Fig. 4, the DeepVP method can obtain a voidage image quite close to that of CFD simulation. The absolute voidage deviations between the DeepVP prediction and the CFD simulation results are acceptable. The three performance metrics, both

Table 7. Comparison of performance metrics of DeepVP and PPM for Test-dataset 1

Method	MAE (%)	PSNR (dB)	Sharpdiff
DeepVP	0.9	35.8	21.1
PPM	1.1	34.2	20.2

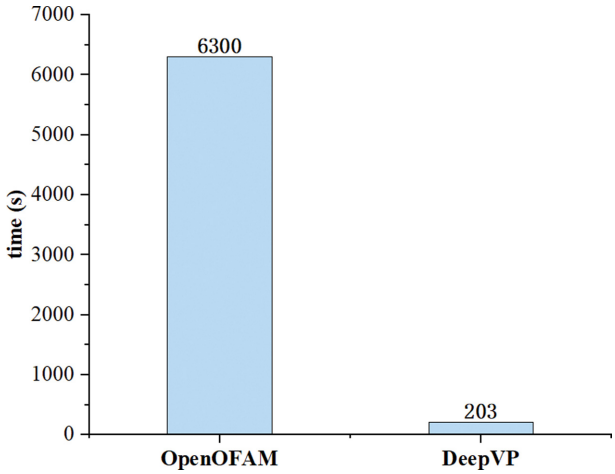

Fig. 5. Acceleration performance of the DeepVP methods against CFD in view of prediction.

for the DeepVP model and the referenced PPM predicted, are listed in Table. The averaged values of the three performance metrics shown in Table 7 indicate that the DeepVP method is better than that of the PPM method, demonstrating that the DeepVP method performs better than the PPM method.

2. Acceleration Performance of the DeepVP Method

The DeepVP method aims at speeding up the process of obtaining a voidage image in the bubbling bed at steady state. Fig. 5 compares the typical computational time (wall time) for voidage prediction required by CFD simulation and the time required by the DeepVP method. The total computational cost for building the DeepVP model is listed in Table 8.

In view of a voidage prediction, the voidage image at steady state can be obtained until 160,800 steps of iteration are finished in the CFD simulation. The CFD simulation needs a total of 1.75 h or 6,300 s. The DeepVP method is much faster when used for a prediction than the CFD simulation. It only takes 203 s, which is 30 times faster. The input simulation data to the deep learning model was prepared with only 5120 CFD iterations that take about 202 s when running the GPU based code of CFD-DEM on two K80 GPUs. It takes only 1 s for the deep learning model to predict the voidage image, of course at a computational cost of 525 hours

Table 8. Time cost for voidage prediction between OpenFOAM and DeepVP

OpenFOAM for a single run	DeepVP		
	Building dataset (once)	Training (once)	Prediction
1.75 h	525 h	55 min	202 s for a simulation as input +1 s prediction

for building the DeepVP model. The DeepVP method achieves a much high acceleration performance in predicting the voidage image of the bubbling bed over CFD simulation.

3. Generalization Capability of the DeepVP

To evaluate the generalization capability of the DeepVP method, the three different test sets of Test-dataset 1, Test-dataset 2, and Test-dataset 3 were used. Accordingly, the total number of samples in the three training data sets are 7,680, 6,400, and 6,400, respectively. The three test sets have different characteristics. The condition parameters of gas velocity, particle density and temperature in Test-dataset 1 all appear in the training set (30,720 samples in total) but in different combinations (see Table 2). However, the value of particle density in Test-dataset 2 for the interpolation test is not present in the training set but within the range of particle density in the training set. While the value of particle density in Test-dataset 3 for the extrapolation test is not present in the training set, which is outside the range of the training set.

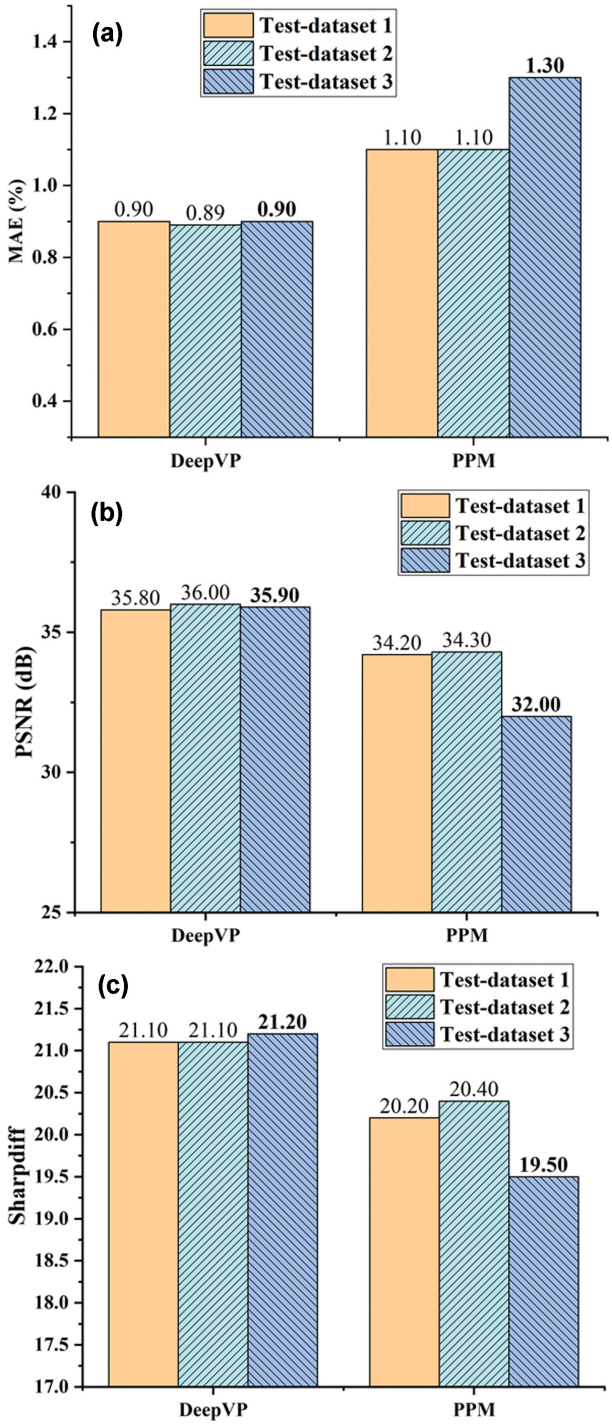

Fig. 6. Comparison of performance metrics between the DeepVP method and the PPM method on the three test sets of Test-dataset 1 for the conventional test, Test-dataset 2 for the interpolation test and Test-dataset 3 for the extrapolation test. (a) MAE, (b) PSNR, (c) Sharpdiff.

Fig. 6 compares the prediction performance metrics of the DeepVP method and the PPM method on the three test sets, in which Test-dataset 1 is for the conventional test, Test-dataset 2 for the interpolation test, and Test-dataset 3 for the extrapolation test. As shown in Fig. 6(a), the MAE of the DeepVP method is less than the PPM method for all three testing datasets. Specifically, the DeepVP method surpasses the PPM method significantly for the extrapolation test set of Test-dataset 3. As shown in Fig. 6(b) and (c), both the PSNR and Sharpdiff metrics indicate the better generalization capability of the DeepVP method over the PPM method too. The validation results exhibit the superiority of the DeepVP method in voidage prediction capability for the bubbling bed at varied operation conditions.

The superiority of the DeepVP method over the PPM method in voidage prediction capability for the bubbling bed is rooted in the differences of the input data for the prediction model. First, the input to the PPM method is a vector of simulated conditions, while the input to the deep learning model of DeepVP is a sequence of images on basis of short-time CFD simulation, which carries not only the simulated conditions, but also the temporal-spatial property of the bubbling bed in operation described by the Navier-Stokes equations. The input simulation condition vector of the PPM method is composed of gas velocity, particle density and temperature only. While the image sequence input of the deep learning model in the DeepVP method derived from CFD simulations not only contains the information of the same simulation condition as the PPM method, but more importantly, the DeepVP method extracts the rich implicit features of the bubbling bed by the convolutional neural network, which is the key to the higher prediction accuracy and better extrapolation capability than the PPM method.

4. Prediction Accuracy Dependency on the Number of CFD Simulations

The applicability of the DeepVP method depends on the model building cost that is largely relying on the number of CFD simulations required in preparing the training and testing dataset. To investigate the effects of the number of CFD simulations on the DeepVP prediction accuracy, the derived training set and test datasets are created for a total of 400 simulations. 50 simulations are randomly selected from these 400 simulations as the derived testing dataset. From the remaining 350 simulations, 30, 40, 60, 80, 120, 180, 240, 300, and 350 simulations are selected to form the derived training set. After data preprocessing, the number of samples in the derived test set is 6400, and the number of samples in the derived training set is 3,840, 5,120, 7,680, 10,240, 15,360, 23,040, 38,400, and 44,800,

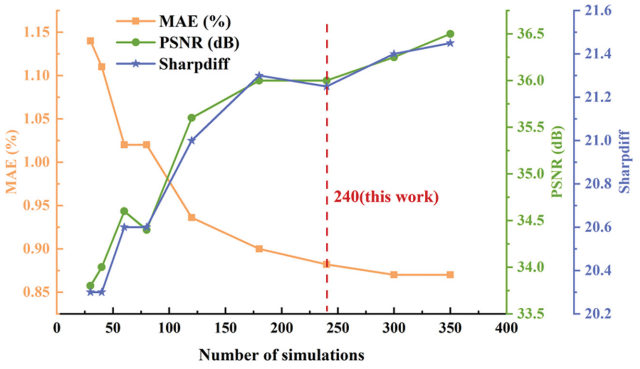

Fig. 7. Averaged performance metrics versus the number of CFD simulations for the training set.

respectively. The deep learning models are rebuilt on these derived training sets and then tested in the derived testing set. The averaged performance metrics of the model are shown in Fig. 7. When the number of simulations is small, increasing the number of simulations has a greater impact on the prediction performance. However, when the number of simulations for the training set reaches more than 240 (this work), the increase in the number of simulations has less impact on the performance metrics. Of course, further increasing the simulation number to 300 slightly improves the prediction accuracy. Considering the practical computational cost, the CFD simulation number of 240 is an acceptable trade-off.

CONCLUSION

This paper proposes a method that combines domain knowledge of CFD simulations for gas-solid fluidized beds with deep learning to predict the voidage image at steady state, called the DeepVP method. Short-term CFD simulations are performed to incorporate domain knowledge into the image sequence. The deep learning model with strong fitting ability was built for a gas-solid fluidized bed. The voidage image at steady state for the bubbling bed can be predicted very efficiently from the image sequence derived from short time CFD simulation. For a bubbling bed, the voidage prediction was accelerated 30 times faster than the CFD simulation alone, while keeping the prediction accuracy within an acceptable range. More importantly, the DeepVP method has better generalization capability than the conventional PPM method.

The application of the DeepVP method to the bubbling bed demonstrates the feasibility and advantages of the DeepVP method for gas-solid fluidized bed over conventional CFD simulation for voidage prediction. From the point of view of CFD simulation, the steady state property distributions of concentration, velocity, and pressure can be predicted as that of voidage. Furthermore, the significantly improved DeepVP voidage prediction speed makes it more practical as a powerful tool in the design and optimization of gas-solid fluidized beds.

ACKNOWLEDGEMENTS

This work was supported by grants from the National Natural Science Foundation of China (62050226), Strategic Priority Research Program of the Chinese Academy of Sciences (XDA21030700) and the Innovation Academy for Green Manufacture, Chinese Academy of Sciences (IAGM-2019-A03).

SUPPORTING INFORMATION

Additional information as noted in the text. This information is available via the Internet at http://www.springer.com/chemistry/journal/11814.

REFERENCES

1. F. Fotovat, X. T. Bi and J. R. Grace, *Chem. Eng. Sci.*, **173**, 303 (2017).
2. J. Sun and Y. Yan, *Meas. Sci. Technol.*, **27**, 112001 (2016).
3. G. Zhu, B. Zhang, P. Zhao, C. Duan, Y. Zhao, Z. Zhang, G. Yan, X. Zhu, W. Ding and Z. Rao, *Fuel*, **252**, 666 (2019).
4. F. Taghipour, N. Ellis and C. Wong, *Chem. Eng. Sci.*, **60**, 6857 (2005).
5. H. Wu, X. Liu, W. An, S. Chen and H. Lyu, *Comput. Fluids*, **198**, 104393 (2020).
6. L. Liang, W. Mao and W. Sun, *J. Biomech.*, **99**, 109544 (2020).
7. W. L. Oberkampf and T. G. Trucano, *Prog. Aerosp. Sci.*, **38**, 209 (2002).
8. O. Obiols-Sales, A. Vishnu, N. Malaya and A. Chandramowliswharan, in *Proceedings of the 34th ACM International Conference on Supercomputing*, 1 (2020).
9. K. Kafui, C. Thornton and M. Adams, *Chem. Eng. Sci.*, **57**, 2395 (2002).
10. M. Marion and R. Temam, *Handbook of Numerical Analysis*, **6**, 503 (1998).
11. Y. Zhao, L. Tang, Z. Luo, C. Liang, H. Xing, W. Wu and C. Duan, *Fuel Process. Technol.*, **91**, 1819 (2010).
12. K. He, X. Zhang, S. Ren and J. Sun, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770 (2016).
13. A. Voulodimos, N. Doulamis, A. Doulamis and E. Protopapadakis, *Comput. Intel. Neurosc.*, **2018**, 7068349 (2018).
14. T. Young, D. Hazarika, S. Poria and E. Cambria, *IEEE Comput. Intell. Mag.*, **13**, 55 (2018).
15. K. Chowdhary, *Fundamentals of Artificial Intelligence*, 603 (2020).
16. S. Choi, I. Jung, H. Kim, J. Na and J. M. Lee, *Korean. J. Chem. Eng.*, **39**, 515 (2022).
17. J. Na, K. Jeon and W. B. Lee, *Chem. Eng. Sci.*, **181**, 68 (2018).
18. H. Kim, M. Park, C. W. Kim and D. Shin, *Comput. Chem. Eng.*, **125**, 476 (2019).
19. J. Li, Q. Li, H. Hao and L. Li, *Process. Saf. Environ. Prot.*, **149**, 711 (2021).
20. A. P. Masoumi, E. Tajalli-Ardekani and A. A. Golneshan, *Sol. Energy*, **207**, 703 (2020).
21. M. Bakhtiari and H. Ghassemi, *Appl. Ocean Res.*, **94**, 101981 (2020).
22. H. Bazai, E. Kargar and M. Mehrabi, *Chem. Eng. Sci.*, **246**, 116886 (2021).
23. J. An, H. Wang, B. Liu, K. H. Luo, F. Qin and G. Q. He, *Int. J. Hydrogen Energy*, **45**, 17992 (2020).
24. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, *Neural. Comput.*, **1**, 541 (1989).
25. T. Salmi, J. Kiljander and D. Pakkala, *Energies*, **13**, 2370 (2020).

26. M. Mathieu, C. Couprie and Y. Lecun, in *ICLR* (2016).
27. S. Aigner and M. Körner, *arXiv preprint arXiv:1810.01325* (2018).
28. H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang and D. N. Metaxas, in *Proceedings of the IEEE International Conference on Computer Vision*, 5907 (2017).
29. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving and M. Isard, in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265 (2016).
30. S. Ruder, *arXiv preprint arXiv:1609.04747* (2016).
31. D. P. Kingma and J. Ba, *arXiv preprint arXiv:1412.6980* (2014).
32. C. M. Boyce, D. J. Holland, S. A. Scott and J. S. Dennis, *Ind. Eng. Chem. Res.*, **52**, 18085 (2013).
33. P. Zhao, J. Xu, W. Ge and J. Wang, *Chem. Eng. J.*, **389**, 124343 (2020).
34. A. Hore and D. Ziou, in *2010 20th International Conference on Pattern Recognition*, 2366 (2010).
35. M. Fey and J. E. Lenssen, *arXiv preprint arXiv:1903.02428* (2019).
36. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, *IEEE Signal Process. Mag.*, **35**, 53 (2018).

Supporting Information

A deep learning approach using temporal-spatial data of computational fluid dynamics for fast property prediction of gas-solid fluidized bed

Pengfei Qin*,, Zhaojie Xia*,**, and Li Guo*,**,***,†**

*State Key Laboratory of Multiphase Complex Systems, Institute of Process Engineering, Chinese Academy of Sciences, Beijing 100190, P. R. China
**School of Chemical Engineering, University of Chinese Academy of Sciences, Beijing 100049, P. R. China
***Innovation Academy for Green Manufacture, Chinese Academy of Sciences, Beijing 100190, P. R. China
(*Received 11 June 2022 • Revised 29 September 2022 • Accepted 6 November 2022*)

S1. Parameters for CFD-DEM simulations [1]

Parameter	Value
Gas phase	
Initial pressure, P (Pa)	10^5
Temperature, T (°C)	0-100
Inlet superficial velocity, u (m/s)	0.4-0.85
Minimum fluidization velocity, u_{mf} (m/s)	0.3
Inlet voidage, ε	0.4
CFD time step, dt (s)	1.25×10^{-4}
Particles	
Particle dynamics time step (s)	1.25×10^{-5}
Diameter of particles, d_p (m)	1.2×10^{-3}
Density of particles, ρ_p (kg/m^3)	800, 900, 1,000
Height of settled bed (m)	0.03
Number of particle, N_p	30,325
Normal spring stiffness, k_n (N/m)	3,000
Tangential spring stiffness, k_t (N/m)	428
Friction coefficient, μ_f	0.1
Restitution coefficient, e_n, e_t	0.95
Geometry	
High of bed, Hbed (m)	0.12
Diameter of bed, dbed (m)	0.044
Cartesian grid length (m)	0.004

S2. Summary of main equations of CFD-DEM method [1]

Equation of motion for every particle:

$$m_a\frac{d^2\mathbf{r}_a}{dt^2}=-V_a\nabla p+\frac{V_a\beta}{1-\varepsilon_g}(\mathbf{u}_g-\mathbf{v}_a)+m_a\mathbf{g}+\mathbf{F}_{contact,\,a}$$

Gas phase mass conservation equation:

$$\frac{\partial(\varepsilon_g\rho_g)}{\partial t}+\nabla\cdot(\varepsilon_g\rho_g\mathbf{u}_g)=0$$

Gas phase momentum conservation equation:

$$\frac{\partial(\varepsilon_g\rho_g\mathbf{u}_g)}{\partial t}+\nabla\cdot(\varepsilon_g\rho_g\mathbf{u}_g\mathbf{u}_g)=-\varepsilon_g\nabla p-\mathbf{S}_p+\nabla(\varepsilon_g\tau_g)+\varepsilon_g\rho_g\mathbf{g}$$

Gas-solid drag force density:

$$\mathbf{S}_p=\frac{1}{V_{cell}}\sum_{a=1}^{N_{part}}\frac{\beta V_a}{1-\varepsilon_g}(\mathbf{u}_g-\mathbf{V}_a)\delta(\mathbf{r}-\mathbf{r}_a)$$

$$\beta=180\frac{\mu_g(1-\varepsilon_g)^2}{d_p^2\varepsilon_g}+18\frac{\mu_g\varepsilon_g^3(1-\varepsilon_g)(1+1.5\sqrt{1-\varepsilon_g})}{d_p^2}+0.31\frac{\mu_g(1-\varepsilon_g)\mathrm{Re}}{\varepsilon_g d_p^2}\frac{\varepsilon_g^{-1}+3\varepsilon_g(1-\varepsilon_g)+8.4\mathrm{Re}^{-0.343}}{1+10^{3(1-\varepsilon_g)}\mathrm{Re}^{-0.5-2(1-\varepsilon_g)}}$$

and

$$\mathrm{Re}=\frac{\varepsilon_g\rho_g d_p|\mathbf{u}_g-\mathbf{v}_a|}{\mu_g}$$

Gas phase density ρ_g from the ideal gas law:

$$\rho_g=\frac{M_g}{RT_g}p$$

Gas phase stress-strain tensor:

$$\tau_g=\mu_g(\nabla\mathbf{u}_g+\nabla\mathbf{u}_g^T)-\frac{2}{3}\mu_g(\nabla\cdot\mathbf{u}_g)\mathbf{I}$$

REFERENCES

1. P. Zhao, J. Xu, W. Ge and J. Wang, *Chem. Eng. J.*, **389**, 124343 (2020).