MFiX Tutorial: DEM Hopper flow initialized with custom distribution of particle sizes

Sathish Sanjeevi and Jeff Dietiker

National Energy Technology Laboratory, USA

September, 2020

1 DEM Tutorial - Initialization of particles using custom distribution in a Hopper

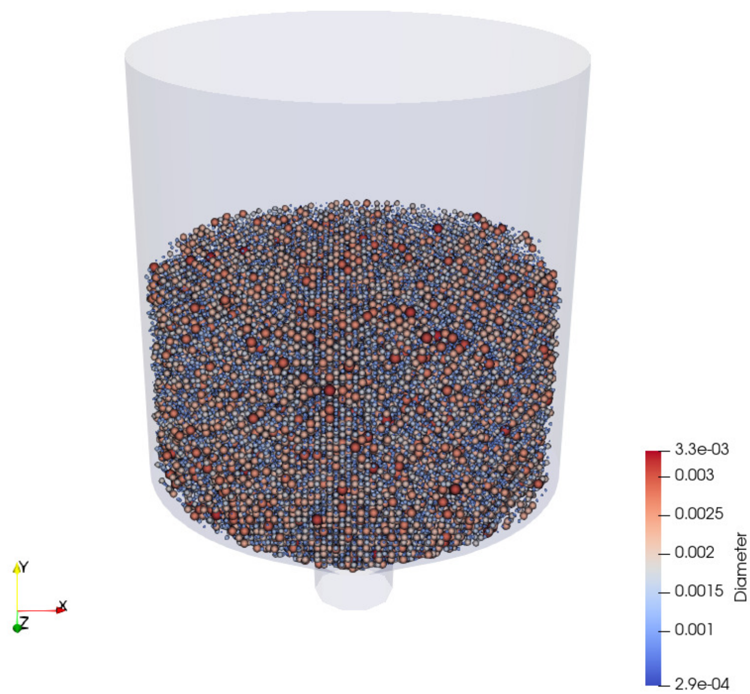


Figure 1: The hopper initialized with custom distribution of polydisperse particles.

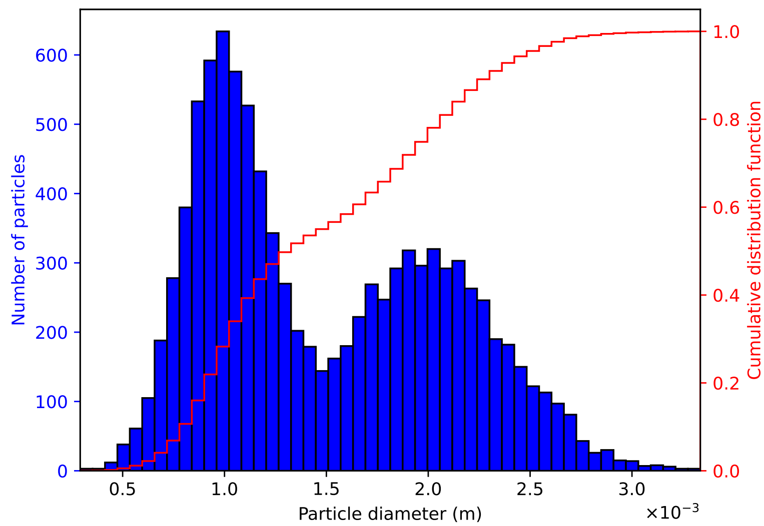


Figure 2: The distribution used for demonstrating the custom distribution feature in MFiX. The histogram in blue represents the number of particles corresponding to the particle diameters, and the red line indicates the corresponding CDF, read from a `txt` file, representing the distribution.

This tutorial performs a Discrete Element Method (DEM) simulation demonstrating a simple initialization of polydisperse particles inside a hopper. The particle sizes are read from a `txt` file (`bimodal_psd.txt`) using `custom` distribution option in the MFiX solver. The corresponding distribution is plotted in figure 2. The input data contains the diameter and corresponding CDF. The resulting initialization of particles is shown in figure 1. Details about the format of this file can be found in the Appendix. This tutorial can be run from the GUI by selecting the `3D Hopper DEM, custom psd` simulation in the list of simulations shown in the `New project` menu.

To use a custom psd file, first create a psd in the `Solids>DEM` pane. Click on the + sign on the psd table, enter a descriptive name, and select the file. Figure 3 shows the panel once the psd file is loaded and plotted. Once defined, the psd can be used in an initial region. Select the psd from the drop-down menu in the `Initial region>Solids` pane (Figure 4).

The corresponding keyword setting in the .mfx files is:

```
ic_psd_type(2,1) = 'CUSTOM'
```

Note that the GUI will automatically copy `bimodal_psd.txt` into `IC_PSD_0002_0001.txt`.

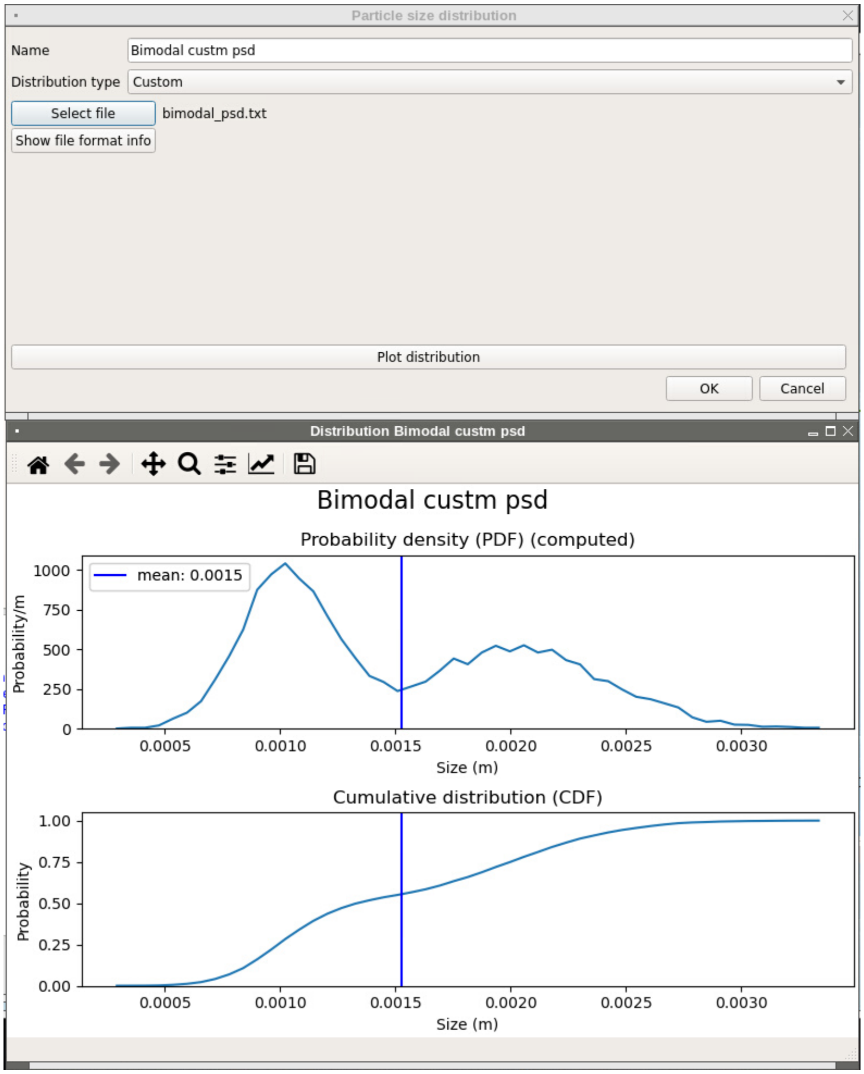


Figure 3: Loading and plotting a custom psd file.

```
51, 1
CDF
! Diameter   CDF
2.9156e-04   0.0000e+00
3.5244e-04   3.0000e-04
4.1332e-04   6.0000e-04
.            .
.            .
3.2139e-03   9.9940e-01
3.2747e-03   9.9970e-01
3.3356e-03   1.0000e+00
```

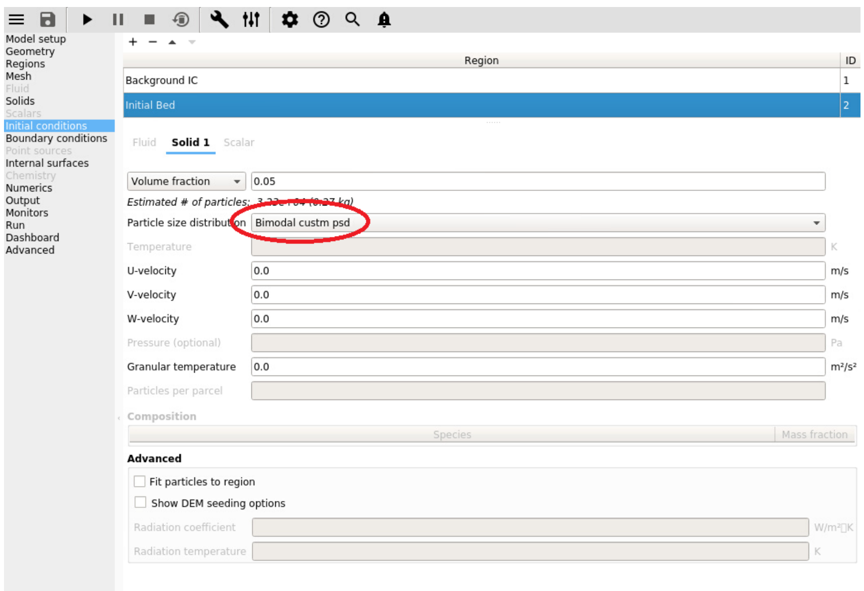


Figure 4: Using a custom psd as initial condition.

2 Appendix - New features in DEM solver

MFiX version 20.3 (September 2020) brings in new features with regards to particles initialization in DEM solver:

- *Polydispersity* - Users can specify a continuous polydisperse distribution of particles within a given solids phase for initial/boundary conditions. Further details are provided in section 2.1. Previous versions of MFiX required manually binning particles into many separate solids phases to achieve a discrete polydisperse system (one diameter per phase). Version 20.3 allows continuous distributions functions (Normal and log-normal). This feature is explained in section 2.1.

- *Lattice style* - For particles initialization, users now have the choice to choose between a hexagonal packing (default setting) and a simple cubic lattice packing (this was the only option in MFiX 20.2 and prior versions). The hexagonal packing can achieve a denser packing limit of 0.74 as opposed to 0.52 for a simple cubic packing. The lattice styles are explained in section 2.2.
- *Lattice spacing* - Scaling factors are now available for particle spacing, either globally in all three directions, or specifically along x, y, or z axis.
- *Spatial randomness* - Apart from adding random velocity components through initialization by using granular temperature, users can now specify random positional shifts in the lattice structure in the initialization step. Both the lattice spacing and spatial randomness feature are explained in section 2.3.
- *Quantity of particles* - Users can now obtain *precise* amount of particles required, for a given phase M, by specifying either the solids volume fraction, number of particles, or total mass of the particles.

2.1 Particle size distribution / Polydispersity

Practical applications involve particles of various sizes and therefore for this purpose, we have introduced the ability to introduce polydisperse spheres [1] in the MFiX solver, either through particle initialization or by mass inflow routines or by both.

. Different type of distributions are possible such as

- normal,
- log-normal,
- custom distribution.

Of the above distributions, normal and log-normal distributions are specified in the `mfx` file using the following parameters

- mean,
- standard deviation,
- minimum diameter, and maximum diameter, in case users would like to clip the distributions with the specified minimum and maximum diameters.

2.1.1 Normal distribution

One of the basic and widely used distributions is a normal distribution. Given a mean μ and standard deviation σ, the probability density function is given by

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]. \tag{1}$$

[1] Current implementation in MFiX assumes particle distribution as based on numbers and not by mass or volume. Users having distributions of particles based on mass/volume are advised to convert them into number based distribution and read it into the solver using `custom` distribution type.

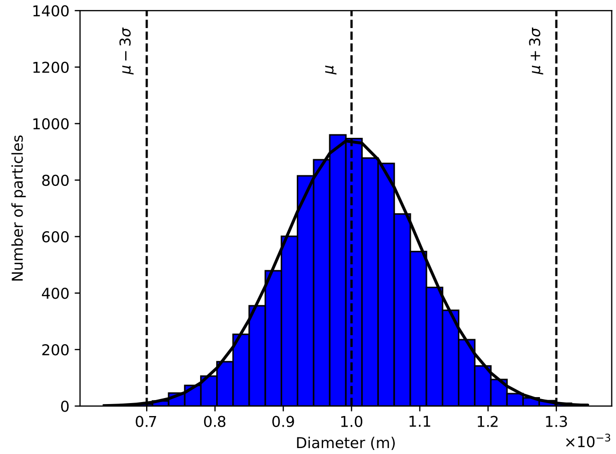


Figure 5: An example normal distribution made of 10,000 particles with mean $\mu = 1 \times 10^{-3}$ m and standard deviation $\sigma = 1 \times 10^{-4}$ m. It should be noted that in the absence of minimum and maximum particle diameters, MFiX solver would automatically set 6σ as extents ($\mu \pm 3\sigma$ as marked in figure) which cover 99.7% number of particles under the distribution.

Interesting feature of the normal distribution is that it follows 68-95-99.7 rule for 2-4-6 sigma respectively. This implies that with 6σ, the distribution can effectively capture 99.7% of the particles falling under the distribution. Therefore, if users do not specify the minimum and maximum diameters, they would be computed based on 6 sigma with $\mu \pm 3\sigma$ as extents. An example normal distribution made of 10,000 particles with mean $\mu = 1 \times 10^{-3}$ m and standard deviation $\sigma = 1 \times 10^{-4}$ m is shown in figure 5.

2.1.2 Log-normal distribution

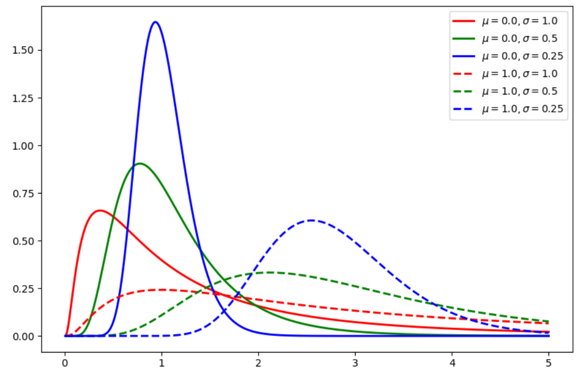
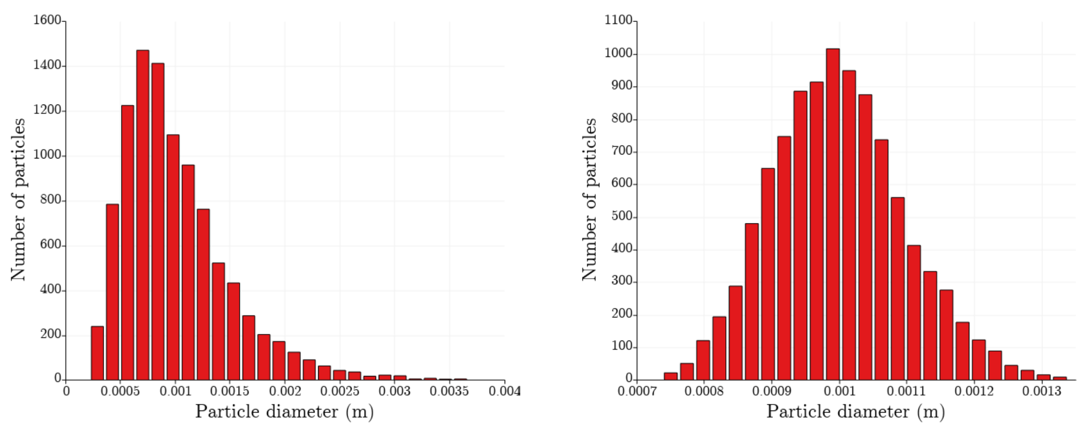


Figure 6: The log-normal distributions for different mean μ and σ.

For distributions with skewness, predominantly to the left, there would be special distributions needed. One such continuous distribution is a log-normal distribution in which the logarithm of a random variable is normally distributed. The probability density function (PDF) for a log-normal distribution is given by

$$P(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right]. \tag{2}$$



(a) $\sigma_x = 5 \times 10^{-4}$ m

(b) $\sigma_x = 1 \times 10^{-4}$ m

Figure 7: The log-normal distribution of 10,000 particles from MFiX simulations with $\mu_x = 1\times10^{-3}$ m and different σ_x. It should be noted that the scales are different for the plots.

The log-normal distributions for different mean μ and standard deviation σ are shown in figure

6. It should be noted here that the log-normal distribution parameters (μ and σ) are not intuitive like the normal distribution discussed earlier. To produce a distribution with mean μ_x and standard deviation σ_x, the following conversion is applied:

$$\mu = \log\left(\frac{\mu_x^2}{\sqrt{\mu_x^2+\sigma_x^2}}\right) \tag{3}$$

$$\sigma = \sqrt{\log\left(1+\frac{\sigma_x^2}{\mu_x^2}\right)} \tag{4}$$

> MFiX users should enter the physical space parameters (μ_x, σ_x), rather than the actual log-normal parameters (μ, σ) as inputs. For example, to obtain a particles size distribution with a mean $\mu_x = 1{\times}10^{-3}$ m and $\sigma_x = 5 \times 10^{-4}$ m , the log-normal parameters are μ = -7.019 and σ = 0.472. The user inputs $\mu_x = 1{\times}10^{-3}$ m and $\sigma_x = 5 \times 10^{-4}$ m, and MFiX will internally convert it to μ = -7.019 and σ = 0.472. .

An example of a log-normal distribution is shown in Figure 7(a)). Reducing the standard deviation will make the distribution more symmetrical (see Figure 7(b)). Similar to the normal distribution, the 68-95-99.7 rule for 2-4-6 sigmas also applies, but as a multiply/divide (99.7% of the particle size falls between $\mu_x \div \sigma_x$ and $\mu_x \times \sigma_x$).

For distributions that cannot be captured with the built-in functions (normal or log-normal), users are advised to use the custom distribution explained below, which can practically introduce any distribution shape to the solver.

2.1.3 Custom distribution

More complex distributions can be created by supplying a custom distribution in a `txt` file . In this section, the basic structure of an input `txt` file for a custom distribution is explained.

```
nrows<=n>, nvar<=1>
<PDF> or <CDF>
!Column header information
 x1   y1<=0>
 .    .
 .    .
 .    .
 .    .
 xn   yn
```

In the first line, `nrows(=n)` indicates the number of data rows in the keyframe file, and `nvar` represents the number of dependent variables which for particle distributions equal 1. In the second line, the user should specify whether the distribution used is either a probability distribution function (PDF) or a cumulative distribution function (CDF). The third line contains the header information. For example, the users can save the variables name in each column.

In the data array, the first column contains the particle diameters x. The second column can either be PDF or a CDF. Following are the conditions that users should remember while specifying the data array.

- The first column (particle diameters) must be sorted in an ascending order. This is because the diameter values are used as the edges of the histogram and non-sorted diameters would represent an erroneous histogram.
- The second column (PDF/CDF) must always start with 0. i.e. $y_1 = 0$. This is to let the solver know that the beginning diameter value (x_1) is the minimum diameter of the distribution.
- In case of a PDF, the y data are normalized inside the solver such that their sum is 1. Therefore, users can even specify absolute number of particles in each bin apart from specifying the actual PDFs.
- In case of a CDF, the last value of y must be equal to 1. i.e. $y_n = 1$. Unlike PDF, CDF data are not normalized and therefore, it is users responsibility to ensure the normalization.

File naming syntax

In a single simulation, there can be multiple custom distributions to be used for different initial conditions (ICs) and boundary conditions (BCs). Similarly, there can be multiple custom distribution with in a single IC/BC ID corresponding to different phases M. Therefore, the naming of the custom distribution `txt` files must be consistent. The naming convention of a custom distribution file has the following convention: `<IC/BC>_PSD_<ID>_<M>.txt`. Here, `IC` or `BC` represents whether the custom distribution corresponds to a IC or a BC, `ID` corresponding to the ID of IC/BC and `M` represents the phase number for the given ID. For example, `IC_PSD_0002_0001.txt` implies a custom distribution file for IC ID 2 and phase number 1. Likewise, `BC_PSD_0005_0003.txt` implies a custom distribution file for BC with ID=5 and for phase number 3.

When setting up a simulation from the GUI, a descriptive name can be used for the custom distribution file depending on the user's choice, for example (`bimodal_psd.txt`). The GUI will copy it to the appropriate file name using the above convention.

2.1.4 Things to remember

- *Diameter ratio* : It should be remembered that every simulation timestep dt is a function of smallest particle in the system. Therefore, having large diameter ratios (d_{max}/d_{min}) imply much slower simulations due to smaller dt. As a rule of thumb, diameter ratio with a maximum of 10 is recommended
- *Packing limits* : Since lattice spacing during initialization is based on the maximum particle diameter, the maximum packing possible effectively reduces with increasing diameter ratios. In other words, smaller particles are occupying spaces allotted for the lattice based on the largest particle. Therefore to achieve tight packing, keep the diameter ratio low.

The polydispersity initialization routines were originally developed by researchers from Arizona State University[2] . The authors of this report have ported the original implementation to the latest MFiX version and implemented the capability of custom distributions and additional lattice/spacing options described in the subsequent sections.

[2] Chen, S., Adepu, M., Emady, H., Jiao, Y. and Gel, A., 2017. Enhancing the physical modeling capability of open-source MFIX-DEM software for handling particle size polydispersity: Implementation and validation. Powder Technology, 317, pp.117-125.

2.2 Initial lattice style

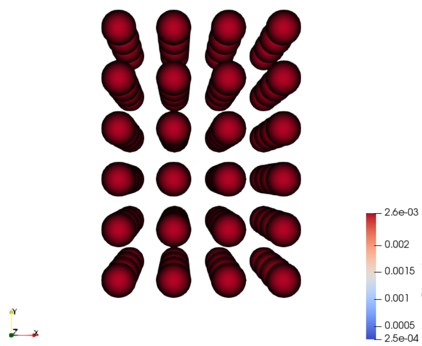
Particle positions are initialized along a lattice structure. Previous versions of MFiX (20.2 and below) used a simple cubic lattice arrangement, which can achieve a theoretical 52% packing limit for monodisperse spheres. The 20.3 version of MFiX provides the option to seed particles along an hexagonal lattice structure, which can achieve 74% of maximum packing. Sample packing of cubic and hexagonal lattice structure along with polydisperse particles are shown in figure 8.

With the availability of polydispersity, the spacing within the lattice is dictated by the largest particle in the given phase. This implies that with increasing (maximum to minimum particle) diameter ratios, the maximum packing possible for a polydisperse system decreases further from the 74% limit. Users should remember that it is simply impossible to achieve high packing limits if the diameter ratios are high.

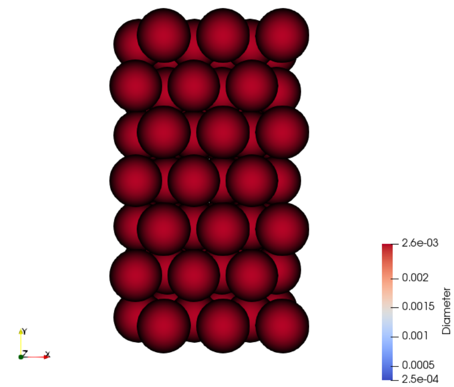
2.3 Lattice spacing and spatial randomness

The lattice spacing feature in MFiX either for specific directions or in all 3 directions is possible now. The different spacing examples are shown in figure 9. By default, a clearance of 5% of the maximum diameter is inserted between particles. This implies that users may not be able to achieve the maximum packing density of 0.52 and 0.74 for simple cubic and hexagonal packing respectively. To achieve highest packing, the users should set the default value of `IC_DES_SPACING` from 0.05 to 0.0. Spacing along specific directions can be controlled using `IC_DES_SPACE_FACTOR_X`, `IC_DES_SPACE_FACTOR_Y`, and `IC_DES_SPACE_FACTOR_Z` respectively for different directions. The above options are displayed in the MFiX-GUI on selecting the `show DEM seeding options` under the `Advanced` pane in the initial condition section.

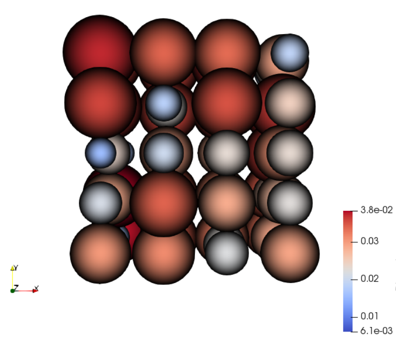
Often an ordered lattice configuration, more specifically simple cubic configuration, imply that the simulation would take longer time to achieve transient behavior, without uniform external forces (such as fluid drag). For example, a perfectly ordered stack of spheres (cubic lattice) falling under gravity and no other forces (say a pure granular flow), would never break the stack. If a user would like to simulate the collapse of such a stack, there should be some asymmetry in the initial condition. Until now, users can specify such asymmetry by initializing the particles with velocity fluctuations by specifying a granular temperature (`IC_THETA_M`). With the new 20.3 MFiX release, users can achieve spatial fluctuations in the lattice by using randomization (`IC_DES_RAND`) option. Likewise, spatial randomness along specific directions can be controlled by using `IC_DES_RAND_FACTOR_X,` `IC_DES_RAND_FACTOR_Y,` and `IC_DES_RAND_FACTOR_Z` respectively. The above randomness options as well are displayed in the MFiX-GUI on selecting the `show DEM seeding options` under the `Advanced` pane in the initial condition section. An example showing the spatial randomness, in all 3 directions, in a initialization of simple cubic configuration is shown in figure 10. It should be noted that a random spacing of 1.0 will nudge particles at most ± half the spacing between particles. This is necessary to avoid overlapping particles. Therefore, if the spacing between particle is zero, the random spacing will have no effect.



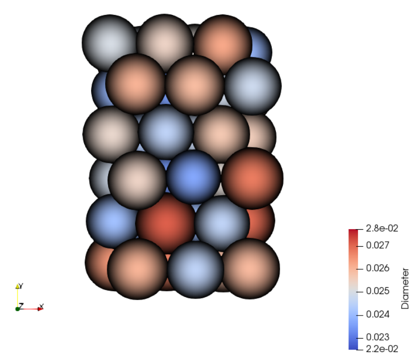
(a) Simple cubic packing for a monodisperse system.



(b) Hexagonal packing for a monodisperse system.

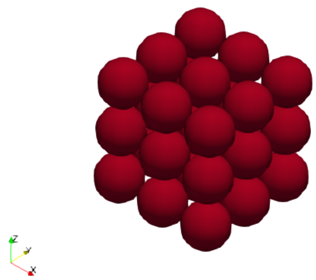


(c) Simple cubic packing for a polydisperse system.

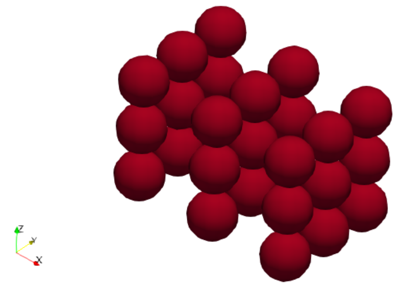


(d) Hexagonal packing for a polydisperse system.

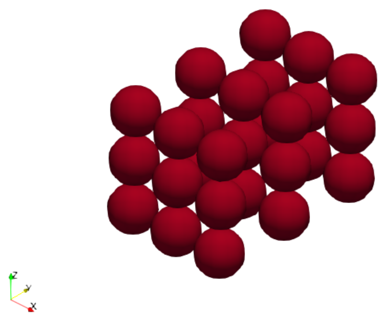
Figure 8: Different types of initialization routines possible now in MFiX. Of the 4 types shown, only (a) is possible earlier in MFiX.



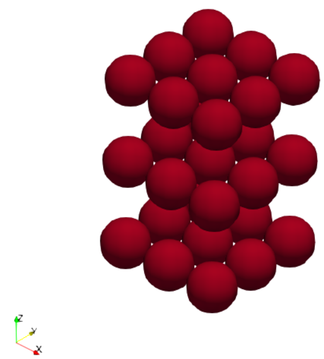
(a) Default packing (with a weak 5% spacing).



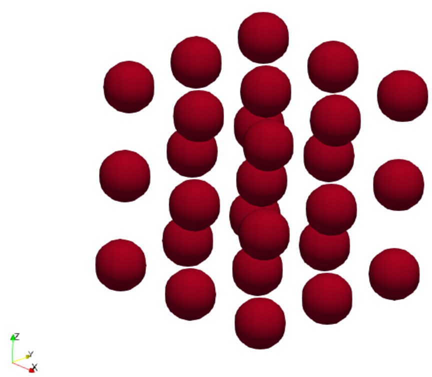
(b) Spacing along x-axis.



(c) Spacing along y-axis.



(d) Spacing along z-axis.



(e) Spacing along all three axes.

Figure 9: The different possible interparticle spacing, shown with a simple cubic lattice configuration of monodisperse particles, along a specific direction or in all three directions.

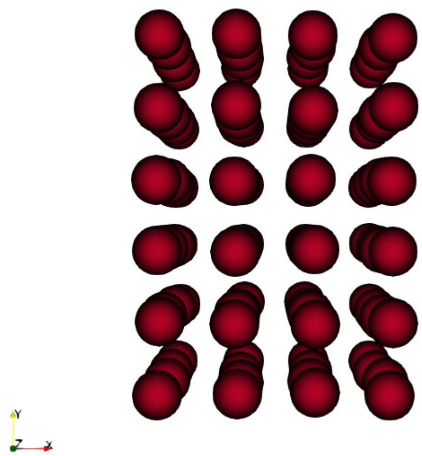


Figure 10: A simple cubic lattice configuration initialized with a spatial randomness.