PCCP

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Realizing high performance gas filters through nanoparticle deposition†

Dhruva Patil,[a] and Tribikram Gupta*[b]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

We have studied separation of hydrogen and methane mixture in equal proportions, using a thin film comprised of 10 layers of nano particles deposited layer-wise using our "two-point sticking algorithm" which simulates controlled agglomeration of such nanoparticles. We simulate the process of gas separation using LAMMPS. We have studied the scenario where nanoparticles act like hard spheres, maintaining their shape and size, similar to what has been demonstrated by experiments involving self-assembled nanoparticle thin films. We consider pressure dependence of the results by working at 3 different initial pressures, 0.1*Po, 0.5*Po and Po, where Po is the atmospheric pressure. Three different diameters of the nanoparticles namely 3 nm, 6 nm and 9 nm are considered, and therefore the overall thickness of the membranes used ranges from 30 nm to 90 nm. We obtained perm-selectivity values that are significantly higher than the Robeson line for hydrogen-methane gas separation indicating the novelty and therefore the significant applications of this work. We find that while the permeance of hydrogen remains more or less steady with a ten-fold increase of pressure, the corresponding fall in methane's permeance is very sharp. The sharpness is more if the size of the nanoparticles is smaller, thereby giving higher selectivity at higher pressure.

1 Particle Deposition Algorithm

The particles are deposited layerwise with periodic boundary condition in the X and Y directions. To deposit the first layer of particles, the Z-coordinate is fixed to R (radius of nanoparticle chosen) and the X and Y coordinates were chosen at random. The coordinate chosen was compared to particles already created. In case a particle in that location would overlap with existing particles, it would be discarded. For a particle to be created, there was another condition besides non-overlapping; if more than 10% of the maximum particles that could be accommodated in that layer were already deposited (for a 10x10 membrane, that would be 10% of 100 nanoparticles sitting uniformly), then the particle would be generated in a given coordinate only if it would be in contact with at least 2 other previously generated particle ("Contact" here is defined as two particles whose centerline distance is between 2*R and 2.1*R, where R is radius). This is to simulate the nanoparticle's tendency to stick together.

In the subsequent layers, the condition that the particles must be in contact with at least 2 particles is retained, with an additional condition that at least 1 of the 2 minimum contacts must be below it ("below" here is defined as two particles whose Z-coordinate differs by 1.5*R). The Z-coordinate is taken as a variable as well, with its value varying over 2*R nanometers starting from previous layer's average Z-coordinate. As the layer gets filled up, a new condition is imposed (after 20% of the maximum particles that could fit in the layer is deposited); for any particle to get created, it must, in addition to the previous conditions, also be in contact with at least one particle in the horizontal plane. This would simulate the nanoparticles aggregating closer together.

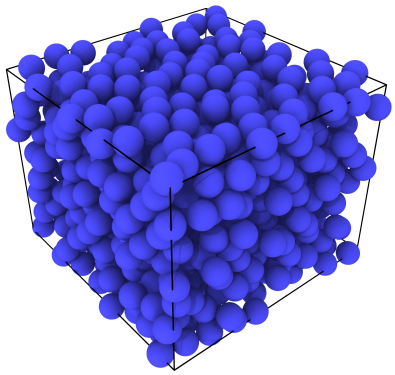
The conditions mentioned above is checked prior to the creation of any particle. If the XYZ coordinate chosen doesn't satisfy the conditions mentioned, that coordinate is discarded and a new one is chosen. This is done at most 1 million times for each particle. If a particle is created within those tries, the counter is reset to 0 and the process starts again for the next particle (1 million tries are given for the new particle). If no particle is created even on the 1 millionth try, then the layer is assumed to be filled, and the algorithm moves to the next layer and starts depositing particles there.

The deposition is shown in Figure 1a. The same is shown with periodic boundary condition in Figure 1b.

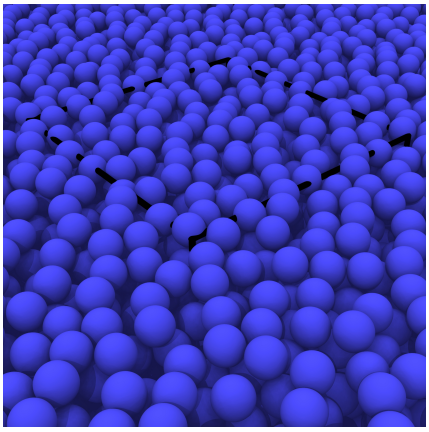
Void Analysis

Once the particles are deposited, we analyse the voids created within the deposition. The next section goes over the methodology used to analyse the voids.

[a] Department of Mechanical Engineering, R. V. College of Engineering, Bangalore, 560059, India
[b] Department of Physics, R. V. College of Engineering, Bangalore, 560059, India. Tel: XX XXXX XXXX; E-mail: tgupta@rvce.edu.in



(a) Isometric view.



(b) View with periodic condition.

Fig. 1 Particles deposited in a 10x10 membrane.

Table 1 Curve fit parameters

Gas Flow Direction	x_0		amp		σ	
	Value	Error (+/- %)	Value	Error (+/- %)	Value	Error (+/- %)
Vertical	1.519017	0.035715	0.087379	0.00456	0.59359	0.03625
Horizontal	1.452188	0.022593	0.108243	0.004588	0.461693	0.022604

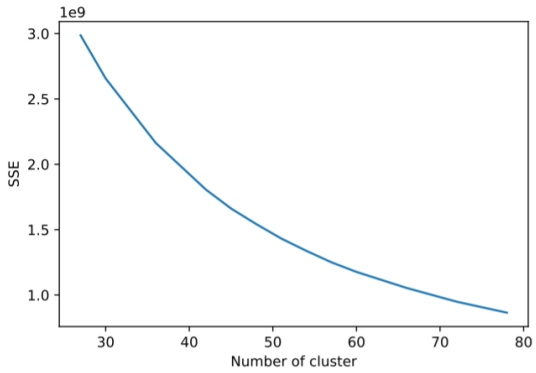


Fig. 2 The knee method of Kmeans.

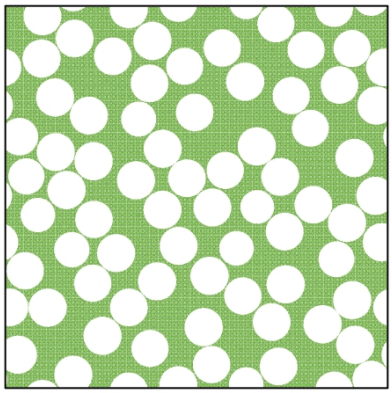


Fig. 3 The top pixel layer where the white region represents the region occupied by particles.

Methodology

1. Twenty horizontal cross sections of the nanoparticle deposition are taken. One each at the average Z coordinate of all particles in a layer, and one in between two concurrent planes.

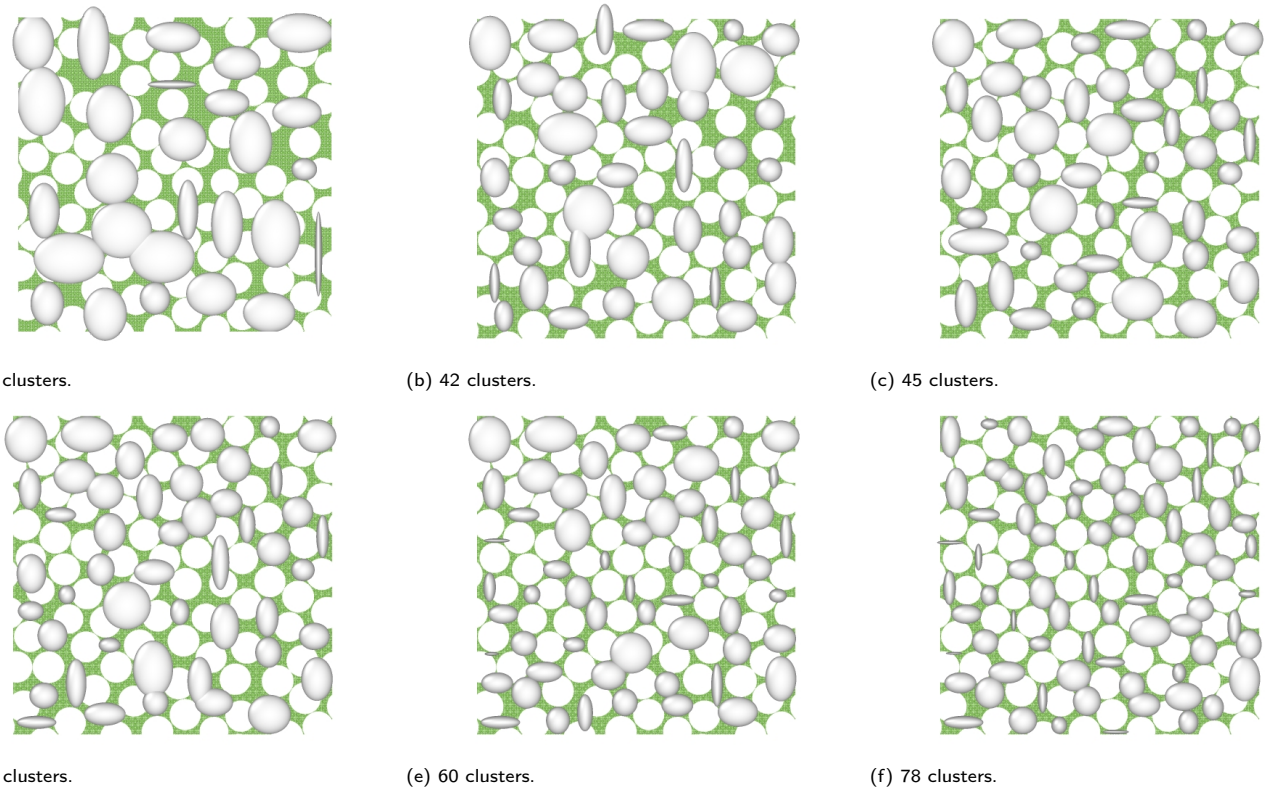
2. The plane is then subdivided into small uniform squares (0.3 Angstroms each side for 3 nm diameter nanoparticle deposition), called pixel. Each pixel is classified as a void or not using Monte Carlo method [1], wherein a number of darts are thrown into each pixel and the location of the dart is used to find if that position is within a particle or in empty space. After throwing a certain number of darts, the pixel is classified as a void space if 50 or greater percentage of the total darts thrown landed in a spot that was a void. If it is void space, that pixel is given a value of 1.

3. Kmeans algorithm is used to cluster these pixels together to form an elliptical approximation of voids. The number of voids (or clusters) that can exist in one layer serves as an input parameter. This was run for many values of the number of clusters. This algorithm has to be run for varying diameter, therefore to get a general value of the most preferred input parameter, a ratio was defined.

$$Cluster\ Ratio = \frac{Clusters}{Diameter} \tag{1}$$

This ratio is varied from 9 till 26 covering the following values (9, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26). The simulation is carried out for deposition of 3 nm diameter particles, therefore the clusters will vary from 27 to 78.

4. The kmeans algorithm aims to minimize the “inertia”, i.e.,



(a) 27 clusters. (b) 42 clusters. (c) 45 clusters.

(d) 48 clusters. (e) 60 clusters. (f) 78 clusters.

Fig. 4 Top layer of pixels for different number of clusters.

the within-cluster sum of squared distance criterion, which is shown in Eq. 2[2]. 300 iterations are performed for each cluster number. The inertia of the model at the end is stored for each cluster number. The Inertia vs clusters graph is plotted and using the knee method, the optimum cluster value is found.

$$\sum_{i=0}^{n} \min_{\mu_j \in C} \left(\|x_i - \mu_j\|^2 \right) \tag{2}$$

5. Once the clusters were created, an ellipse was approximated around the cluster to get the minor and major axis of it

Result

From the Figure 2, we found the optimum cluster size for 3nm dia particles to be 45, i.e., the Cluster Ratio is 15. This was found using the knee method on the inertia vs clusters graph, Figure 3. We can see the gradual decrease in inertia as we increase the number of clusters. Using the inherently approximate Knee method, we picked 45 as the optimum number of clusters. We also used visual means of verifying the results through Ovito.

The Figure 3 shows the top layer of the pixel layers. The white regions are those pixels that are un-activated as there is no void space there.

Figure 4 shows how the top layer looks for each cluster value. The oddly shaped objects are the elliptical approximation made based on the data given by Kmeans regarding the cluster each pixel belongs to. As we can see, if the clusters input is too low, we'll end up with bigger ellipses that very often overlap with a particle. If the input is too high, then the void spaces are broken up into 2 or more ellipses. The ratio of 15, i.e., the input of 45 as the cluster number per layer seems to be the most optimal, so that was chosen to represent the void data in the paper.

Curve Fitting Void Distribution

The total count of pores at different diameters of the pore was extracted to estimate a pore distribution function (PDF). A normal distribution was assumed to fit the curve, shown in Eq. 3.

$$\psi(x) = amp * \frac{1}{\sqrt{2\pi}} * e^{-\frac{(x-x_0)^2}{2\sigma^2}} \tag{3}$$

Table 1 summarizes the curve fit parameters obtained. It is observed that the average pore size is 4.6% smaller for gas flowing horizontally through the substrate. This is expected due to the imposition of additional constraints to ensure the particle always "sit" on top of other particles, in order to simulate the effect of gravity.

Conflicts of interest

There are no conflicts to declare.

Notes and references

1 D. M. Benov, *Monte Carlo Methods and Applications*, 2016, **22**, 73–79.

2 T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2002, 881–892.