Evaluating Sampling Strategies for Visualizing Uncertain Multi-Phase Fluid Simulation Data

Mathias Hummel^{1,a*}, Lisa Jöckel^{1,b}, Jan Schäfer^{2,c}, Mark W. Hlawitschka^{2,d}
and Christoph Garth^{1,e}

¹Scientific Visualization Lab, University of Kaiserslautern, Germany

²Chair of Separation Science and Technology, University of Kaiserslautern, Germany

^am_hummel@cs.uni-kl.de, ^bl_joeckel10@cs.uni-kl.de, ^cjan.schaefer@mv.uni-kl.de,

^dmark.hlawitschka@mv.uni-kl.de, ^egarth@cs.uni-kl.de

Keywords: Method of Moments, Sampling, Flow Visualization.

Abstract. With Eulerian Method of Moment (MoM) solvers for the CFD simulation of multi-phase fluid flow, the positions of bubbles or droplets are not modeled explicitly, but through scalar fields of moments. These fields can be interpreted as probability density functions describing the distribution of bubble locations. To enable intuitive visualization that allows direct visual comparisons between simulation and physical experiment, explicit instances of the bubble distribution are required.

In this work, we examine one sampling-based method for obtaining sets of bubble positions from density fields. Based on an example dataset, we study the influence of the main parameter, the kernel size, on the resulting bubble set. We identify a tradeoff between numerical accuracy and temporal continuity for the visualization.

Introduction

In process engineering, Method of Moment (MoM) solvers are used to simulate scenarios involving liquid and gaseous phases, with the ultimate goal of reducing the need for physical experiments. In such simulations, the exact positions of bubbles are considered uncertain. Bubbles are modeled via scalar density fields describing the stochastic distribution of positions as well as fields of local properties such as the bubble diameter and phase velocities.

The visualization of such datasets is challenging because the simulation output does not lend itself to visual comparison with the physical experiment. This work considers a sampling approach, where an explicit set of bubbles is generated to match the density field at a time step. The bubble set can then be displayed in a way that resembles the physical experiment. To provide temporal coherence in the visualization, the bubble set is updated over time, matching each time step of the density field as closely as possible. This approach requires a method to measure the density of a given set of bubble locations. We describe how this can be achieved using kernel density estimation, where kernel functions are evaluated across the domain. The bandwidth of the kernel functions strongly influences the resulting measured density field, and thereby the updated bubble set. Given a dataset of a bubble column, we study how different choices of this parameter influence the results of our approach.

The paper is structured as follows: We begin by providing a short overview of related work in the field of visualization and some background for Method of Moments simulations. Then, we describe the sampling approach in detail, before discussing the results of applying it to a bubble column dataset. Finally, we summarize our findings and list further research opportunities.

Related Work

Uncertainty Visualization. The visualization of uncertainty and uncertain data is a diverse area of research that has seen many works over the years. Several works focus on the visual quantification of uncertainty, for example using glyphs such as box plots [1], violin plots [2], or summary plots [3].

Potter et al. [4] and Bonneau et al. [5] provide overviews of existing approaches for uncertainty visualization.

Flow Visualization. For the visualization of uncertain flow fields, Lodha et al. [6] used envelopes, ribbons, and glyphs to encode uncertainty of particle traces. Botchen et al. [7] presented a texture-based method using blurred streaklines to convey uncertainty information in two-dimensional flow fields obtained from measurements. A focus-and-context flow visualization method was presented by Bürger et al. [8], who combined spatial importance measures with particle glyphs. Specifically for Method of Moments-based simulations, Hlawitschka et al. [9] and Hummel et al. [10] proposed a visualization approach featuring sampling in addition to parametric bubble glyphs, which were shaped according to local flow properties. We build on their approach by extending the sampling method and evaluating its performance.

Method of Moments Multi-Phase Fluid Simulation. Using multi-phase fluid simulations, the interaction of liquid or gaseous media can be studied. For this purpose, Eulerian-Eulerian methods such as the Method of Moments [11] have long been in the focus of research, with a number of improvements regarding performance and flexibility being proposed [12, 13, 14, 9]. A fundamental aspect common to all these variants is that bubbles are not represented explicitly. Instead, a field of volume fractions describes, for each cell, the ratio of the amount of gas to the amount of liquid. Since the gaseous phase consists of bubbles that are dispersed throughout the liquid, it is referred to as the *dispersed* phase. Accordingly, the liquid phase is called the *continuous* phase.

The bubble population itself is modeled by moments of the local bubble distribution [15]:

$$M_r(\mathbf{x}, t) = \int_0^{\infty} n(L, \mathbf{x}, t) L^r dL. \quad (1)$$

The number density function $n(L, \mathbf{x}, t)$, which describes the number of bubbles with the characteristic length L , is not tracked directly by the simulation. Instead, information such as the local density and bubble size can be derived from the moments. In addition, velocity fields are available for each phase.

For the purpose of this work, the following fields, defined for each time step, are relevant: First and foremost, the zeroth moment $M_0(\mathbf{x})$ describes, for each cell, how many bubbles per volume are present in the area. The third moment $M_3(\mathbf{x})$ is related to the volume of the bubble distribution. It can be used to calculate the size of individual bubbles [14]. Even though this size has no influence on the bubble positions, it is needed afterwards for the visualization [10]. Finally, the vector field $\mathbf{u}_{\text{disp}}(\mathbf{x})$ defines the local velocity of the dispersed phase, and thereby the motion of the bubbles.

Sampling the Bubble Distribution

In this section, we describe the algorithm that is used to obtain a set of bubble locations based on the density field. Starting with an initial sampling, the set of bubbles is updated over time to reflect changes of the density in the simulation output.

The density field $M_0(\mathbf{x})$ defines, for each location $\mathbf{x}$ within the domain, how many bubbles per volume should be present. Since M_0 is constant within each mesh cell, the exact number of bubbles required within each cell i can be calculated:

$$n_{\text{required}}(\text{cell}_i) = M_0(\text{cell}_i) \cdot \text{volume}(\text{cell}_i) \quad (2)$$

This number is generally not integer, which means that a set of bubbles with the correct number of bubbles within each cell cannot be achieved in practice; instead, the number of bubbles should on average satisfy the density requirement.

In a naïve approach, a set of bubble positions could be obtained by rounding and choosing e.g. $\lfloor n_{\text{required}}(\text{cell}_i) \rfloor$ random positions within the cell. However, the overall density of the resulting set of

Fig. 1: Initial sampling and update using rejection sampling; one-dimensional examples. a) In the basic rejection sampling algorithm, samples (x, y) are drawn uniformly from $[x_{\min}, x_{\max}] \times [0, 1]$ and kept if $y \leq af(x)$ (green); otherwise they are rejected (cyan). b) Rejection sampling is performed separately for each cell, minimizing the number of rejected samples. The y -axis in the illustration represents n_{required} (eq. 2). c) An existing bubble set is updated using the measured density (enclosing the red and gray area) and the target density (enclosing the gray and white area). Superfluous bubbles are removed by calculating a new y for each bubble in cells where the measured density exceeds the target density. If the particle falls into the red area, it is removed from the set. d) Finally, missing bubbles are added in areas where the target density is greater than the measured density. The approach is identical to the initial sampling, with the difference that the green acceptance area is defined by the difference between measured and target density.

bubbles would always be smaller than the required density M_0 ; a problem that is exacerbated by small cell sizes or low densities. If, for instance, the density field were to mandate slightly less than one bubble per cell, the naïve algorithm would never produce any bubble locations.

Instead of simply counting the expected number of bubbles per cell, we decide to interpret the density function like the probability density function of a stochastic distribution. A straight-forward method for drawing samples from any given probability density function $f(x)$ is rejection sampling [16, 17]: A factor a is chosen such that $af(x) \leq 1$. Then, two random numbers $x, y \in [0, 1]$ are chosen uniformly. x is kept as a sample if $y \leq af(x)$ and rejected, otherwise. An illustration of this algorithm can be found in Figure 1a).

In the following, we describe how this basic approach can be adapted to bubble density fields. **Initial Sampling.** Starting with an empty set of bubble locations, we first produce an initial sample using a rejection sampling approach. To avoid having to reject too many samples, we treat each cell separately (Fig. 1b)). In contrast to the classic rejection sampling algorithm, which can produce an arbitrary number of samples, a specific density has to be achieved. For cell_{*i*}, we first define the number of attempts as follows:

$$n_{\text{attempts}(\text{cell}_i)} = \lceil n_{\text{required}(\text{cell}_i)} \rceil \quad (3)$$

We then use the following criterion for acceptance:

$$y \leq \frac{n_{\text{required}}(\text{cell}_i)}{n_{\text{attempts}}(\text{cell}_i)} \quad (4)$$

Updating the Bubble Positions. As the time progresses in the simulation, bubbles move through the domain in accordance with buoyancy and the velocity of the surrounding liquid medium. The simulation models this behavior via a separate, time-varying velocity field $\mathbf{u}_{\text{disp}}(\mathbf{x}, t)$ for the dispersed phase. In part due to this motion, a set of bubble positions that reflects the density field at the time t_{i-1} generally does not reflect the density at the next time step t_i . The motion can be accounted for by moving the bubbles in accordance with the dispersed velocity field. For each bubble positioned at $\mathbf{x}(t_{i-1})$, the differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{u}_{\text{disp}}(\mathbf{x}(t), t) \quad t \in [t_{i-1} \dots t_i] \quad (5)$$

is solved using a numerical integration method such as the Dormand-Prince (4)5 scheme [18]. The resulting set of new bubble positions should then match the new density field more closely.

However, aside from motion, there are more causes for change in the density field. Coalescence and separation cause bubbles in a given location to merge or split up. Even more noticeable are changes brought on by in- and outflow — new bubbles introduced into the domain at inlets as well as bubbles leaving the domain through outlets — and of bubbles bursting at the transitional area between the liquid medium and the gas-filled area above the surface of the liquid.

To incorporate these changes, after the bubbles have been moved along their trajectories, the bubble set is updated to reflect the new density field. This is achieved in a two-step approach that is based on the rejection sampling algorithm used for the initial sampling.

Using an estimate of the current density field of the bubble set, it is straight-forward to identify cells with too low or too high density by looking at the difference to the target density field. Similar to equation 2, the measured density field can be converted to a number of bubbles for each cell i :

$$n_{\text{measured}}(\text{cell}_i) = \text{measured_density}(\text{cell}_i) \cdot \text{volume}(\text{cell}_i) \quad (6)$$

In the first step of the update, the rejection sampling algorithm is used to filter the existing bubble set. Instead of drawing new samples $(\mathbf{x}, y)$, the existing bubble positions are used, while each bubble is assigned a random value $y \in [0, 1]$. The bubble is then kept in the set only if the acceptance criterion

$$y \leq \frac{n_{\text{required}}(\text{cell}_i)}{n_{\text{measured}}(\text{cell}_i)} \quad (7)$$

is fulfilled; otherwise, it is removed. This approach is illustrated by Figure 1c).

In the second step (Fig. 1d)), new bubbles are added in cells where the measured density is lower than the target density. Again, the rejection sampling algorithm is applied. For each affected cell, a number of samples according to Equation 3 is drawn. The acceptance criterion for adding these new bubbles to the set is slightly modified to account for the bubbles that are already present:

$$\frac{n_{\text{measured}}(\text{cell}_i)}{n_{\text{attempts}}(\text{cell}_i)} < y \leq \frac{n_{\text{required}}(\text{cell}_i)}{n_{\text{attempts}}(\text{cell}_i)} \quad (8)$$

The resulting, updated set should reflect the new density field.

Measuring the Bubble Density. The quality of the updated bubble set clearly depends on the method used to measure the density field of the bubble set. A naïve approach would be to simply count, for each cell, the number of bubbles that are located within the cell. The problem with such an approach is analogue to the problem discussed in the beginning of this section: For a set of bubbles that was drawn from a density field which calls for only few bubbles in some areas, the number of bubbles contained

Fig. 2: Measuring the density of a given point set, 2D example. Three points are arranged in a mesh with a cell width of 1. A box kernel of width $h = 2$ is used to estimate the density. Integrating the sum of kernels over each cell is straight forward: For each point and cell, the density value is proportional to the ratio of overlapping area and kernel volume. When, in this example, an entire cell is covered by a kernel, the contribution of the corresponding point to the measured density of the cell is 0.25.

within cells in these areas will be either much higher or much lower than the required number n_{required} , even if the overall density in the neighborhood is correct.

This problem is not unique to bubble density fields. In statistics, methods for kernel density estimation are used to estimate the probability density function of a distribution based on a given set of samples [19]. For a set of samples $[x_1 \dots x_n]$, the probability density function $f(x)$ is estimated as follows:

$$f_n(x) = \frac{1}{n} \sum_{i=1}^n w(x - x_i), \quad (9)$$

where w is a kernel with the property $\int_{-\infty}^{\infty} w(u) du = 1$. We can adapt this approach to measure the local density of a set of positions $[x_1 \dots x_n]$:

$$\text{density_measured}(\mathbf{x}) = \sum_{i=1}^n w(\mathbf{x} - \mathbf{x}_i) \quad (10)$$

In this case, the three-dimensional kernel w has to fulfill the requirement

$$\iiint_{\mathbb{R}^3} w(\mathbf{v}) d\mathbf{v} = 1. \quad (11)$$

The density for an entire cell i can then be obtained as follows:

$$\text{density_measured}(\text{cell}_i) = \iiint_{\text{cell}_i} \text{density_measured}(\mathbf{x}) d\mathbf{x} \quad (12)$$

We choose a box kernel of width h

$$w(\mathbf{v}) = \begin{cases} h^{-3} & \text{if } \mathbf{v} \in [-h/2, h/2]^3 \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

because it allows us to calculate the cell-based density value (Eq. 12) in a straight-forward manner. This is illustrated for a two-dimensional example in Figure 2.

Fig. 3: Average integrated squared error with cell volume, and therefore the number of bubbles, artificially inflated by a range of factors.

Fig. 4: Average integrated squared error and average number of changes (bubble removals and additions) for different choices of the kernel width h .

Finally, the width h of the kernel has to be chosen. h is the only parameter of our method for which the optimal choice is not obvious. For the purpose of kernel density estimation, h should be chosen to minimize the integrated mean square error [19]

$$\text{imse}(f_n, f) = \int_{-\infty}^{\infty} E|f_n(x) - f(x)|^2 dx, \quad (14)$$

which is difficult in practice since $f(x)$ is not known. In general, a smaller number of available samples mandates a larger kernel width. A possible approach would be to choose h such that the integrated mean square error is minimal when comparing the required density field to the measured density field of a freshly sampled set of bubble locations:

$$\text{imse}(\text{density_required}, \text{density_measured}) = \sum_i \left(\frac{\text{density_required}(\text{cell}_i) - \text{density_measured}(\text{cell}_i)}{\sum_j \text{density_required}(\text{cell}_j) \cdot \text{volume}(\text{cell}_j)} \right)^2 \text{volume}(\text{cell}_i) \quad (15)$$

On the other hand, in the context of visualization, it may be preferable to minimize the number of changes in the bubble set, i.e. to avoid removing existing bubbles and inserting new bubbles as much as possible.

Results

We evaluate the approach using a three-dimensional multi-phase fluid simulation dataset for a bubble column. The domain is a cuboid 1.5m tall, 18cm wide, and 3cm deep, filled with water to a height of about 1.1m. From an inlet positioned at the bottom, bubbles of air stream into the domain. Figure 6 shows a color-coded two dimensional slice of the density field in a single time step as well as two possible bubble sets for the same time step.

Initial Sampling. We begin by evaluating the initial sampling step. First, an initial set of bubbles is computed for several time steps. This is achieved by removing all bubbles after each time step, providing a clean slate for the next. Then, the average of the integrated mean square error (Eq. 14) is calculated. Since the density field requires only up to about three bubbles per cell, some noise can be

Fig. 5: Measured densities (green) and density differences (blue-red) in a single time step of the bubble column dataset for two choices of the kernel width. The images show a two-dimensional slice through the center of the z-Axis. Red color in the difference fields indicates that the density is too low; blue indicates that the density is too high.

expected in the result. Therefore, a number of samplings are calculated, with the cell volume artificially multiplied by a factor to increase the number of bubbles and therefore the quality of the density measurement. Figure 3 (blue) shows the results for factors between 1 and 2048, using a logarithmic scale for both the integrated mean square error (vertical axis) and the volume factor (horizontal). It can be seen that the integrated mean squared error converges towards zero as the number of bubbles is increased.

Updated Sampling. Next, we repeat the same measurements as for the initial sampling, but without clearing the bubble set between time steps. The bubble set is updated between each time step according to the method described in the previous section. Since ultimately, a very large number of bubbles will be used, the kernel width h is set to zero for the density measurement. Effectively, this means that the density value is determined simply by the number of bubbles within each cell. The results are also shown as orange dots in Figure 3. The error still approaches zero as the number of bubbles is increased. However, the slope of the curve is considerably less steep than in the case of initial sampling, indicating possible opportunities for improving the update algorithm. The practical impact of this discrepancy is not clear, because for initial sampling and updating with the original cell volumes, the average integrated squared error is nearly identical.

Kernel Width. So far, the the trivial kernel width of zero has been used for the measurements. We now study how the kernel width affects the sampling quality when updating a bubble set. First, the update method described in the previous section is used to produce and update sets of bubbles, using a range of different choices for the kernel width h . The density of each bubble set is measured for every time step both before and after updating. Then, for each kernel width, the average integrated squared error of the measured density after updating compared to the required density is calculated. As can be seen in Figure 4 (bottom), the error measure starts out with a value of 306 at $h = 0\text{m}$ and drops slightly until its minimum of 265 at $h = 0.01\text{m}$, which corresponds to a kernel size equal to the cell size. From there on, the error value increases rapidly for larger kernel sizes.

Fig. 6: For the required density field (a), bubble glyphs [9] illustrate bubble sets generated using two different width values (b,c). Black bubbles have been removed in the current step; the color cyan indicates that the bubble has been newly added. The glyph representation allows direct comparison to photographs of physical experiments (d).

Judging by this result, the optimal kernel width for this dataset would be equal to the cell size. However, from a visualization standpoint, there is another quality measure that could be of interest. For the purpose of visualizing the bubble set over time, the bubble set should of course follow the density field as closely as possible, with the additional requirement that bubbles should not be removed or added more often than necessary. Without this additional requirement, it would make sense to simply compute an entirely new set of bubbles at each time step, an approach that, as we have already seen, produces a good fit with the required density field. At the same time, such an approach provides no continuity over time, making it impossible to track individual bubbles visually. Therefore, we consider an additional measure of quality: Minimizing the average number of bubbles that are added or removed in each time step improves the continuity of the bubble set over time. Figure 4 (top) shows the average sum of the number of added bubbles and the number or removed bubbles for each tested kernel width. The minimum of this measure does not coincide with the smallest integrated squared error; instead, the kernel widths between 0.02m and 0.03m (two and three times the cell size, respectively) provide the smallest number of changes. Since the two criteria for the kernel width are in disagreement, the tradeoff between correctness and visual coherence has to be considered.

Figure 5 illustrates the tradeoff between different choices of the kernel width h . The left half shows the results for $h = 0.01\text{m}$, which was shown above to provide the lowest integrated mean square error, while the right half shows results for $h = 0.03\text{m}$. It is clearly visible that for the smaller kernel width, the measured densities are subject to some noise when compared to the target density (Fig. 6). This behavior is expected because of the relatively small spatial density of bubbles in relation to the kernel size. In contrast, the larger kernel width produces smoother measurements which appear blurry in comparison to the target density. For both choices, the approach fulfils its purpose of updating the bubble set. This is most apparent when looking at the area above the inlet: as the existing bubbles have moved up since the previous time step, a gap is clearly visible where more bubbles should stream

in from the inlet. The gap can also be identified by the red color in the pre-update difference plots. After the update, the gap has been filled in. Overall, the smaller kernel width appears to produce a smaller density difference after the update. However, both choices produce very similar results for the final bubble visualization (Fig. 6). In the figure, the bubbles are colored to indicate removals (black) and new additions (cyan). Compared to the larger kernel width on the right, the smaller kernel width results in a larger number of removals, which in turn has to be compensated by a larger number of new additions.

Conclusion and Future Work

We have described a sampling approach to generate bubble positions for the purpose of visualization in moment-based time-varying multi-phase fluid simulation data. Sets of bubble positions are generated in an initial rejection sampling step. To update the set of bubbles over time, kernel density estimation is used to measure the current density field. Discovered discrepancies of measured and required density are then mitigated by adding and removing bubbles appropriately. The approach features a central parameter that has major influence on the resulting bubble set, the kernel width h . To examine the effect of this parameter, and to provide hints for choosing its value, we have studied the results of applying our method to a bubble column simulation dataset. We have shown that a tradeoff exists between the temporal continuity of the visualization, quantified through the number of bubble additions and removals, and the measured correctness of the bubble set.

In the future, we will examine whether more complex kernel functions, such as the Gaussian kernel, can improve the quality of the density measurement. We will also examine why the errors of the update algorithm approach zero more slowly than with initial sampling as the number of bubbles is increased.

Acknowledgements

This research was supported in part by the German Research Foundation (DFG) within the IRTG 2057 "Physical Modeling for Virtual Manufacturing" and by the Center for Mathematical and Computational Modeling (CM²) at the University of Kaiserslautern. We also thank the Allianz für Hochleistungsrechnen Rheinland-Pfalz (AHRP).

References

- [1] J. W. Tukey, *Exploratory data analysis*. Pearson, 1977.
- [2] J. L. Hintze and R. D. Nelson, "Violin plots: a box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998.
- [3] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson, "Visualizing summary statistics and uncertainty," in *Computer Graphics Forum*, vol. 29, pp. 823–832, Wiley Online Library, 2010.
- [4] K. Potter, P. Rosen, and C. R. Johnson, *From Quantification to Visualization: A Taxonomy of Uncertainty Visualization Approaches*, pp. 226–249. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- [5] G.-P. Bonneau, H.-C. Hege, C. R. Johnson, M. M. Oliveira, K. Potter, P. Rheingans, and T. Schultz, *Overview and State-of-the-Art of Uncertainty Visualization*, pp. 3–27. London: Springer London, 2014.
- [6] S. K. Lodha, A. Pang, R. E. Sheehan, and C. M. Wittenbrink, "Uflow: Visualizing uncertainty in fluid flow," in *Visualization '96. Proceedings.*, pp. 249–254, IEEE, 1996.

-
- [7] R. P. Botchen and D. Weiskopf, "Texture-based visualization of uncertainty in flow fields," in *16th IEEE Visualization Conference, VIS 2005, Minneapolis, MN, USA, October 23-28, 2005*, pp. 647–654, 2005.
 - [8] K. Bürger, P. Kondratieva, J. H. Krüger, and R. Westermann, "Importance-driven particle techniques for flow visualization," in *IEEE VGTC Pacific Visualization Symposium 2008, PacificVis 2008, Kyoto, Japan, March 5-7, 2008*, pp. 71–78, 2008.
 - [9] M. W. Hlawitschka, J. Schäfer, M. Hummel, C. Garth, and H.-J. Bart, "Populationsbilanzmodellierung mit einem Mehrphasen-CFD-Code und vergleichende Visualisierung," *Chemie Ingenieur Technik*, vol. 88, no. 10, pp. 1480–1491, 2016.
 - [10] M. Hummel, L. Jöckel, J. Schäfer, M. W. Hlawitschka, and C. Garth, "Visualizing uncertain multi-phase fluid simulation data using a sampling approach," in *EuroVis 2017 Proceedings*, 2017. Accepted.
 - [11] H. Hulburt and S. Katz, "Some problems in particle technology - a statistical mechanical formulation," *Chemical Engineering Science*, vol. 19, no. 8, pp. 555–574, 1964.
 - [12] R. McGraw, "Description of aerosol dynamics by the quadrature method of moments," *Aerosol Science and Technology*, vol. 27, pp. 255–265, AUG 1997.
 - [13] D. Marchisio and R. Fox, "Solution of population balance equations using the direct quadrature method of moments," *Journal of Aerosol Science*, vol. 36, pp. 43–73, Jan 2005.
 - [14] C. Drumm, M. Attarakih, M. W. Hlawitschka, and H.-J. Bart, "One-group reduced population balance model for CFD simulation of a pilot-plant extraction column," *Industrial & Engineering Chemistry Research*, vol. 49, no. 7, pp. 3442–3451, 2010.
 - [15] D. L. Marchisio, J. T. Pikturka, R. O. Fox, R. D. Vigil, and A. A. Barresi, "Quadrature method of moments for population-balance equations," *AIChE Journal*, vol. 49, no. 5, pp. 1266–1276, 2003.
 - [16] J. Von Neumann, "Various techniques used in connection with random digits," *Monte Carlo Method, National Bureau of Standards Series*, vol. 12, pp. 36–38, 1951.
 - [17] J. S. Liu, *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.
 - [18] J. R. Dormand and P. J. Prince, "A family of embedded runge-kutta formulae," *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.
 - [19] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, pp. 832–837, 09 1956.