Reducing Noise for PIC Simulations Using Kernel Density Estimation Algorithm

Wentao Wu\textsuperscript{1} and Hong Qin\textsuperscript{1,2}

\textsuperscript{1) School of Physics, University of Science and Technology of China, Hefei, Anhui 230026, China
\textsuperscript{2) Plasma Physics Laboratory, Princeton University, Princeton, NJ 08543, USA}

Noise is a major concern for Particle-In-Cell (PIC) simulations. We propose a new theoretical and algorithmic framework to evaluate and reduce the noise level for PIC simulations based on the Kernel Density Estimation (KDE) theory, which has been widely adopted in the machine learning and big data science. According to this framework, the error on particle density estimation for PIC simulations can be characterized by the Mean Integrated Square Error (MISE), which consists of two parts, systematic error and noise. A careful analysis shows that in the standard PIC methods noise is the dominate error, and the noise level can be reduced if we select different shape functions that are capable of balancing the systematic error and the noise. To improve, we use the von Mises distribution as the shape function and seek an optimal particle width that minimizes the Mean Integrated Square Error (MISE), represented by a Cross-Validation (CV) function. It is shown that this procedure significantly reduces the noise and the MISE for PIC simulations. A particle-wise width adjustment algorithm and a width update algorithm are also developed to further reduce the MISE. Simulations using the examples of Langmuir wave and Landau Damping demonstrate that relative to the standard PIC methods, the KDE algorithm developed in the present study reduces the noise level on density estimation by 98\%, and gives a much more accurate result on the linear damping rate.

\textsuperscript{a) Corresponding author, hongqin@ustc.edu.cn}
I. INTRODUCTION

Noise is a major concern in particle-in-cell (PIC) simulations\textsuperscript{[1-35]}. This problem stems from the usage of pseudo-particle. The spirit of pseudo-particle is to combine a large group of real particles into a pseudo one in order to reduce the computation cost. However, the reduction of simulation particles boosts the noise in simulation because the noise level is inversely proportional to the square root of the number of particles. An appropriate choice of shape function can effectively reduce the noise. In standard PIC methods, one intuitively lets shape function to be a triangular function with a width of double grid size, which is known as the first order weighting scheme\textsuperscript{[3]}. Although this simple shape function alleviates some noise, it leaves a large room to improve. The kernel density estimation (KDE) algorithm\textsuperscript{[36-39]}, a nonparametric method developed in machine learning and widely applied in big data science\textsuperscript{[40,41]}, computer vision\textsuperscript{[42]}, biology statistics\textsuperscript{[43-45]}, and quantitative finance\textsuperscript{[38]}, provides a path forward.

According to the KDE theory, the probability density is estimated by summing up kernel functions, or the shape function, located at all samples, which are simulation particles in PIC simulations. The key question is to determine the kernel function, and it is answered by two steps. First, the shape of the kernel function is designed. Second, the size of the kernel function is selected. When the shape is chosen to be triangular and size to be double grid size, the KDE algorithm is the same as the standard PIC method. When the shape and size of the kernel function are allowed to vary, the KDE method provides a tool to systematically reduce the noise level of the PIC simulations. In this paper, we initiate an investigation in this direction.

Of course, different noise reduction techniques had been investigated in the past. For example, low-pass filters\textsuperscript{[3]} can be used to filter out high frequency components, which may contain undesirable noise. Another important technique is the delta-f method\textsuperscript{[14,17,21,46-48]}, applicable for simulating system with small perturbations. The KDE algorithm developed in the present study can be viewed as a new attempt in this endeavor using modern techniques exemplified by machine learning and big data science.

We adopt the wrapped normal distribution\textsuperscript{[49,51]} as the shape of kernel function. This function can be closely approximated by Von Mises distribution\textsuperscript{[52]}, which is infinitely differentiable everywhere and satisfies the periodic boundary conditions. In addition, there are
many efficient algorithms\textsuperscript{53,54} to compute this function.

The width of kernel function will be optimized according to the KDE theory\textsuperscript{55–58}. The goal is to minimize the Mean Integrated Square Error (MISE)\textsuperscript{59}, which is the integrated square of the difference between the estimated distribution and the true distribution. The MISE consists of two parts, the bias and variance. The bias error corresponds to the systematic error, and the variance error is the noise for PIC simulations. Thus in the present study, bias error will also be called systematic error and the variance error will also be called noise. The optimized width is the one that minimizes the MISE. This is very different from the standard PIC methods, where one selects a small width, i.e., the double grid size, which gives a small system error (bias), but a large noise (variance). Obviously, the KDE algorithm has an advantage, because it balances the bias and variance. The bias-variance trade-off is a key issue in all types of supervised machine learning. It is interesting to realize that it is also the dominant factor in the selection of particle shape function for PIC algorithms.

Since the true density distribution is always unknown in practice, we cannot calculate the MISE directly. There are two indirect ways to estimate this error\textsuperscript{37,38}. The first is the plugin method, which uses a pivot width to obtain an approximate density for estimating the MISE by a first order expansion. However, the results involve the second derivative of the density function in the denominator. In PIC simulations, density perturbations often contains small, short wavelength structures. The reciprocal of second derivative of density function can be large, which introduces significant numerical error. Therefore, the plugin method is not applicable for our purpose.

Instead, we adopt the second method, i.e., the cross-validation (CV) method\textsuperscript{36–38}. The idea is to estimate the true distribution using a leave-one-out distribution, constructed by summing through all samples expect for the one at the position of evaluation. With this technique, the MISE can be estimated up to a constant, and the optimization can be carried out.

Some distribution may have sharp peaks or narrow valleys. Because judgment of the optimal width is according to an integrated sense, the estimated distribution may “cut the peak and fill the valley”\textsuperscript{37}. To overcome this difficulty, a variate width adjustment is utilized. We shrink the width around high density and enlarge the width in the area of low density. This adjustment will further reduces the MISE, because it lightens the bias when the density is large and alleviates the variance when the density is small.
After determining the shape and the optimal width, we further allow the width to be updated through simulations. For this purpose, we must decide at which steps the width should be resized. The simplest approach is to recalculate the width at every time-step. But it suffers from two critical disadvantages. First, the computational cost is quite large since one has to solve a optimization problem at each time-step. Second, when the distribution is close to a uniform distribution, the optimal width will be infinitely large. Therefore, an more sophisticated update algorithm is needed. We will adjust the kernel size only at time-steps when the density is significantly non-uniform as determined by the Anderson-Darling test and has changed significantly since the previous width update. At all other time-steps, the kernel size is not updated.

We note that the idea of KDE has appeared previously in the smoothed particle hydrodynamics (SPH) method for fluid simulations. In this context, the kernel width is determined by solving simultaneous equations with one exogenous parameter, which is selected according to human experience. The SPH method does not determine an optimal kernel size using a systematic approach based on first principles. Diego and Kai used wavelets as kernels to estimate the density. However, the order of wavelet transformation is selected manually. Since the order implicitly decides the width of wavelet kernels, this estimation does not provide a data-driven width selection mechanism either.

As numerical examples, Langmuir wave and its linear Landau damping are simulated by a PIC method using the KDE algorithm. For the Langmuir wave, we compare the optimal CV function with that at the double grid size to demonstrate the reduction of total error by the KDE algorithm. The density and electric field calculated by the KDE algorithm show a significant reduction in noise level, compared with the standard PIC method. In addition, the relation between the optimal kernel width and the number of simulation particles is investigated. For the Landau damping, simulation results using the common random numbers variance reduction technique show that the linear damping rate is improved significantly when the KDE algorithm is applied.

This paper is organized as follows. Section II introduces the methodology of the KDE algorithm with four subsections on von Mises distribution, width selection, sample-wise adaptive method, and width update algorithm. In Section III., Langmuir wave and its Landau damping are simulated to demonstrate the noise-reduction effect of the KDE algorithm.
II. NOISE REDUCTION USING THE KDE ALGORITHM

A. The von Mises Distribution

Among different kinds of boundary conditions for plasma simulations, periodical boundary condition is frequently used. To apply the KDE algorithm to PIC simulations, we would like to select a kernel that satisfies the periodic boundary condition. The normal distribution function is a commonly used as a kernel in KDE algorithm with infinite smoothness. However, the normal distribution is defined on the entire real line and does not satisfy the periodic boundary condition. We can resolve this seeming conflict by using the following wrapped normal distribution as the kernel,

\[ f_{WN}(\theta; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \sum_{k=-\infty}^{+\infty} \exp \left[ -\frac{(\theta - \mu + 2\pi k)^2}{2\sigma^2} \right]. \] (1)

The wrapped normal distribution is just the sum of all normal distribution shifted by multiples of 2π. To reflect the periodic property of this distribution, the position is denoted as θ instead of x. Obviously, it automatically satisfies the periodical boundary condition.

The parameter \( \mu \) controls the peak position of the distribution, and the parameter \( \sigma \) is no longer the variance. A small \( \sigma \) results in a sharp peak, and the distribution converges to normal distribution when \( \sigma / 2\pi \ll 1 \). A large \( \sigma \) flattens the distribution, and the distribution converge to uniform distribution as \( \sigma \) goes to infinity.

Instead of evaluating the infinite series directly, the wrapped normal distribution can be approximated by the von Mises distribution defined as

\[ f(\theta|\mu, \kappa) = \frac{e^{\kappa \cos(\theta-\mu)}}{2\pi I_0(\kappa)}, \] (2)

where \( I_0 \) is the modified Bessel function of order 0. The parameter \( 1/\kappa \), analogous to the \( \sigma^2 \) in the wrapped normal distribution, measures the concentration. The von Mises function can be efficiently computed using many algorithms, and we will adopt the polynomial fit procedure.

B. The optimized size using CV optimization

According to the KDE theory, given \( n \) independent samples \( \{X_i|i = 1 \ldots n\} \) from identical distribution \( f(x) \), the probability density of the distribution can be estimated according to
the following formula,
\[
\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K \left( \frac{X_i - x}{h} \right).
\] (3)

Here, \( K \) is the kernel function, \( h \) is the bandwidth or width of the kernel. A larger width results in a lower and flatter shape function, and a smaller width gives a higher and narrower shape function. If the kernel function satisfies
\[
\int K(x)dx = 1,
\] (4)
\[
K(x) = K(-x),
\] (5)
\[
\int x^2K(x)dx < \infty,
\] (6)

the estimation is called consistent.

We seek the optimized width that will minimize the error of the density estimator. But before we dive into the error analysis, we must clarify how the error of the density estimator is defined. In standard PIC methods, we focus on the accuracy of the density distribution on a sampling Eulerian grid point, as measured by the bias error, or the systematic error. However, it is not the only error that matters, and the variance error, or the noise, of the estimation is also important. If we choose a larger width, the kernel function will be flatter, and will give a smoother density estimation. But it loses detailed information of the distribution. If we choose a very small kernel width, the resulting density estimation will be spikier, and it reflects more details of the density information. As a price paid, it will introduce more statistical noise in the density estimation, corresponding to a large variance. A good density estimator needs to keep a balance between bias and variance. This is the well-known bias-variance trade-off problem.

Quantitatively, the error of a density estimator is defined as the mean integrated square error (MISE)
\[
MISE = E \left[ \int_0^{2\pi} \left[ f(x) - \hat{f}(x) \right]^2 dx \right].
\] (7)

In this paper, the expectation without subscript \( E \) is taken on random variables \( \{X_i | i = 1 \ldots n\} \), contained in the definition of \( \hat{f}(x) \). The MISE in essence is the sum of squared bias
and variance, because at each $x$ we can decompose the mean square error as

$$E \left\{ [f(x) - \hat{f}(x)]^2 \right\} = E \left\{ [f(x) - E\hat{f}(x) + E\hat{f}(x) - \hat{f}(x)]^2 \right\}$$  \hspace{1cm} (8)

$$= \left[f(x) - E\hat{f}(x)\right]^2 + E \left\{ [\hat{f}(x) - E\hat{f}(x)]^2 \right\}$$  \hspace{1cm} (9)

$$= \text{bias}^2 + \text{variance}. \hspace{1cm} (10)$$

The bias part, by definition, indicates the deviation of the estimated density from the true value, and thus corresponds to a systematic error. The variance part measures the fluctuation of the density estimation, which is known as noise for PIC simulations. The mean integrated square error is the integral of the sum of the variance and the square of the bias over all $x$, which provides a global measure of estimation error.

The key problem now is to find the minimum of MISE. There are several approaches. A common and simple way is to use the plugin method in circular data. For the von Mises kernel defined in Eq. (2), as the sample number goes to infinity, the variance is asymptotically proportional to $\kappa^{1/2}$, and the bias is proportional to $\kappa^{-2}$ asymptotically. We can see that when the size of kernel decreases as $1/\kappa$ decreases, the bias will be reduced and the variance will be enlarged. On the contrary, when the size of kernel increases, the bias will become large and the variance will be controlled. The multiplication of forth power of leading term of variance width the bias is a constant. So there will be an optimized point that minimized the MISE that best balance the variance and bias error. But this plugin method requires knowledge on the second derivative of the true distribution, which is hard to obtain. In addition, for a distribution with small, short-wave-length perturbation, calculation of second derivative induces large numerical error. Instead, in the present study, we will use the cross-validation (CV) method. The definition of MISE involves the true distribution of density, which is always unknown. The CV method provides a new pathway to estimate the MISE without calculating the density function.

The MISE can be expanded as

$$MISE = E \int \left[ \hat{f}(x) - f(x) \right]^2 \, dx$$  \hspace{1cm} (11)

$$= E \int \hat{f}(x)^2 \, dx - 2E \int \hat{f}(x)f(x) \, dx + \int f(x)^2 \, dx. \hspace{1cm} (12)$$

The last term can be viewed as a constant since it depends only on $f(x)$ and thus independent from the kernel width. Given a width, the first term can also be calculated using Eq. (3),
because it only involves the estimated density itself. The second term is more difficult, since
the true density is required. It can expressed as
\[ \int \hat{f}(x)f(x) \, dx = E_X[\hat{f}(X)]. \] (13)
The expectation on the right hand side of Eq. (13) can be estimated by the average of \( \hat{f}(X_i) \),
i.e. \( \sum_{i=1}^{n} \hat{f}(X_i)/n \). However, this estimation leads to \( h = 0 \) when minimizing the MISE.
This fact is proved in Appendix A. To overcome this difficulty, the leave-one-out estimator is applied. The expectation is estimated as
\[ \hat{E}_X[\hat{f}(X)] = \frac{1}{n} \sum_{i=1}^{n} \hat{f}_{-i}(X_i), \] (14)
where
\[ \hat{f}_{-i}(X_i) = \frac{1}{(n-1)h} \sum_{j \neq i, j=1}^{n} K \left( \frac{X_i - X_j}{h} \right) \] (15)
is the leave-one-out estimator of Eq. (13). We mention in passing that there also exists a similar jackknife re-sampling method to achieve the same goal.

The CV function of kernel width \( h \) is defined as
\[ CV(h) = \int \hat{f}(x)^2 \, dx - 2E[\hat{E}_X[\hat{f}(X)]] \] (16)
\[ = \frac{1}{n^2h^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \int K \left( \frac{X_i - x}{h} \right) K \left( \frac{X_j - x}{h} \right) \, dx \]
\[ - \frac{2}{n} \sum_{i=1}^{n} \left[ \frac{1}{(n-1)h} \sum_{j \neq i, j=1}^{n} K \left( \frac{X_i - X_j}{h} \right) \right] \] (17)
\[ = \frac{1}{n^2h} \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{K} \left( \frac{X_i - X_j}{h} \right) \]
\[ - \frac{2}{n(n-1)h} \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} K \left( \frac{X_i - X_j}{h} \right) \] (18)
where \( \tilde{K}(x) = \int K(x-t)K(t) \, dt \) is the two-fold convolution. It is straight to show that \( \tilde{K}(x) \) is a symmetric probability density function as well.

Since it can be proved that
\[ E[\hat{E}_X[\hat{f}(X)]] = E[E_X[\hat{f}(X)]], \] (19)
in which the outer expectation \( E \) is taken on \( X_i \) and inner expectation \( E_X \) is taken on \( X \),
function \( CV(h) + \int f^2(x) \, dx \) is an unbiased estimator for the MISE. To minimize \( E[CV(h)] \)
is to minimize the MISE, since $\int f^2(x)dx$ is a constant independent from $h$. Assuming that $\hat{h}$ that minimizes $CV(h)$ also approximately minimizes $E[CV(h)]$, we take $\hat{h}$ as the optimal width for density estimation. Thus, the CV function can be used as the proxy of the MISE, and we can apply various algorithms, such as Newton gradient method, to minimize the CV function. The KDE algorithm replaces the fixed double grid width in the standard PIC methods with an optimal width that balances the variance error (noise) and the bias error (systematic error).

C. Adaptive width

From the minimization of the CV function, an optimal width is obtained. However, if the density has sharp peaks or narrow valleys, it is likely that the density estimator that minimizes the mean and integrated error will “cut the peaks and fill the valleys”.

To avoid this kind of shortcoming, instead of a constant width for the entire domain, the sample-wise adaptive widths will be used. The adaptive method generates different widths for all sample points, or particles, according to the priori density distribution using the CV-optimal width. It shrinks the widths for particles where the density is relative high and enlarges the widths where the density is relatively low. This procedure can be intuitively understood as follows. When the priori density is high, there are more particles crowded, and the bias can be further reduced with smaller width because the sample number is large, and the bias error is the domain error. On the contrary, in the place where the priori density is low, the sample number is small. Thus the variance error dominates, and it needs more particles to be included to reduce the variance of the estimation. Hence the width should be enlarged to include more particles in estimation. With this additional procedure, the MISE is further reduced.

We will adopt the adaptive method described by Silverman to perform the adjustment. It consists of the following steps.

1. Find a pilot estimate $\tilde{f}(x)$ by minimizing the CV function.

2. Define local width factor $\lambda_i$ by

$$\lambda_i = \left\{ f(X_i)/g \right\}^{-\alpha},$$  \hspace{1cm} (20)
where $g$ is the geometric mean of the $\tilde{f}(x)$, i.e.,

$$
\log(g) = \frac{1}{n} \sum \log \left( \tilde{f}(X_i) \right).
$$

(21)

Here, $\alpha$ is a sensitive factor. Larger $\alpha$ corresponds to more sensitive dependence of the corrected density on the priori density. In general, $\alpha$ is chosen to be 0.5, which keeps the same sensitivity relative to the priori density and to the samples.

3. Use the adjusted width to estimate the density function by summing up all kernels located on the samples,

$$
\hat{f}(t) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_\lambda_i} K \left( \frac{t - X_i}{h_\lambda_i} \right).
$$

(22)

D. Width update strategy

Compared with the KDE algorithm in which the samples are static and only one estimation needs to be performed, density estimation in plasma simulation is a dynamical process. The density of particles has to be estimated in each step. Naively, we may wish to optimize the width of kernels for each step, but this brings two difficulties. First, width optimization is expansive to compute. Performing width optimization at each time-step will slow down the simulation. Second and worse, when the distribution is close to uniform, the optimal width is as large as the positive infinity, and the width should not be optimized in this situation.

Therefore, we have to design an update algorithm to determine on which steps the width should be re-evaluated. The update algorithm developed in the present study utilizes a nonparametric test, the Anderson-Darling test\textsuperscript{60}, to tell whether a given sample is sampled from a uniform distribution or not. The confidence of the test is measured by the p-value. Smaller p-value indicates a more severe deviation from uniform. If we have 99% confidence, i.e., $p \leq 0.01$, on that the distribution is significantly different from a uniform, then we are allowed to perform the optimization. For the steps in which we cannot reject the hypothesis that the distribution is uniform, the width will not be updated. In order to effectively reduce the computational cost, we will use the same width for successive time-steps until the change of distribution has a significant accumulation. To measure the change of the distribution, we use the difference between the p-values of the Anderson-Darling test with respect to uniform
distribution at the two time-steps. In another word, we use the difference of deviation level from the uniform distribution as a measurement for the changes. Although the same p-value may correspond to the different distributions, this situation happens only when the distribution experiences a sudden change. Since particle density varies continuously with time, the variation of the p-value is an adequate indicator for the change of the distribution. In the algorithm implemented, only when the p-value at the a time-step reduces by half, we perform the CV optimization.

To sum up, we re-evaluate the width only when the distribution is different from uniform and is significantly different from that when the width was last optimized. The update algorithm can be summarized as follows.

\[
\begin{align*}
\text{if } & \ pvalue > 2 \times \text{threshold} \text{ then} \\
& pvalue_{th} = 2 \times \text{threshold} \\
\text{else if } & \ pvalue < 0.5 \times pvalue_{th} \text{ then} \\
& \text{calculate } h_{opt}^{target} \text{ by minimizing CV.} \\
& pvalue_{th} = pvalue_{th} \\
\text{end if} \\
& h_{opt} = adjustrate \times h_{opt}^{target} + (1 - adjustrate) \times h_{opt}
\end{align*}
\]

Here, the \textit{adjustrate} variable provides a smooth transition from the old width to the target optimal width. Empirically, the it is chosen to be 0.05.

III. SIMULATIONS OF ELECTROSTATIC WAVES

A. The Langmuir Waves

The Langmuir wave is a rapid electrostatic oscillation in plasma driven by the Coulomb force. We compare the results of simulations of Langmuir wave using the standard PIC method and the KDE algorithm.

We follows the convention of Birdsall\cite{5} for the normalization of physical quantities. The electron plasma oscillation frequency $\omega_{pe}$ is normalized to 1. The 1D spatial domain is $[0, 2\pi]$ with 512 grid points, and time-step $\Delta t$ is chosen to be 0.01. We use $2^{14}$ particles in the simulations. The initial equilibrium density distribution $\rho_0$ is uniform and the perturbation is $\rho_1 = 0.02 \cos(x)$. Temperature is set to zero. We run 10000 time-steps in the simulation.
and save the sampling particles’ positions and velocities at every time-step.

To demonstrate the MISE reduction effect of the KDE algorithm. The CV function on a given sample with different widths is drawn in Fig. 1a. By subtracting $\int \hat{E} \hat{f}(x)^2 dx$ from the first term in Eq. (18), the variance term is recovered up to a constant that is independent from $h$. And similarly the bias squared term, up to a constant, can be contracted by subtracting the second term in Eq. (18) from $\int \hat{E} \hat{f}(x)^2 dx$. The blue curve is the variance, which decreases with width $h$. The orange curve is the bias squared, which increases with width when $h$ is greater than 0.1. Summing up the two curves, we obtained the total CV function (green curve) which drops dramatically at first, keeps a steady plateau for while, and then rises up slowly with increasing width. For the convenience of comparison, these three curves are shifted so that the minimum of the three curves are all zeros. The purple dot is when $h = 2 dx$, the width used in standard PIC methods, and the red dot is the optimal point using the CV optimization. It is evident that at the purple dot, the dominant error is the variance error, which reflects the well-known fact that the noise level in the standard PIC methods is large. More importantly, Fig. 1a suggests that variance error, or the noise, can be significantly reduced with a larger width. The MISE error is minimized at the red dot, which is what is used in our simulations with the KDE algorithm.

According to the KDE algorithm, the optimal width decreases with the particle number. When sample number is large, the bias error dominates. If we add more samples, the bias can be compensated by the additional information carried by the extra samples. To reduce the bias error, the width calculated by the KDE algorithm will decrease. We demonstrate the relation between the width and the particle numbers in Fig. 1b. It is clear that the mean of the width as indicated by the dashed black curve decreases with the number of particles. The 95% confidence interval of t-distribution using 20 ensembles for the optimized width is plotted as the blue band, which shows the same trend.

To further reduce the MISE for the density with large variation, adaptive width for every simulation particle is calculated using the algorithms described in Section II.C. The density and electrical field at the initial time-step are shown in Fig. 2. The black dashed curve in Fig. 2a indicates the theoretical density function, and the blue curve and the orange curve is estimated the density function using the standard PIC method and the KDE algorithm, respectively. The standard PIC method suffers from a large variance error, as evident from the high noise level. On the contrary, the KDE algorithm balances the variance and the bias.
FIG. 1. (a) The decomposition of the cross validation function. The blue curve and the orange curve indicate the variance and the bias squared term, respectively. The green curve is the total cross validation function. The purple dot and the red dot denote the $2 \, dx$ for the standard PIC methods and the optimal width, respectively. (b) The relation between width and number of particles. The blue band is the 95% confidence interval with 20 ensembles and the dashed black curve is the mean.

The width is dynamically determined according to update algorithm described in Sec. II.D. The p-values of Anderson-Darling test is plotted in Fig. 3a. The width is updated at the time-steps in which p-values are less than the threshold and p-values decreases by half.
FIG. 2. The density (a) estimated using the KDE algorithm (orange curve) and the standard PIC method (the blue curve), respectively. The black dashed curve is the theoretical density. The electric fields (b) calculated by the KDE algorithm (orange curve) and the standard PIC method (the blue curve), respectively. The black dashed curve is the theoretical electrical field.

Orange dots indicate the time-steps at which the widths are re-evaluated. In Fig. 3b, the width is plotted as a function of time.

Using the KDE algorithm, the Langmuir oscillation is simulated. The density and electric field at all time-steps are compared with those obtained using the standard PIC method in Fig. 4.

For this specific numerical example, the MISE can be calculated by definition according to Eq. 11, since the theoretical density is known. The MISE for density estimation using the standard PIC method is 0.1260, and that using the KDE algorithm is 0.0023. The MISE is reduced by 98% using the KDE algorithm. The MISE for electric field using the standard PIC method is 0.112 and is 0.101 for KDE algorithm. The MISE reduction is 10%.
FIG. 3. (a) The dynamics of log(p-values) of the Anderson-Darling test. The dots indicate the time-steps at which the optimal width is re-evaluated. (b) The width as a function of time according to the update algorithm.

B. Linear Landau damping rate

When the electrons have a finite temperature, the Langmuir wave will be damped by the wave-particle interaction. This is the Landau damping\footnote{69}. To simulate this phenomena, particles are drawn from an initial distribution which is perturbed in configuration space and Maxwellian with variance $\sigma_v^2$ in velocity space. One drawn sample of initial positions and velocities is call a ensemble. On the same particle ensemble, we carry out simulations using the KDE algorithm and the standard PIC methods respectively. To do a fair comparison, the two methods should start from the same ensemble.

In the simulation, the number of grid point is 256, the simulation time-step is 0.01, and $2^{15}$ particles are used. The initial velocity distribution is $f(v) = 1/\sqrt{2\pi v_{th}^2} \exp[-v^2/(2v_{th}^2)]$, 
FIG. 4. The density fluctuation as a function of time calculated by the standard PIC method (a) and the KDE algorithm (b). The electric field as a function of time calculated by the standard PIC method (c) and the KDE algorithm (d). The plasma is cold

where \( v_{th} \) is the thermal velocity. The thermal velocity is normalized to \( v_{th} = 0.40 \), which determines the spatial normalization. The initial density distribution of electrons is a uniform distribution modulated by perturbation of the form \( \rho_1 = 0.02 \cos(1.0 \, x) \). This simulation runs for 1500 steps.

For one specific ensemble, the density distribution as a function of time using the standard PIC method is plotted in Fig. 5a and result using the KDE algorithm is plotted in Fig. 5b. The electric field using the standard PIC method and the KDE algorithm are plotted in Fig. 5c and Fig. 5d, respectively. Compared with the standard PIC method, the KDE algorithm reduces the noise level on both the density and electrical field. This noise reduction is expected to render more accurate physical results. We now show that KDE algorithm generates a more accurate Landau damping rate.

The Landau damping rate is calculated by the drop of energy of electric field. When the logarithm of electric energy is plotted, the slope of the line pass through the first two peaks after \( t = 0 \) is assigned to be the damping rate. The logarithms of total electric field energy of this ensemble using the standard PIC method and the KDE algorithm are plotted in Fig. 6 with blue and red curves respectively. The energy peaks are marked by orange and purple dots. In this ensemble, theoretical damping rate is -0.096. Using the standard PIC
FIG. 5. The density fluctuation as a function of time calculated by the standard PIC method (a) and the KDE algorithm (b). The electric field calculated by the standard PIC method (c) and the KDE algorithm (d). The normalized thermal velocity is \( v_{th} = 0.40 \).

FIG. 6. The logarithm of electric field energy as a function of time. The result using the standard PIC method (blue curve) is compared to that using the KDE algorithm (red curve).

However, comparison of the two methods on only one ensemble is not enough. To make the comparison more meaningful in statistical sense, the standard PIC method and the KDE method should be compared on the mean damping rate averaged over different en-
sembles corresponding to the same macroscopic state specified by plasma density, plasma temperature and velocity distribution. With the common random numbers variance reduction technique, we draw 30 different ensembles drawn from the same distribution as the same initialization for both methods. The absolute error of the mean damping rate using the standard PIC method is 0.0113 with standard deviation of 0.0032, and the t-statistic is $t_{PIC} = 3.82$. The absolute error of mean using the KDE algorithm is 0.0038 with standard deviation of 0.0034, and the t-static is $t_{KDE} = 1.17$. At 95% confidence level, we cannot reject the null hypothesis that the damping rate obtained using the KDE algorithm is identical to the theoretical one. But the null hypothesis for the standard PIC method can be reject, that is to say, the damping rate from the standard PIC method deviates from theoretical result at 95% confidence level. Since the variance is reduced by the common random numbers technique, the null hypothesis that the error of the standard PIC method is the same as that of the KDE algorithm can be safely rejected at 95% confidence level. Therefore, we can draw the conclusion that the KDE algorithm improves the simulation result of the linear damping rate by reducing the noise level.

Next, we extend the results to different temperatures. We scan the thermal velocity from 0.20 to 0.45 with a 0.05 interval. The results are plotted in Fig. 7a. The blue curve is the theoretical result, the green curve is the result using the KDE algorithm, and the orange is the that using the standard PIC method. It clear that the result by KDE algorithm is closer to the theoretical values. The numerical errors with error bar of 95% confidence interval is plotted in Fig. 7b, from which we can conclude that for each thermal velocity, the mean damping rate calculated by the KDE algorithm is more accurate than that calculated by the standard PIC method.

IV. CONCLUSION

In this paper, we proposed to use the Kernel Density Estimation (KDE) algorithm to reduce the noise for Particle-In-Cell (PIC) simulations. A framework for quantitatively evaluating and minimizing the error of density estimation for PIC simulations is established using the KDE theory. Under this framework, the error on particle density estimation for PIC algorithms, as measured by the Mean Integrated Square Error (MISE), consists of two parts, the bias error and the variance error. The bias error is the systematic error and
FIG. 7. For the different thermal velocity, the simulations are run for 30 ensembles and the main damping rate is evaluated. The result using the KDE algorithm (green curve) is closer to the theoretical damping rate (blue curve) than that using the standard PIC method (orange curve) (a). The estimation error using the KDE algorithm (orange curve) and the standard PIC method (blue curve) are compared in (b). The 95% confidence intervals are plotted as error bar.

the variance error is the familiar noise for PIC simulations. A careful analysis showed that the error on particle density estimation in the standard PIC methods is dominated by the variance error, which is consistent with well-known fact that PIC simulations suffer from large noise. Analysis also suggested that the noise level and total MISE can be significantly reduced by increasing the width of the kernel function, i.e., particle shape function. In the KDE algorithm developed in the present study, an optimal width is obtained by minimizing the MISE, represented by a Cross-Validation (CV) function, which is constructed using the leave-one-out estimation. The algorithm is further improved by adopting a particle-wise width optimization scheme. For the dynamics of the PIC systems, an efficient width update
algorithm is also developed.

To test the KDE algorithm, we carried out simulations study of the Langmuir wave and its Landau damping. Simulation results showed that relative to the standard PIC methods, the KDE algorithm can reduce the noise level on density estimation by 98%, and give a much more accurate result on the linear damping rate.

ACKNOWLEDGMENTS

This research is supported by National Natural Science Foundation of China (NSFC-11775219 and NSFC-11575186), National Key Research and Development Program (2016YFA0400600, 2016YFA0400601 and 2016YFA0400602), and the GeoAlgorithmic Plasma Simulator (GAPS) Project.

REFERENCES

1. J. M. Dawson, Reviews of Modern Physics 55, 403 (1983).
2. R. W. Hockney and J. W. Eastwood, Computer Simulation Using Particles (CRC Press, 1988).
3. C. K. Birdsall and A. B. Langdon, Plasma physics via computer simulation (CRC press, 2004).
4. H. Okuda, Journal of Computational Physics 10, 475 (1972).
5. B. I. Cohen, A. B. Langdon, and A. Friedman, Journal of Computational Physics 46, 15 (1982).
6. A. B. Langdon, B. I. Cohen, and A. Friedman, Journal of Computational Physics 51, 107 (1983).
7. W. W. Lee, Physics of Fluids 26, 556 (1983).
8. B. I. Cohen, A. B. Langdon, D. W. Hewett, and R. J. Procassini, Journal of Computational Physics 81, 151 (1989).
9. P. C. Liewer and V. K. Decyk, Journal of Computational Physics 85, 302 (1989).
10. A. Friedman, S. E. Parker, S. L. Ray, and C. K. Birdsall, Journal of Computational Physics 96, 54 (1991).
11. J. W. Eastwood, Computer Physics Communications 64, 252 (1991).
12. J. Cary and I. Doxas, Journal of Computational Physics 107, 98 (1993).
13. J. Villasenor and O. Buneman, Computer Physics Communications 69, 306 (1992).
14. S. E. Parker, W. W. Lee, and R. A. Santoro, Physical Review Letters 71, 2042 (1993).
15. D. P. Grote, A. Friedman, I. Haber, W. Fawley, and J. L. Vay, Nuclear Instruments and Methods in Physics Research A 415, 428 (1998).
16. V. K. Decyk, Computer Physics Communications 87, 87 (1995).
17. H. Qin, R. C. Davidson, and W. W. Lee, Physical Review Special Topics - Accelerators and Beams 3, 084401 (2000).
18. H. Qin, R. C. Davidson, and W. W. Lee, Physics Letters A 272, 389 (2000).
19. J. Qiang, R. D. Ryne, S. Habib, and V. Decyk, Journal of Computational Physics 163, 434 (2000).
20. R. C. Davidson and H. Qin, “Physics of intense charged particle beams in high energy accelerators,” (Imperial College Press and World Scientific, 2001).
21. Y. Chen and S. E. Parker, Journal of Computational Physics 189, 463 (2003).
22. H. Qin, R. C. Davidson, W. W. Lee, and R. Kolesnikov, Nuclear Instruments and Methods in Physics Research Section A 464, 477 (2001).
23. T. Z. Esirkepov, Computer Physics Communications 135, 144 (2001).
24. J.-L. Vay, P. Colella, P. McCorquodale, B. van Straalen, A. Friedman, and D. P. Grote, Laser and Particle Beams 20, 569 (2002).
25. C. Nieter and J. R. Cary, Journal of Computational Physics 196, 448 (2004).
26. C. Huang, V. K. Decyk, C. Ren, M. Zhou, W. Lu, W. B. Mori, J. H. Cooley, T. M. Antonsen, and T. Katsouleas, Journal of Computational Physics 217, 658 (2006).
27. G. Chen, L. Chacón, and D. C. Barnes, Journal of Computational Physics 230, 7018 (2011).
28. J. Squire, H. Qin, and W. M. Tang, Physics of Plasmas 19, 084501 (2012).
29. J. Xiao, J. Liu, H. Qin, and Z. Yu, Physics of Plasmas 20, 102517 (2013).
30. J. Xiao, J. Liu, H. Qin, Z. Yu, and N. Xiang, Physics of Plasmas 22, 092305 (2015).
31. J. Xiao, H. Qin, J. Liu, Y. He, R. Zhang, and Y. Sun, Physics of Plasmas 22, 112504 (2015).
32. H. Qin, J. Liu, J. Xiao, R. Zhang, Y. He, Y. Wang, Y. Sun, J. W. Burby, L. Ellison, and Y. Zhou, Nuclear Fusion 56, 014001 (2016).
33. Y. He, Y. Sun, H. Qin, and J. Liu, Physics of Plasmas 23, 092108 (2016).
34M. Kraus, K. Kormann, P. J. Morrison, and E. Sonnendrücker, Journal of Plasma Physics 83 (2017).

35J. Xiao, H. Qin, and J. Liu, arXiv, 1804.08823v1 (2018).

36J. Friedman, T. Hastie, and R. Tibshirani, The elements of statistical learning, Vol. 1 (Springer series in statistics New York, 2001).

37B. W. Silverman, Density estimation for statistics and data analysis, Vol. 26 (CRC press, 1986).

38Q. Li and J. S. Racine, Nonparametric econometrics: theory and practice (Princeton University Press, 2007).

39K. Fukunaga and L. Hostetler, IEEE Transactions on information theory 21, 32 (1975).

40J. DiNardo, N. M. Fortin, and T. Lemieux, Labor market institutions and the distribution of wages, 1973-1992: A semiparametric approach, Tech. Rep. (National bureau of economic research, 1995).

41T. Peterka, H. Croubois, N. Li, E. Rangel, and F. Cappello, SIAM Journal on Scientific Computing 38, S646 (2016).

42A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, Proceedings of the IEEE 90, 1151 (2002).

43A. Valouev, D. S. Johnson, A. Sundquist, C. Medina, E. Anton, S. Batzoglou, R. M. Myers, and A. Sidow, Nature methods 5, 829 (2008).

44C. C. Taylor, K. V. Mardia, M. Di Marzio, and A. Panzera, Journal of Applied Statistics 39, 2379 (2012).

45P. Vermeesch, Chemical Geology 312, 190 (2012).

46H. Qin, Physics of Plasmas 10, 2078 (2003).

47H. Qin, E. A. Startsev, and R. C. Davidson, Physical Review Special Topics-Accelerators and Beams 6, 014401 (2003).

48H. Qin, R. C. Davidson, and E. A. Startsev, Physics of Plasmas 15, 063101 (2008).

49E. Gumbel, J. A. Greenwood, and D. Durand, Journal of the American Statistical Association 48, 131 (1953).

50M. Oliveira, R. M. Crujeiras, and A. Rodríguez-Casal, Computational Statistics & Data Analysis 56, 3898 (2012).

51C. C. Taylor, Computational Statistics & Data Analysis 52, 3493 (2008).

52D. Collett and T. Lewis, Australian & New Zealand Journal of Statistics 23, 73 (1981).
53 W. H. Press, Numerical recipes 3rd edition: The art of scientific computing (Cambridge university press, 2007).
54 J. Harrison, in Computer Arithmetic, 2009. ARITH 2009. 19th IEEE Symposium on (IEEE, 2009) pp. 104–113.
55 N. Ushakov and V. Ushakov, Journal of Nonparametric Statistics 24, 419 (2012).
56 N.-B. Heidenreich, A. Schindler, and S. Sperlich, AStA Advances in Statistical Analysis 97, 403 (2013).
57 W. Hardle, J. Marron, and M. Wand, Journal of the Royal Statistical Society. Series B (Methodological) , 223 (1990).
58 S. J. Sheather and M. C. Jones, Journal of the Royal Statistical Society. Series B (Methodological) , 683 (1991).
59 M. Rosenblatt, The Annals of Mathematical Statistics , 832 (1956).
60 T. W. Anderson and D. A. Darling, Journal of the American Statistical Association 49, 765 (1954).
61 N. M. Razali, Y. B. Wah, et al., Journal of statistical modeling and analytics 2, 21 (2011).
62 J. J. Monaghan, Annual review of astronomy and astrophysics 30, 543 (1992).
63 D. J. Price, Journal of Computational Physics 231, 759 (2012).
64 D. del Castillo-Negrete, K. Schneider, M. Farge, G. Chen, et al., Journal of Computational Physics 229, 2821 (2010).
65 A. M. Law, W. D. Kelton, and W. D. Kelton, Simulation modeling and analysis, Vol. 3 (McGraw-Hill New York, 2007).
66 E. Platen and N. Bruti-Liberati, in Numerical Solution of Stochastic Differential Equations with Jumps in Finance (Springer, 2010) pp. 637–695.
67 M. Di Marzio, A. Panzera, and C. C. Taylor, Statistics & Probability Letters 79, 2066 (2009).
68 R. G. Miller, Biometrika 61, 1 (1974).
69 T. H. Stix, Waves in plasmas (Springer Science & Business Media, 1992).
Appendix A: Leave-one-out Necessity

In this appendix, we prove the fact that $h = 0$ minimizes the the KDE estimator defined by Eq. (3). In this case, the CV function is

$$CV(h) = \int \hat{f}(x)^2 dx - 2E[\hat{E}_X[\hat{f}(X)]]$$

$$= \frac{1}{n^2h} \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{K} \left( \frac{X_i - X_j}{h} \right)$$

$$- \frac{2}{n^2h} \sum_{i=1}^{n} \sum_{j=1}^{n} K \left( \frac{X_i - X_j}{h} \right). \quad \text{(A1)}$$

By extracting the terms with $i = j$, the CV function is rearranged into

$$CV(h) = \frac{1}{nh} (\bar{K}(0) - 2K(0))$$

$$+ \frac{1}{n^2h} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \left[ \tilde{K} \left( \frac{X_i - X_j}{h} \right) - 2K \left( \frac{X_i - X_j}{h} \right) \right]. \quad \text{(A2)}$$

Minimizing the MISE is equivalent to minimizing $E[CV(h)]$. We now investigate how the expectation of the CV function depends on the width $h$. First, we look at the expectation of kernel function,

$$E \left[ K \left( \frac{X_i - x}{h} \right) \right] = \int f(\xi)K \left( \frac{\xi - x}{h} \right) d\xi$$

$$= \int f(x + h\eta)K(\eta)h d\eta \quad \text{(A4)}$$

$$= h \int \left[ f(x) + f'(x)h\eta + \frac{1}{2}f''(x)h^2\eta^2 + O(h^3) \right] K(\eta) d\eta \quad \text{(A5)}$$

$$= h \left[ f(x) + \frac{h^2\kappa_2}{2} f''(x) + \frac{h^4\kappa_4}{4!} f^{(4)}(x) + O(h^6) \right], \quad \text{(A6)}$$

where $\kappa_2 = \int \eta^2 K(\eta) d\eta$ and $\kappa_4 = \int \eta^4 K(\eta) d\eta$ are constants depending only on the kernel.

Now we look at each term in the second part of CV function. Since $X_i$ and $X_j$ are independent, the first term is

$$E \left[ \tilde{K} \left( \frac{X_i - X_j}{h} \right) \right] = \frac{1}{h} E \int K \left( \frac{X_i - x}{h} \right) K \left( \frac{X_j - x}{h} \right) dx$$

$$= \frac{1}{h} \int E \left[ K \left( \frac{X_i - x}{h} \right) \right] E \left[ K \left( \frac{X_j - x}{h} \right) \right] dx \quad \text{(A8)}$$

$$= \frac{1}{h} \int E \left[ K \left( \frac{X_i - x}{h} \right) \right]^2 dx \quad \text{(A9)}$$

$$= h \int \left[ f(x) + \frac{h^2\kappa_2}{2} f''(x) + \frac{h^4\kappa_4}{4!} f^{(4)}(x) \right]^2 dx. \quad \text{(A10)}$$
The second term is
\[
E \left[ K \left( \frac{X_i - X_j}{h} \right) \right] = \int f(x)dx \int K \left( \frac{\xi - x}{h} \right) f(\xi) d\xi 
\]
\[
= h \int f(x) \left[ f(x) + \frac{h^2\kappa^2}{2} f''(x) + \frac{h^4\kappa^4}{4!} f^{(4)}(x) \right] dx, 
\]
(A12)

Therefore, the leading terms for the CV function are
\[
CV(f) = \frac{1}{nh} (\bar{K}(0) - 2K(0)) + h^4\frac{\kappa^2(n-1)}{4n} \int [f''(x)]^2 dx - \frac{n-1}{n} \int f(x)^2 dx + O(h^5). 
\]
(A14)

Assuming that the kernel function has peak at \( x = 0 \) such that for all \( x, K(0) \geq \bar{K}(x) \), which is the case for the von Mises distribution, we have
\[
\bar{K}(0) - 2K(0) = \int K(x)^2 dx - 2K(0) 
\]
\[
\leq \int K(0)K(x) dx - 2K(0) 
\]
\[
= K(0) - 2K(0) < 0. 
\]
(A15)

Meanwhile,
\[
\frac{\kappa^2}{4} \int [f''(x)]^2 dx > 0. 
\]
(A18)

It is thus straightforward to show that the the CV function goes to \(-\infty\) when \( h \to 0 \). This proves the fact that \( h = 0 \) minimizes the KDE estimator defined by Eq. (3).