Advances in Water Resources 152 (2021) 103889

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

Nonparametric, data-based kernel interpolation for particle-tracking simulations and kernel density estimation

David A. Benson [a,*], Diogo Bolster [b], Stephen Pankavich [c], Michael J. Schmidt [d]

[a] Hydrologic Science and Engineering, Colorado School of Mines, Golden, CO 80401, USA
[b] Department of Civil and Environmental Engineering and Earth Sciences, University of Notre Dame, Notre Dame, IN 46556, USA
[c] Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO, 80401, USA
[d] Center for Computing Research, Sandia National Laboratories, Albuquerque, NM 87185, USA

ARTICLE INFO

Keywords:
Particle methods
Kernel density estimation
Machine learning
Nonparametric kernel

ABSTRACT

Traditional interpolation techniques for particle tracking include binning and convolutional formulas that use pre-determined (i.e., closed-form, parameteric) kernels. In many instances, the particles are introduced as point sources in time and space, so the cloud of particles (either in space or time) is a discrete representation of the Green's function of an underlying PDE. As such, each particle is a sample from the Green's function; therefore, each particle should be distributed according to the Green's function. In short, the kernel of a convolutional interpolation of the particle sample "cloud" should be a replica of the cloud itself. This idea gives rise to an iterative method by which the form of the kernel may be discerned in the process of interpolating the Green's function. When the Green's function is a density, this method is broadly applicable to interpolating a kernel density estimate based on random data drawn from a single distribution. We formulate and construct the algorithm and demonstrate its ability to perform kernel density estimation of skewed and/or heavy-tailed data including breakthrough curves.

1. Introduction

In many applications, discrete samples of a continuous, and potentially complex, random process are generated as output, even though a continuous solution is desired. Some examples are given by particle-tracking of passive solute transport (e.g., Fernàndez-Garcia and Sanchez-Vila, 2011; Pedretti and Fernàndez-Garcia, 2013; Siirila-Woodburn et al., 2015; Carrel et al., 2018), reactive particle transport (e.g., Ding et al., 2012; 2017; Schmidt et al., 2017; Sole-Mari et al., 2017; Sole-Mari et al., 2019; Sole-Mari and Fernàndez-Garcia, 2018; Benson et al., 2019; Perez et al., 2019; Engdahl et al., 2017; 2019), and Monte Carlo and Bayesian simulation (e.g., Taverniers et al., 2020). In short, many of the quantities used by hydrologists are probability density functions that are constructed by users, even thought there is no concrete and accepted methodology for their construction. A long history of statistical estimation has sought to best-fit some continuous density function to a sequence of random samples, including maximum likelihood estimation (Brockwell and Davis, 2016) and kernel density estimation (Silverman, 1986). The former assumes a functional density form and estimates its parameters, while the latter fits a continuous function to discrete data. Tests of functional fits or other statistical properties may be conducted later. In hydrology (and many other sciences), the underlying processes being simulated may be sufficiently uncertain that a functional form for the density function cannot be assumed, and kernel density estimation is preferred.

Given a true underlying pdf $f(x)$, kernel density estimation is based on the convolution-like interpolation (or extrapolation) of discrete random data $\{x_1, x_2, \ldots, x_n\}$ with some kernel function $K(x)$ producing the estimated pdf

$$\overline{f}(x) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{j=1}^{n} \frac{w_j}{h_j} K\left(\frac{x - X_j}{h_j}\right), \tag{1}$$

where w_j are weights associated with data points X_j (which could be prior "concentrations" that come from binning), h_j are "bandwidths" associated with the kernel applied at each data point, and K is some pre-determined, non-negative function with the requirement $\int K(x)dx = 1$ (i.e., K is a pdf). For random samples, the weights are equal constants that cancel from expression Eq. (1), resulting in a factor of $1/n$. The common forms of K are relatively simple (e.g., triangles or standard Gaussians) and yield estimates of $\overline{f}(x)$ with different properties such as regularity (i.e., number of derivatives) or compact support. Kernels that are symmetric around $x = 0$ are most commonly used (but certainly not always, see Hirukawa, 2018), inasmuch as the eventual form of $\overline{f}(x)$, including skewness or heavy tails, are unknown *a priori*.

* Corresponding author.
E-mail address: dbenson@mines.edu (D.A. Benson).

https://doi.org/10.1016/j.advwatres.2021.103889
Received 9 October 2020; Received in revised form 9 February 2021; Accepted 8 March 2021
Available online 12 March 2021

It is well known that a pre-chosen kernel (such as a standard Gaussian) does not perform well if all of the bandwidths are chosen to be the same size Silverman (1986). Where data is more dense, the kernel bandwidths must be made smaller. This has led to "adaptive bandwidths" that are adjusted based on the apparent or estimated density at the data points. Higher estimated density values at data points are given smaller bandwidths. But one may ask, should the functional form of the kernel also be adjusted based on the estimated density? We suggest (and provide evidence in Appendix C) that the optimal kernel should be the same shape as the underlying true density, which is best estimated by the interpolated density. But clearly, the estimated density changes if the kernel shape changes, therefore an iterative procedure is required. We define this procedure in Section 3 after a brief review of kernel density estimation in Section 2. A series of examples are given in Sections 4 and 5, and we conclude in Section 6.

2. Classical bandwidth selection

Intuitively, one would like to choose a bandwidth as small as possible, because the convolution adds the variance of the kernels to the data itself. On the other hand, as $h \to 0$, the kernels become delta functions and continuity of $\overline{f}(x)$ disappears. Additionally, the choice of h_i will depend strongly on both the eventual shape of $f(x)$ and the availability of random samples in any interval $[x, x+\Delta x]$. This has led to expressions that balance the bias and variance of the estimates (Silverman, 1986) that we review here and re-derive in Appendix A. A common place to start is to minimize the mean integrated squared error (MISE) between the estimated and unknown, real densities given by

$$\text{MISE} = \mathbb{E}\left[\int (f(x) - \overline{f}(x))^2 dx\right]. \tag{2}$$

Taking the expectation inside the integral and realizing that the mean squared error of an estimate is composed of squared bias and variance terms, one finds

$$\mathbb{E}[(f - \overline{f})^2] = (\mathbb{E}[\overline{f}] - f)^2 + \mathbb{E}[(\overline{f} - \mathbb{E}(\overline{f}))^2],$$

which gives a target functional for minimization. Typically, a truncated Taylor series is used to derive asymptotic ($h \to 0, nh \to \infty$) expressions for the bias and variance that depend on the properties of the kernel and underlying density (Silverman, 1986). This process (Appendix A) results in approximations for the bias

$$B(x) = \text{bias}[\overline{f}(x)] = \mathbb{E}[\overline{f}(x)] - f(x) = \frac{h^2}{2} f''(x)\mu_2(K) + \mathcal{O}(h^3), \tag{3}$$

and variance

$$\text{Var}[\overline{f}(x)] = \mathbb{E}[(\overline{f} - \mathbb{E}(\overline{f}))^2] = \frac{1}{nh} f(x) \int K^2(x)dx + \mathcal{O}\big((nh)^{-2}\big). \tag{4}$$

All other things held equal, letting $h \to 0$ minimizes bias, but variance grows without bound (i.e. accuracy increases but smoothness decreases), while letting h grow large decreases the variance of estimates, but accuracy is sacrificed. Minimizing the sum gives a value for the optimal global bandwidth

$$h_0 = \left(\frac{d \int K^2(x)\mathrm{d}x}{n(\mu_2(K))^2 \int (f''(x))^2 \mathrm{d}x}\right)^{1/(d+4)} \tag{5}$$

where d is the number of dimensions of the random variable ($d = 1$ herein). Notice that a finite second moment $\mu_2(K)$ is necessary to use this method in the estimation of the optimal bandwidth; we remove that requirement herein (Appendices A–C). Without any information at all, it is common to assume Gaussian $f(x)$ and Gaussian kernels, in which case a constant global bandwidth is used with size

$$h_0 \approx 1.06 n^{-1/5} \hat{\sigma}, \tag{6}$$

where $\hat{\sigma}$ is the sample variance. Greater data density means smaller bandwidth (until as $n \to \infty$, $h_0 \to 0$). Because this estimation of finite h_0 is based, in part, on an assumption of $h_0 \to 0$, we might expect significant error in any estimate of the global bandwidth value using Eq. (5). Indeed, an exact value of h_0 can instead be derived using the Fourier transform (Appendix B), and we show that the result in Eq. (6) can be significantly erroneous.

Furthermore, it is largely recognized (e.g., Silverman, 1986) that the *local* data density is a better indicator of bandwidths that should be uniquely defined at each data point. In regions where data density is smaller, the bandwidth should be greater. There are several methods used to estimate local data density (e.g., Silverman, 1986; Wu et al., 2007; Sole-Mari and Fernàndez-Garcia, 2018). For example, Silverman (1986) shows that for large n, the local data density can be approximated by the value of the estimated pdf, so that an adaptive bandwidth can be estimated by

$$h_i = h_0 \lambda_i = h_0 \left(\frac{\widetilde{f}(X_i)}{G}\right)^{-\xi}, \tag{7}$$

where the tilde indicates some intermediate estimate of the density, and the normalization factor G is the geometric mean of estimated density values, namely

$$G = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \ln \widetilde{f}(X_i)\right). \tag{8}$$

The exponent $0 \le \xi \le 1$ is an empirical weighting factor shown to be 0.5 under ideal conditions (Abramson, 1982).

In a novel way, Pedretti and Fernàndez-Garcia (2013) investigated the use of the adaptive kernel methods (Eqs. (5), (7), and (8)) for interpolating breakthrough curves (BTCs) for simulated push-pull, single-well tests with trapping in relatively immobile (low-velocity) zones. These BTCs are noteworthy for their thin early tails and fat late tails, or rapid (steep) early breakthrough and delayed, power-law decline of concentration. Importantly, Pedretti and Fernàndez-Garcia (2013) found that adjusting the bandwidth based only on particle density tended to overly broaden the early BTCs in order to more properly represent the late tail. Pedretti and Fernàndez-Garcia (2013) then imposed a restriction on broadening the kernel bandwidth based on whether particles (concentrations) in Eq. (1) occurred early or late in the BTC. This, of course, means that the user must decide how the bandwidths must be adjusted. But this is simply a side effect of choosing, *a priori*, a non-physical and symmetric kernel. If each particle were treated as a single realization of the Green's function, then its highly skewed kernel would transfer little mass to earlier portions of the BTC, and no adjustment may be needed. We investigate that possibility here.

3. Iterative algorithm

We show (Appendix C) that asymptotically as $n \to \infty$, to minimize the MISE, the kernel applied to each random sample should be a scaled version of the underlying true density itself. This suggests that for a reasonably large number of data n, the kernel K should be made functionally similar to the estimated density $\overline{f}$, as this is the best representation of the true density f. Of course the shape of the density is not known *a priori*, so the shape of the kernel must be learned during the estimation process. We seek to find $\overline{f}$ to best approximate f, and we find $\overline{f}$ through successive intermediate estimates that we call $\widetilde{f}$. Our proposed algorithm discovers the kernel shape and size recursively according to the following steps:

1. Build an initial candidate $\widetilde{f}_0(x)$ using constant bandwidth h_0 and standard Gaussian kernel $K(x) = (2\pi)^{-1/2}\exp(-x^2/2)$ in Eq. (1).
2. Use $\widetilde{f}_0(x)$ to interpolate values at data points $\widetilde{f}(X_i)$.
3. Use the values $\widetilde{f}(X_i)$ in Eq. (7) to estimate adaptive bandwidths h_i for the Gaussian kernel and re-estimate $\widetilde{f}_1(x)$. This would end classical estimation. Set counter $\ell = 1$.
4. Use $\widetilde{f}_\ell(x)$ as the new kernel $K_\ell(x) = \widetilde{f}_\ell(x)$.
5. Adjust kernel K_ℓ to have zero mean and unit "width".

6. Use $\widetilde{f}_{\ell}(x)$ to interpolate values at data points $\widetilde{f}(X_i)$.
7. Use the values $\widetilde{f}(X_i)$ in Eq. (7) to estimate adaptive bandwidths h_i for the new kernel $K_{\ell}(x)$.
8. Use new kernel function K_{ℓ} and bandwidths h_i to estimate $\widetilde{f}_{\ell+1}(x)$ using Eq. (1).
9. Return to step 4 until desired closure between $\widetilde{f}_{\ell+1}(x)$ and $\widetilde{f}_{\ell}(x)$. Upon closure, $\widetilde{f}_{\ell+1}(x)$ is the best estimate of $\overline{f}(x)$.

The potentially tricky parts of the algorithm are associated with steps 1, 4, 5, and 7. For step 1, the distributional qualities of the data are unknown, so we use a Fourier transform algorithm to estimate the data density function (see Eq. (17) in the Appendix). By assuming a Gaussian kernel, the initial h_0 can be easily estimated. For step 4, it is important to use a numerical domain for x that is wider than the data values, so that the kernel may extrapolate sufficiently before the smallest data point and after the largest. Furthermore, for widely-spaced and sparse data, the density of calculated points in x must also be chosen to provide sufficient resolution. For step 5, it is not always clear that the mean and standard deviation exist or are the proper scaling metrics for the iterated kernel. A simple example is a stable density, which may have diverging moments, and also rescale differently from, say, a Gaussian density. Here, we suggest using the interquartile range of the data and the kernel for a reasonably close and reliable estimate of the scale of many different density functions. For step 7, we are now using a kernel that is thought to resemble the underlying density, so using $K \to f$ or $\overline{f}$ in Eqs. (3)–(5) and (7) will give different values of ,h_0 etc. More on these points is provided below.

In order to find a "standard" kernel from the previous iteration's density estimate $\overline{f}$, the kernel must have zero mean (so that the subsequent addition of kernels has the same mean as the data). The width of the kernel should be standardized, such as normalizing by a scale factor equal to the standard deviation of the data or central second moment of $\widetilde{f}$. However, many densities have diverging second moments, so a robust method must be found for situations in which the underlying density is unknown. A quick survey of the interquartile range (IQR) and the scale factor of many densities shows reasonably similar relationships. For finite-variance distributions we find, for example, the Gaussian has $\sigma \approx \mathrm{IQR}/1.35$; the exponential $\sigma \approx \mathrm{IQR}/1.1$; the Laplace $\sigma \approx \mathrm{IQR}/0.98$. Infinite variance distributions with closed-form distribution functions (characterized by scale parameter σ) include the symmetric Cauchy with $\sigma \approx \mathrm{IQR}/2$ and the maximally-skewed, 1/2-stable Lévy density with $\sigma = \mathrm{IQR}/9$. Using MATLAB's routine for calculating the CDF of a stable law, we find that, for a maximally-skewed 1.5-stable, $\sigma \approx \mathrm{IQR}/2.13$. With the exception of the Lévy density, it is a reasonable approximation to say that the "width" of the density function may be standardized using $\sigma \approx \mathrm{IQR}/1.5$. Therefore, in the following, to arrive at a "standard" density from the data-kernel, we numerically integrate the intermediate density $\widetilde{f}_{\ell-1}$ to find $\mathrm{IQR} = x_{0.75} - x_{0.25}$, where

$$x_z = \min\left\{ x_j \;\middle|\; \Delta x \sum_{\ell=1}^{n} \widetilde{f}_{\ell-1}(x_j) \geq z \right\}$$

and simply shift and rescale the experimental density by its first moment m_1 and a generic multiple of the IQR so that

$$K_{\ell}(x) = \frac{1}{(\mathrm{IQR}/1.5)} \widetilde{f}_{\ell-1}\left(\frac{x - m_1}{(\mathrm{IQR}/1.5)} \right).$$

As noted before, the kernel is allowed to change after each iteration, and this kernel is checked against the previous iteration's kernel. Iteration is terminated when the kernel converges and the difference between the density estimated with those kernels in successive approximations is sufficiently small. Here, we choose to discontinue the algorithm when the L^2 difference between successive iterations is less then 10^{-9}, where

$$L^2 = \sqrt{\Delta x \sum_{i=1}^{n} \left| \widetilde{f}_{\ell}(x) - \widetilde{f}_{\ell-1}(x) \right|^2}. \tag{9}$$

If the L^2 difference is found to *increase* between iterations, this indicates too large a global bandwidth h_0 (assuming that one starts with a conservatively large value from the Fourier-transform procedure). The too-large value of h_0 makes the kernel itself too smooth and also gives it too large a scale based on the IQR, so convergence will not occur. In practice, there may be a range of h_0 values that leads to convergence based on some numerical threshold of the L^2 norm, so some care needs to be used when adjusting h_0. When the initial h_0 estimate is far from the correct range, one may decrease h_0 by a factor of 0.9. As the algorithm gets nearer the correct kernel and h_0, the minimum of Eq. (9) gets progressively smaller, and there is a danger of overshooting the optimal h_0, so the algorithm slows the adjustment of h_0 by incrementally moving the factor toward unity. After h_0 is adjusted, iteration resumes. In practice, if the minimum L^2 (Eq. (9)) reached for a given h_0 is on the order of 10^{-4}, the value of h_0 is reasonably far from the optimal and may be decreased by about 10%. Each order-of-magnitude improvement in the L_2 convergence is accompanied by moving the factor 2% closer to unity. We also add that other h_0 estimators can be used that may underestimate the optimal h_0, and so the same procedure to adjust the value is done in reverse, say starting with an adjustment factor of 1.1 that decreases toward unity based on minimum L^2 norm seen with any h_0 value. Examples can be seen in the matlab code provided at https://github.com/dbenson5225/kernel-density-estimation

Another important consideration is the construction of the set of points at which the density is calculated. Through some experimentation we find that an optimal set of points is made from a union of (1) a set of appropriately-spaced points between a desired minimum and maximum that is larger than the measured data range and (2) the set of actual random data values X_i. The first set is important so that sufficient interpolation between widely-spaced data is made. The second set is (sometimes) important so that the weights are accurately calculated within Eq. (7). We experimented with neglecting the second set and simply interpolating the density at points x_i from points in the first set, but for "spiky" densities, the results depend too much on the density of points that are specified. If the number of points at which the density is calculated becomes large, it is a simple matter to parallelize a large part of the procedure, because the calculation of $\overline{f}(x)$ in Eq. (1) is independent for any x value.

4. Examples

We investigate the iterative algorithm versus classical (assumed Gaussian kernel) methods for four types of data: (1) symmetric and thin-tailed; (2) maximally-skewed and exponentially-tailed, (3) Symmetric and heavy, power-law tailed; and, (4) maximally skewed and heavy, power-law-tailed. The last is chosen because BTC data are strictly positive and often observed to fall off like $x^{-1-\alpha}$, where α is on the order of 0.5. We also investigate how well the estimators perform over a large realization of random samples and a range of population sizes $n = \{100, 1000, 10,000\}$, inasmuch as large particle numbers (and random arrival times) are typically used. In each case we use estimates of the MISE to measure bias and variance of the estimated density versus a known density on a regular grid in x. A numerical estimate of the MISE is given by an ensemble mean of the L^2 norm, namely

$$\overline{\mathrm{MISE}} = \frac{\Delta x}{M} \sum_{m=1}^{M} \sum_{j=1}^{n} |\overline{f}_m(x_j) - f(x_j)|^2,$$

for a set of M realizations of data with an underlying density f and the corresponding estimates of the density $\overline{f}_m$ on a common grid of estimation points x_j with spacing Δx. For each of the examples, we generate $M = 100$ independent realizations of data from known distributions in order to estimate the densities and resulting MISE (Table 1).

Table 1
Computed ensemble MISE for various kernel estimates from 100 realizations of 1000 random variables. The first row is for uniform initial bandwidth h_0. The second is for single-pass application of adaptive bandwidth $h(x_i)$. The third is iterated Gauss kernel until closure. The fourth is data-based kernel iterated to closure.

Kernel	Normal	Exponential	Cauchy	1.5-Stable
Gauss w/h_0	0.0013	0.157	0.0279	0.0197
Gauss w/h_i	0.0012	0.135	0.0243	0.0118
Gauss iter.	0.0016	0.127	0.0231	0.0100
$\overline{f}$ iter.	0.0014	0.073	0.0150	0.0020

4.1. Gaussian data

We start with Gaussian random variables, in which the data kernel should (nearly) converge to an a priori Gaussian kernel, because the underlying data that builds the data-kernel is Gaussian. Indeed, for a large number of data points (1000), the iterated KDE for the data-based and Gaussian-based kernel are nearly identical, even in the extreme tails (Fig. 1). This example shows the robust nature of the estimation, inasmuch as the Gaussian kernel uses the exact width of the kernel $\sigma = 1$, while the iterated kernel uses a general width estimate of IQR/1.5. In actuality, the width of a Gaussian is $\sigma = \text{IQR}/1.34$. It is worth noting that closure to the final kernel usually takes between 5 and 7 iterations. Furthermore, because the value of h_0 is not set exactly (which would require identifying the data as Gaussian before interpolating), the iterated kernels have similar magnitudes of MISE as single-pass adaptive Gaussian kernels and convolution with a single value of h_0 (Table 1).

4.2. Exponential data

Next, we use a shifted exponential (the arbitrary shift is added to ensure functionality of the code) with density function

$$f(x) = \begin{cases} \frac{1}{\sigma}\exp\left(-\frac{x-\mu+1/\sigma}{\sigma}\right), & \text{for } x \geq \mu - 1/\sigma \\ 0, & \text{else.} \end{cases}$$

This density has arbitrary mean μ and variance $1/\sigma^2$. In the plots that follow we set $\mu = 0$ and $\sigma = 1$. For this skewed density it becomes clear that a symmetric (Gaussian in this case) kernel is not an effective interpolant (Fig. 2). While Silverman (1986) suggests using a skewed (say, lognormal) kernel for this kind of data, our method does not rely on interpretation and user intervention for kernel selection. And while a Gaussian kernel is not particularly useful for this kind of data, the iterated data-based kernel typically has MISE of about half that of the Gaussian (Table 1). Because the underlying optimal h_0 is much smaller than that estimated from assuming a Gaussian kernel, the iterations do not converge (and in fact tend to diverge) until h_0 decreases several times, requiring on the order of 30 or more iterations for 1000 data points.

4.3. Cauchy data

Heavy-tailed data present a problem for kernel density estimation because of the extremes that may accompany the data. This leads to very wide spacing between extreme data points and difficulty interpolating the density here. This also means that the x-discretization of the kernel must use a large number of points in order to represent the near-origin "spikiness" of the density as well as the very long range. The existence of one or two super-extreme values can lead to numerical problems. In our 100-realization ensemble of 1000 Cauchy data points, two realizations failed to converge in 100 iterations with the data-based kernel because of data values in the 50,000 range. A typical realization shows that the converged data-based kernel tends to both interpolate between, and extrapolate beyond, extreme values better than the Gaussian kernel, but still represents the fine-scale near the origin where most of the data reside (Fig. 3). In the ensemble, the MISE estimated for the data-based kernel is substantially less than for the Gaussian kernels (Table 1).

4.4. Maximally-skewed α-stable data

Stable random variables (RV) are characterized (among many other ways) as those to which sums of IID random variables converge (Samorodnitsky and Taqqu, 1994). Sums of finite-variance RVs converge to (are in the domain of attraction of) a Gaussian, which is itself a stable RV. When, for some constant $0 < \alpha < 2$, only those moments of order α and greater are infinite (such as for Pareto (power-law) distributed RVs), then those RVs are in the domain of attraction of a α-stable. These RVs arise in hydrology quite naturally, because they describe waiting times that a particle might take when trapped in a sequence of fractal immobile zones (see Schumer et al., 2003; Benson et al., 2013). Depending on the skewness parameter, one or both of the tails of a α-stable density function decay like $\sim |x|^{-1-\alpha}$; therefore all moments of order α and greater diverge. The density is only expressible in closed-form for a few instances, but most statistical packages will readily calculate the density to any desired tolerance and generate sequences of the random variable. Here, we choose a maximally-skewed, standard 1.5-stable for analysis, using the parameterization in the MATLAB statistics package (also called the 0-parameterization in Nolan, 2018). The ensemble MISE for the data-based kernel is about 1/5 that of the iterated Gaussian kernel, suggesting that both the heavy-tailed and skewed nature of this example is especially well-suited to our proposed method (Fig. 4).

5. Particle breakthrough (concentration) data

The creation of "breakthrough curves" (BTC) from particle-tracking simulations is a tricky proposition. Classically, histograms are used, which means manually choosing either constant or variable bin sizes and locations. The variance of the estimated density is inversely proportional to bin size, total number of particles, and the estimated concentration (Chakraborty et al., 2009), and the histogram-based density is discontinuous and may frequently be zero when particle arrival times are widely separated, especially in the late-time tail. The zeros make comparison to non-zero data difficult (e.g., using weighted least-squares), so several methods are typically used to create a non-zero PDF interpolation.

The first set of constructions of arrival time pdfs, which we will call "naive estimators" is based on simple linear interpolation of arrival times. For example, one may construct (by several means) an empirical cumulative distribution function (ECDF) that is strictly increasing and, then make a non-zero empirical PDF using finite differences on the ECDF. In particular, order the particle arrival times of N particles $T_1, T_2, \ldots T_N$ and at each point the $\text{ECDF}(T_i) = i/N$. Then the empirical PDF is $\text{EPDF}((T_{i+1} + Ti)/2) = (\text{ECDF}(T_{i+1}) - \text{ECDF}(T_i))/(T_{i+1} - T_i); i = 1..N-1$. The ECDF can also use a regularly spaced time grid and count numbers of particles arriving between grid points (i.e., bins), and empty bins are neglected, once again giving a strictly increasing ECDF. In this section, we compare these two naive estimators to the iterative kernel-based techniques developed in this paper along with prior deterministic kernel-based methods (Fernàndez-Garcia and Sanchez-Vila, 2011; Pedretti and Fernàndez-Garcia, 2013).

For particle arrival times, we solved for the hydraulic head H in the steady-state groundwater flow equation $\nabla \cdot K \nabla H = 0$ in 2-D using finite-differences on a square 128 × 128 m grid with constant grid discretization of 1 × 1 m (Fig. 5a). The hydraulic conductivity K is a scalar log-Normal random variable with a mean of $\ln(K) = 1$, standard deviation of $\ln(K) = 4$, and an exponential autocorrelation function for $\ln(K)$ with correlation length of 5 m. The left and right boundaries $x = 0$ and $x = 128$ are Dirichlet with $H = 1$ and $H = 0$, respectively. The top and bottom boundaries $y = 0$ and $y = 128$ are Neumann with

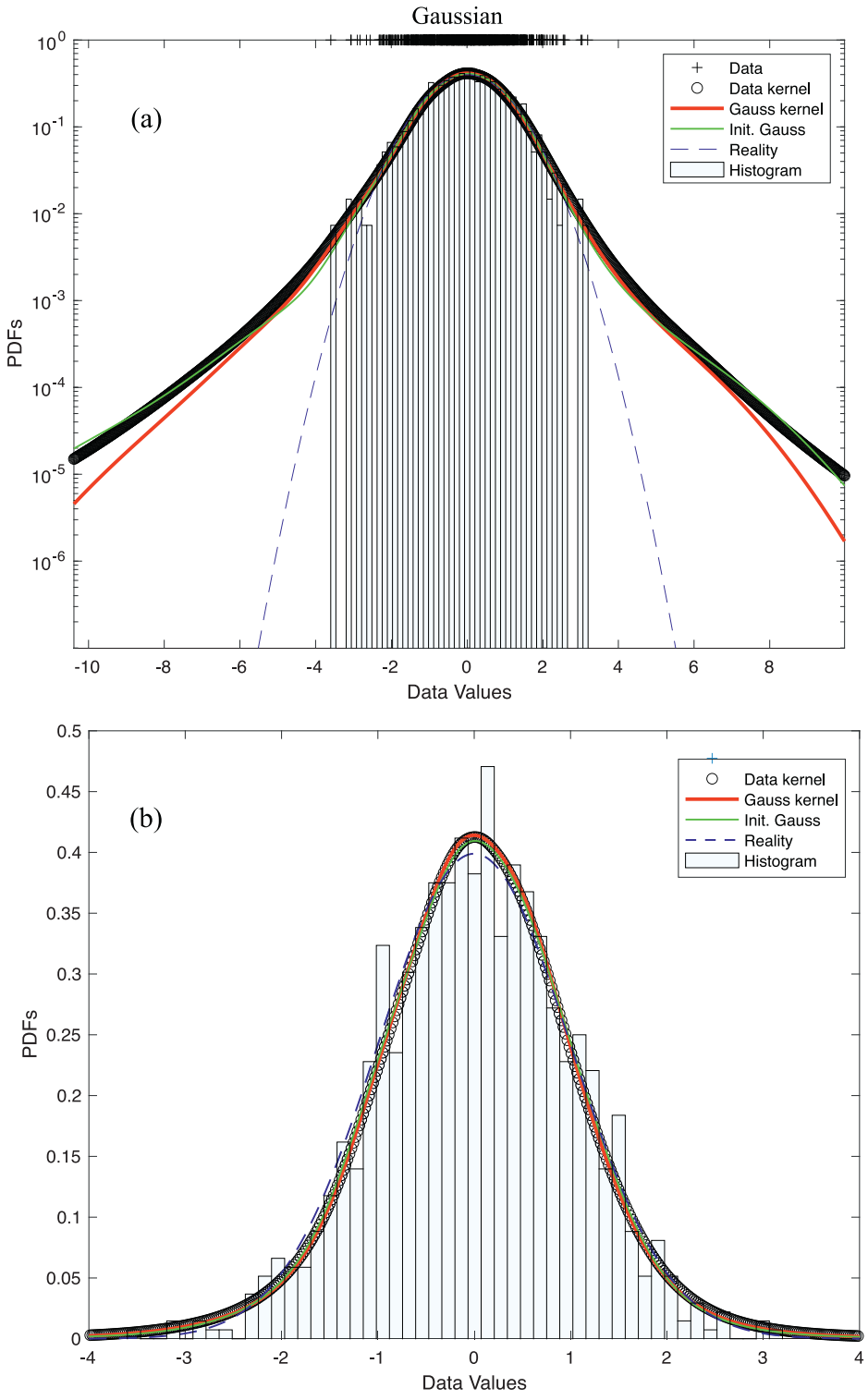

Fig. 1. (a) Semilog and (b) linear plots of iteratively estimated densities for a single realization of 1000 Gaussian data points using data-based kernel (black symbols) and Gaussian-based kernels (red curves). Also shown are the single-pass Gaussian kernel estimate (green curves) and 20-bin histograms. Blue dashed line is underlying "true" density function. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\partial H/\partial y = 0$. The resulting velocities take the solved H field and apply $v = -K\nabla H/\phi$, where ϕ is assumed a constant porosity of unity, once again using finite-differences. These velocities vary in magnitude from about 3×10^{-7} to 2.1 m/d (Fig. 5a). Particles are placed in a line near the left boundary and each particle's position vector tracked via a discretized Ito equation $X(t + \Delta t) = X(t) + (v + \nabla \cdot (D)) + \sqrt{2\Delta t} B\mathcal{N}$, where $D = (D_m + A_T|v|)I + (A_L - A_T)vv^T/|v|$ is a dispersion tensor that has a decomposition $D = BB^T$, $\mathcal{N}$ is an independent 2-D standard normal vector, $D_m = 8 \times 10^{-5}$ m^2/d is molecular diffusion, $A_T = 10^{-3}$ m is transverse dispersivity, and $A_L = 5 \times 10^{-3}$ m is longitudinal dispersivity. The number of particles placed in any cell is proportional to the velocity magnitude in that cell (i.e., a flux-weighted source). A plot of particle positions at elapsed times of 1 and 250 days (just before arrival of first particle at the right-hand side) suggests the wide range of arrival times that can be expected. We ran simulations using 500, 5000, and 50,000 particles to judge the efficacy of density estimates.

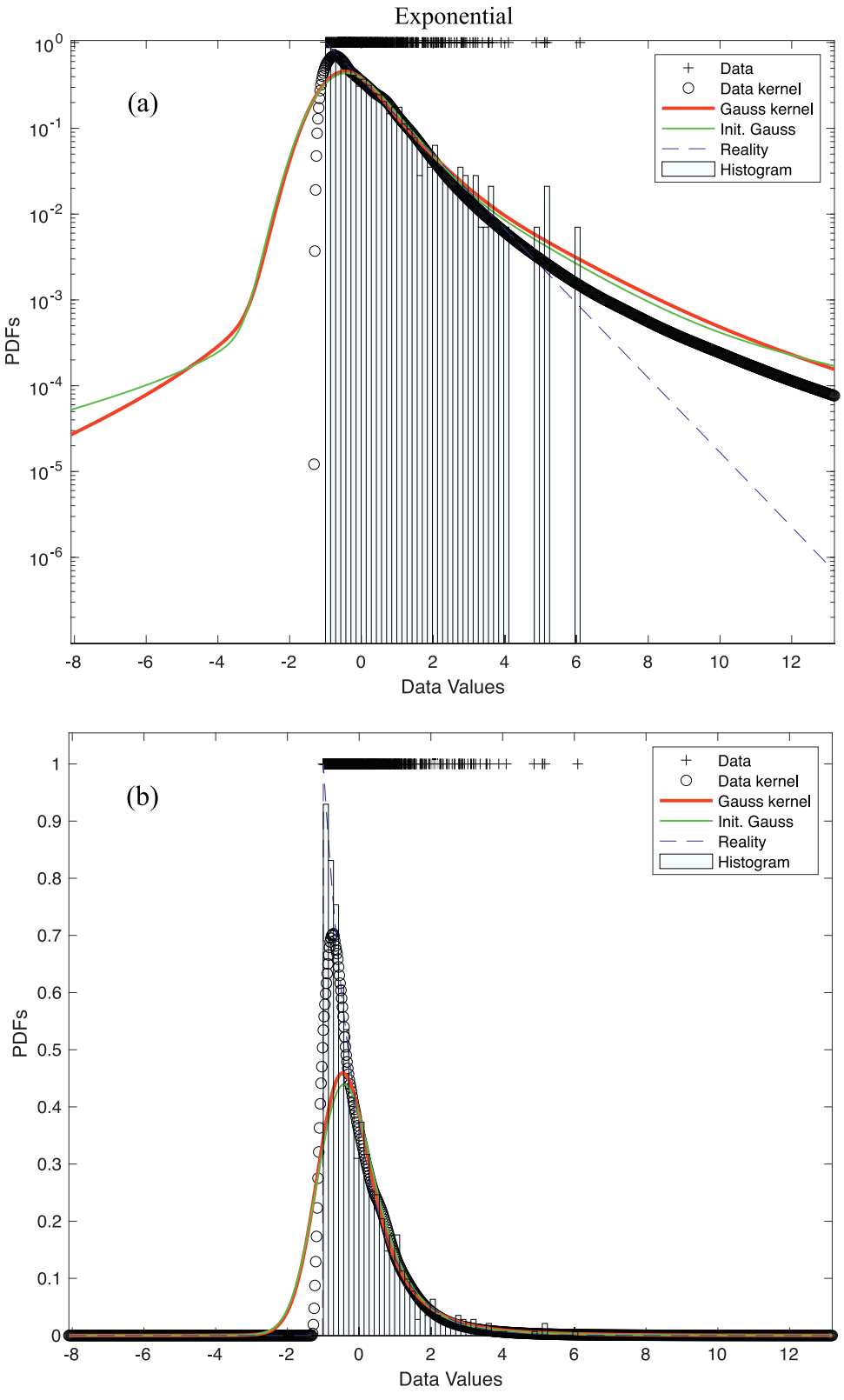

Fig. 2. (a) Semilog and (b) linear plots of iteratively estimated densities for a single realization of 1000 shifted Exponential data points using data-based kernel (black symbols) and Gaussian-based kernels (red curves). Also shown are the single-pass Gaussian kernel estimate (green curves) and 20-bin histograms. Blue dashed line is underlying "true" density function. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The data-based kernel estimates developed in this work are remarkably similar for the different particle numbers on a linear plot (Fig. 5b). Two naive estimates of the EPDF using 50,000-particle arrival times (Fig. 5f) show considerable noise at late time due to wide separation of late particle arrival times. This effect can be counteracted by using much larger particle numbers (e.g., Labolle et al., 1996; Kang et al., 2017; Carrel et al., 2018). This computational burden may be reduced in the case of particle-tracking simulations for conservative solutes because they are highly parallelizable (Rizzo et al., 2019). However, nonlinearly reacting solutes have yet to be parallelized in three-dimensions

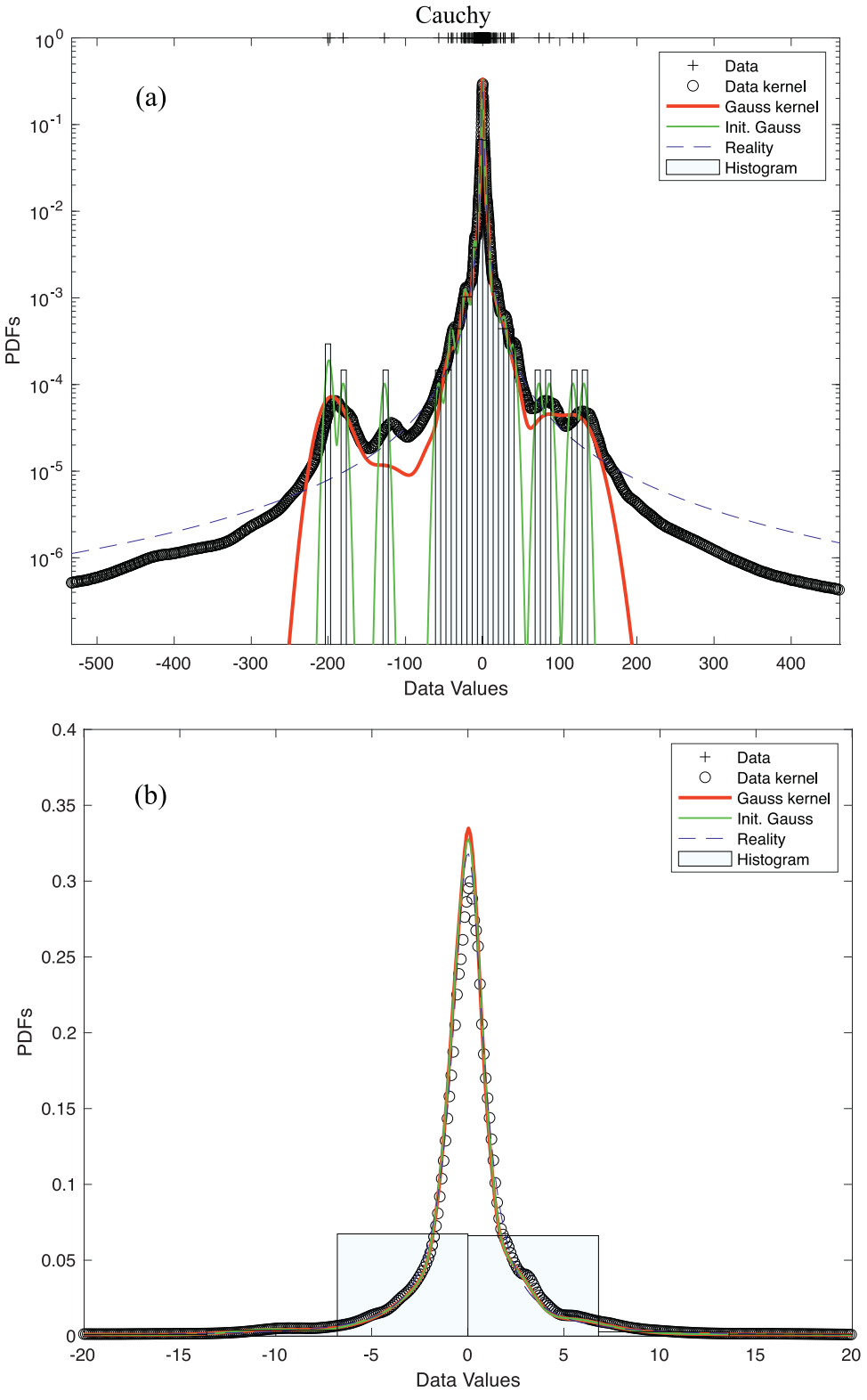

Fig. 3. (a) Semilog and (b) linear plots of iteratively estimated densities for a single realization of 1000 Cauchy data points using data-based kernel (black symbols) and Gaussian-based kernels (red curves). Also shown are the single-pass Gaussian kernel estimate (green curves) and uniform 20-bin histograms. Blue dashed line is underlying "true" density function. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(Engdahl et al., 2019). The naive estimators also do not have density weight before the first particle arrival because the ECDF is zero for any time before the first particle. This is a commonly accepted, but ultimately incorrect, feature: as the number of particles becomes larger (or goes to infinity in the case of kernel density estimates) the empirical density of early arrivals should grow. In other words, by calculating the density on a time grid from zero to 10^6 days, the only imposed constraint is that the first arrival is non-negative. The early-time density estimates for larger particle numbers have greater probability for early time than the 500-particle, which can be identified using logarithmic time axes

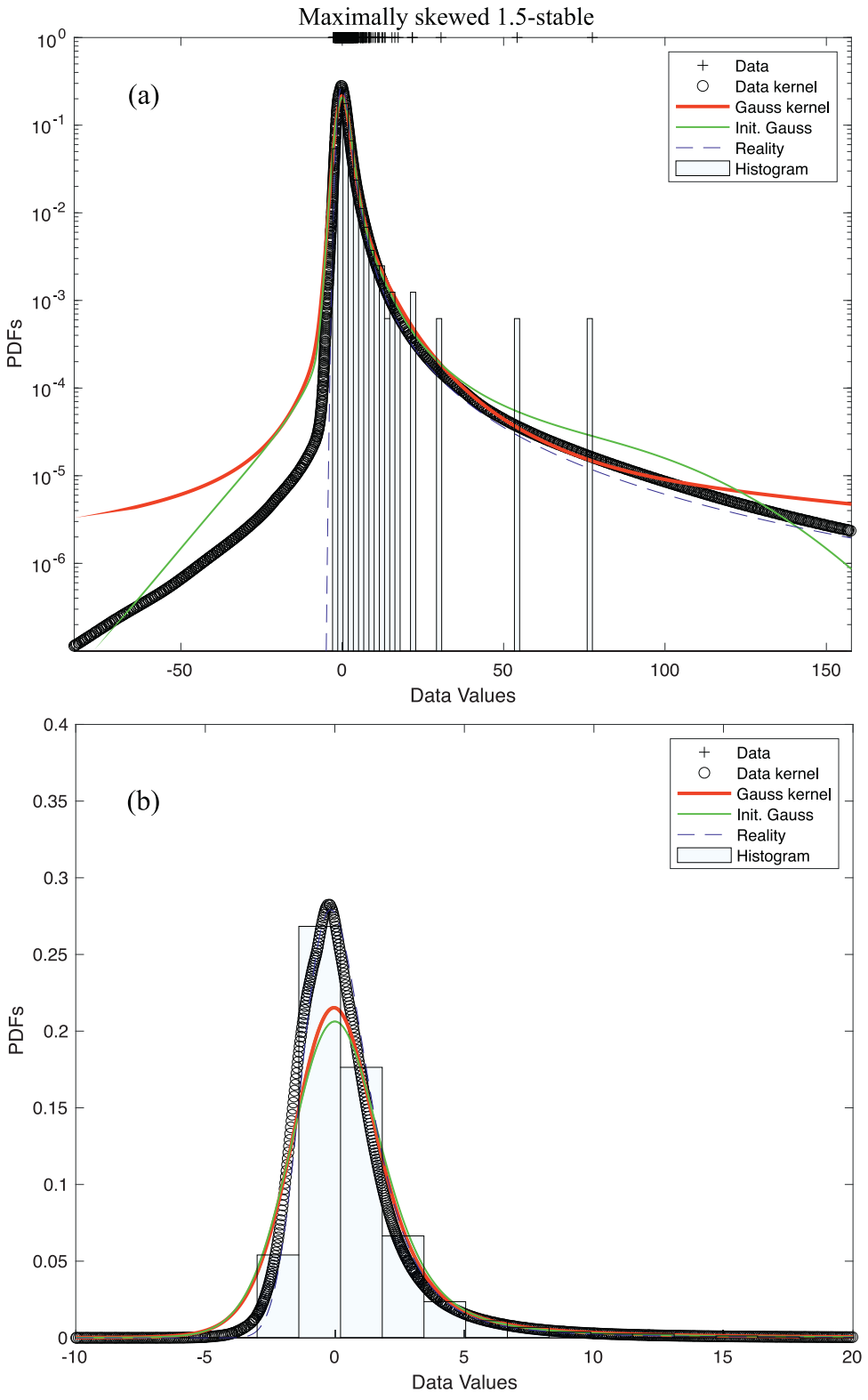

Fig. 4. (a) Semilog and (b) linear plots of iteratively estimated densities for a single realization of 1000 maximally-skewed, 1.5-stable data points using data-based kernel (black symbols) and Gaussian-based kernels (red curves). Also shown are the single-pass Gaussian kernel estimate (green curves) and uniform 20-bin histograms. Blue dashed line is underlying "true" density function. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(Fig. 5d and e). The late-time tail estimates are smoothly interpolated and very close for all three particle numbers (Fig. 5e) until some time after the final particle arrival time in the 500-particle simulation, when that tail starts to drop somewhat compared to the higher particle numbers. The 500-particle density is smoothly extrapolated over 50 times longer than the final arrival because of the kernel shape. Overall, it is fair to say that the 5000-particle simulation gives similar enough results to the 50,000-particle simulation that the latter is superfluous.

A serious problem with the kernel density estimates is that the time grid along which the density (hence kernel for subsequent iterations)

Table 2
Computed values of global bandwidth h_0 for the particle-tracking data using either 500, 5000, or 50,000 particles. FT denotes Fourier-transform routine in Appendix C. Plug-in refers to method of Engel et al. (1994). Iterated refers to values from the iterated kernel and h_0 method from Section 5. Values in days.

Method	500	5000	50,000
FT	2033	2981	2370
Plug-in	108	50	22
Iterated	256	267	253

needs to be quite large. The density is "spiky" enough to warrant a grid size of about 20 days or less, and the last particle arrives on the order of 10^6 days, so a linearly partitioned grid is a vector on the order of 50,000 to 100,000 elements, making the convolutions quite slow. Convergence is also slow because of the high skewness, so the estimates are computationally expensive. Because of this we looked at two alternatives: (1) use of a log-spaced discretization grid (which was used to generate Fig. 5), and (2) the *ad-hoc* correction of Pedretti and Fernàndez-Garcia (2013), which is detailed immediately.

5.1. Experiments with the universal adaptive bandwidth of Pedretti and Fernàndez-Garcia (2013)

These authors recognize that early arrival tails of a BTC are often much thinner than late-arrival tails and seek to adjust the bandwidth assigned to early versus late data accordingly. The authors choose to use the smaller global bandwidth h_0 at smaller T_i that relatively smoothly transitions to the density-adjusted value for later data. There are an unlimited number of possible schemes to do this. Pedretti and Fernàndez-Garcia (2013) suggest constructing the ECDF(T_i), which is monotonically increasing with arrival time T_i, and constructing a variable bandwidth at each point by taking a weighted average of the single global bandwidth h_0 and the classical adaptive bandwidth:

$$h_2(T_i) = (1 - \mathrm{ECDF}(T_i))h_0 + \mathrm{ECDF}(T_i) \times h_i, \tag{10}$$

where h_2 is their "universal global bandwidth" (UAB), and h_i is the adaptive bandwidth given in Eq. (7). Pedretti and Fernàndez-Garcia (2013) choose a standard Gaussian kernel in their paper so we do the same here. This leaves only the selection of the global bandwidth h_0 as a potential difference in the implementation. Pedretti and Fernàndez-Garcia (2013) use a code supplied by Engel et al. (1994) that uses a prescribed kernel to interpolate data points to predict the value of $\int f''(x)dx$ only once, prior to estimation of $\widetilde{f}(x)$. This value is used in (5) to get a value of h_0. We have shown above that there are several approaches to arriving at a value of h_0 to be used in Eqs. (5) and (10). For example, we may use the Fourier methods in Appendix C.

We apply the UAB method using fixed estimates of h_0 from the plug-in method and from our Fourier-transform method (Fig. 6). Once again we calculate the densities on time points made from a union of two sets: (1) a set of 20,000 logarithmically-spaced time points between zero and 10^6 days and (2) the set of actual arrival times. For the 50,000 particle simulation, our Fourier transform algorithm gives $h_0 = 2370$ days. The plug-in method of estimating (5) from Engel et al. (1994), used by Pedretti and Fernàndez-Garcia (2013), gives an estimated $h_0 = 22$ days (Table 2). The fact that these two estimates differ by two orders-of-magnitude is remarkable by itself and points to the potential errors of *a priori* h_0 estimates. Using a Gaussian kernel with these estimates and the UAB (10) gives clearly over-smoothed and under-smoothed density estimates (Fig. 6a). The under-smoothing by the plug-in value of h_0 used in the UAB is shown by the failure to interpolate between the many late-time arrivals (due to the relatively narrow Gaussian kernels there), while the over-smoothing of the initial Fourier h_0 is shown by the relatively high weight of the near-zero arrival time PDF. Similar discrepancies are seen in both the densities and values of h_0 from the FT and plug-in methods for the other particle numbers (Table 2). Also shown in Table 2 and Fig. 6a–c are the intermediate, iterated values of h_0 that accompany the iteration of the kernels from Section 5.

This analysis of BTC did not allow an assessment of which model had the better "fit" because the underlying true density was unknown. The particle arrival-time data have several features common to the 1.5-stable density used in Section 4.4 including thin leading (early-time) tail, fat trailing tail, and a high degree of skewness. We applied the UAB method using a standard Gaussian kernel and the iterated kernel algorithm developed in this paper to data taken from a known maximally-skewed, 1.5-stable distribution with one caveat: to eliminate one variable, in both methods we use the h_0 from the plug-in method (Engel et al., 1994), as did Pedretti and Fernàndez-Garcia (2013). An ensemble of 100 data realizations, each with 1000 random variables, were generated to calculate the ensemble mean MISE using Eq. (4). These values were 1.1×10^{-4} and 1.5×10^{-4} for the UAB and iterated kernels approaches, respectively. The UAB method paired with the plug-in h_0 clearly does a good job in the areas around the peak (Fig. 7a,b), where the densities have the greatest weight in Eq. (4). Similar to the BTC data above, the UAB method succeeds in re-creating the thin leading tail of the 1.5-stable density, but fails to interpolate between large data values or extrapolate beyond the largest data value (Fig. 7a,b). The iterated kernel method outperforms the Gauss-kernel method of Pedretti and Fernàndez-Garcia (2013) in both interpolating and extrapolating the large-data tail (Fig. 7c,d), but does tend to put too much density weight on the thin-tailed small data values relative to the known, real density. One might also conclude that the UAB method could be combined with the iterated kernel method to achieve good estimates of both the early and the late tails. Indeed, iterating the kernel function until closure and then applying the UAB does give essentially identical late tail estimates and steeper early tail estimates (red curve, Fig 7 c), although we note that the estimated MISE using the UAB and the iterated kernel was about 20% worse due to slightly poorer fits around the peak. Of course using the UAB requires inspection of the data to decide whether this adjustment is appropriate.

It is interesting to note that the plug-in estimates of h_0 use a method that evaluates the integrals in Eq. (5) based only on data values. Our method of iterating the kernel recognizes that the "best" estimate of the true density $f(x) \approx \overline{f}(x)$ evolves, and that $\int f''(x)dx$ might be improved using intermediate values of $\overline{f}(x)$. We implemented this procedure by repeatedly estimating $\int f''(x)dx$ by finite differences and trapezoidal integration. The new value of h_0 was then used in the UAB (Eq. (10)) until closure was reached. In all cases, the value of h_0 that was estimated was smaller than the one-time plug-in estimate and the overall MISE was worse, so those density estimates are not shown.

Finally, in the context of fitting models to data, seeking to minimize the MISE is not always the most appropriate choice. Any kernel density estimate constitutes a model of the data, and the classical measures of model fit should apply, including maximum likelihood estimation (MLE) and entropy considerations that include parametric and computational parsimony (e.g., Akaike, 1974; Benson et al., 2020). In particular, Chakraborty et al. (2009) make an argument that the variance of concentration values in a binned density estimate would have a variance proportional to the estimated concentration, and that those variances, while dependent upon each other, could be treated independently. In the present case, we conjecture that an MLE would seek to minimize an integrated weighted squared difference of the estimated and real densities, where the weights are $1/\overline{f}(x)$. Applying this formula to the data in this section returned machine infinities for the UAB method using Gaussian kernels because of the machine zeros for the estimates of the density in many places (e.g., Fig. 7a,b). The iterated kernel returned finite values for all realizations in the ensemble with an average weighted MISE of 0.02. From a standpoint of comparing some estimated BTC to real data, this coincides with a desire to have good interpolations of particle-tracking simulations on the low concentration tails.

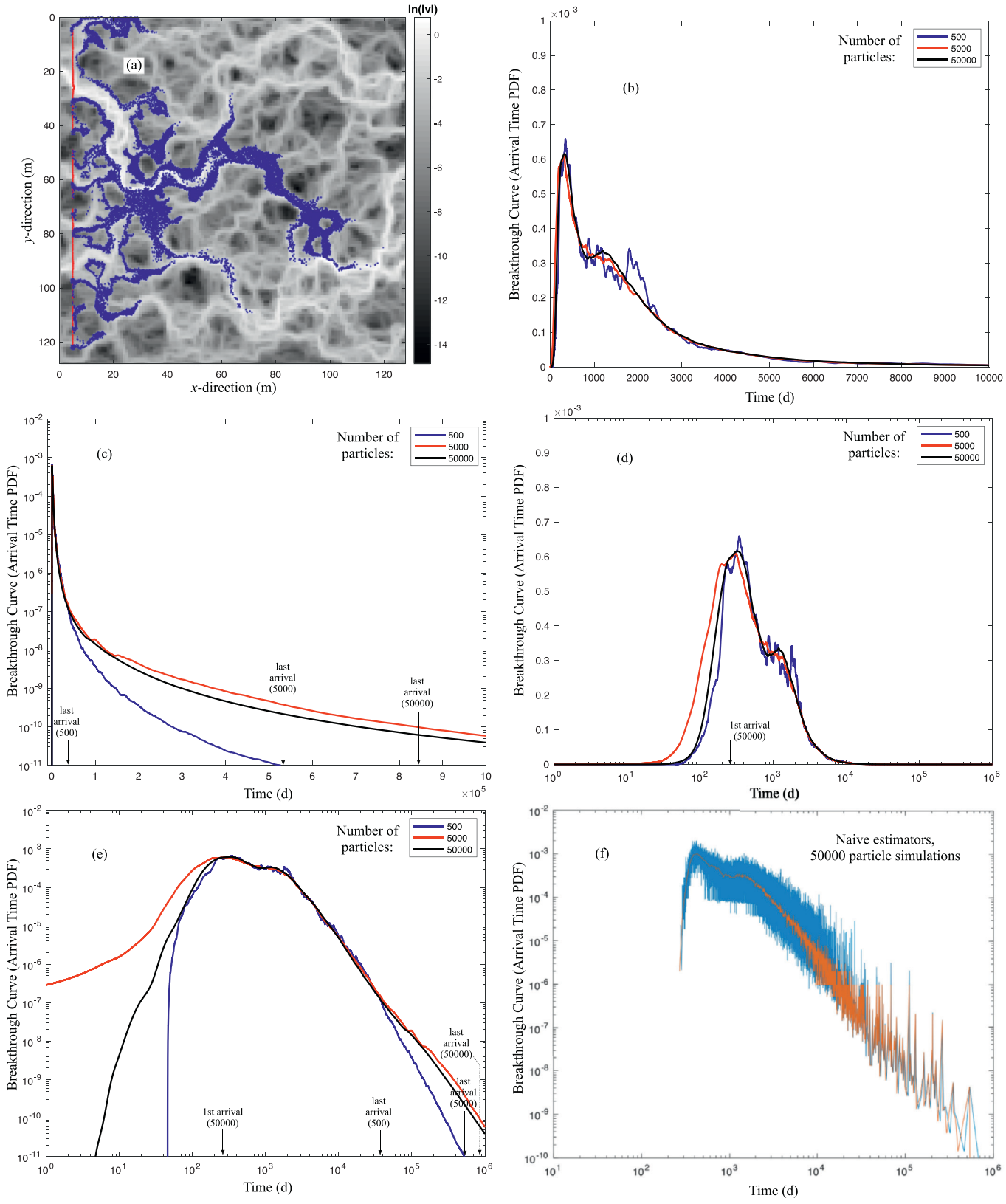

Fig. 5. (a) 50,000-particle simulation positions at $t = 1$ day (red) and 250 days (blue) on a gray-scale quilt of log-velocity magnitude. Mean flow is left-to-right. (b) through (e) Plots of estimated arrival-time densities (breakthrough curves) using differently scaled axes for 500, 5000, and 50,000 particle simulations. The densities are estimated using the data-based kernel. (f) Plots of naive estimators based on construction the ECDF: blue lines use all arrivals, orange lines use binned data, which smooths the plot where multiple arrivals are found in each bin. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

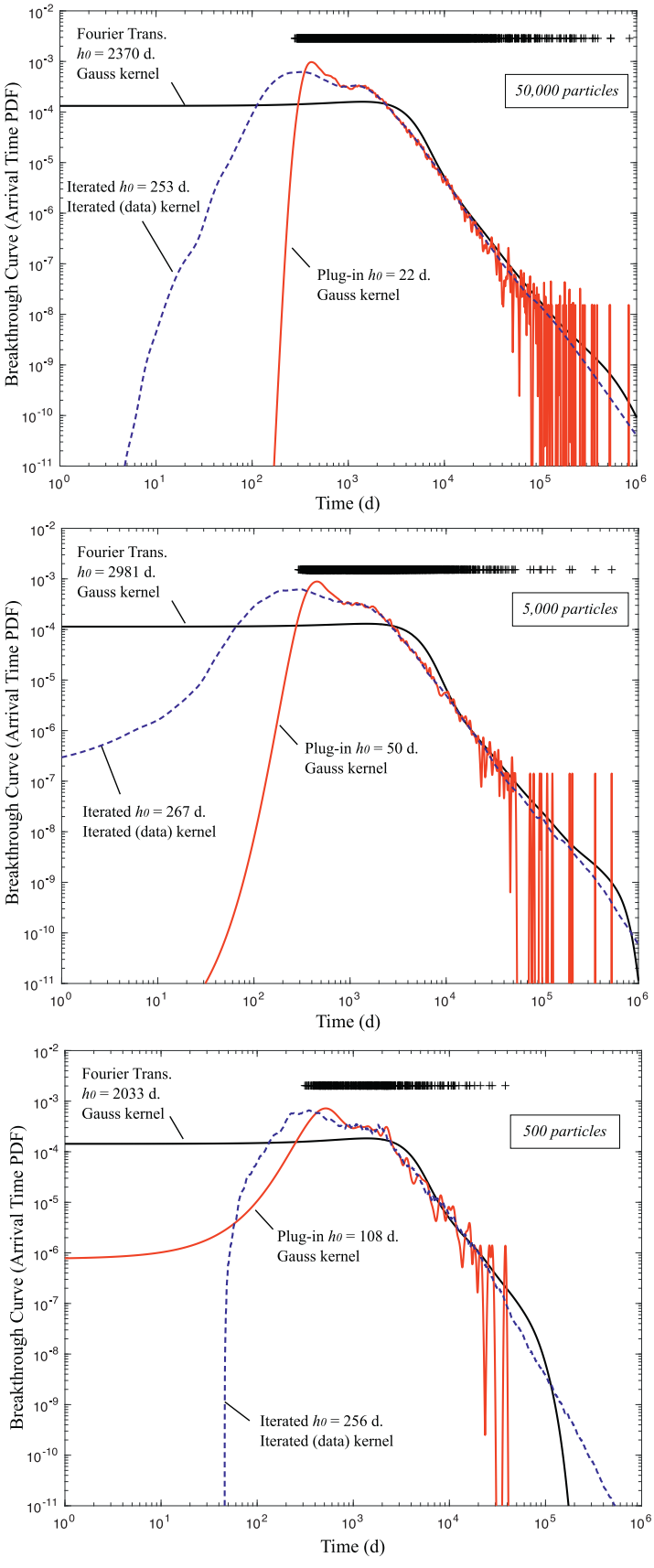

Fig. 6. Log-log plots of estimated arrival-time densities (breakthrough curves) using UAB of Pedretti and Fernàndez-Garcia (2013) with different values of h_0 estimated either by Fourier transform (black), or Plug-in method Engel et al. (1994) (red). Also shown as blue dashed lines on the plots are the curves from Fig. 5e that use our iterated kernel (and iterated h_0) method. (a) 50,000-particle simulation. (b) 5000-particle simulation. (c) 500-particle simulation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

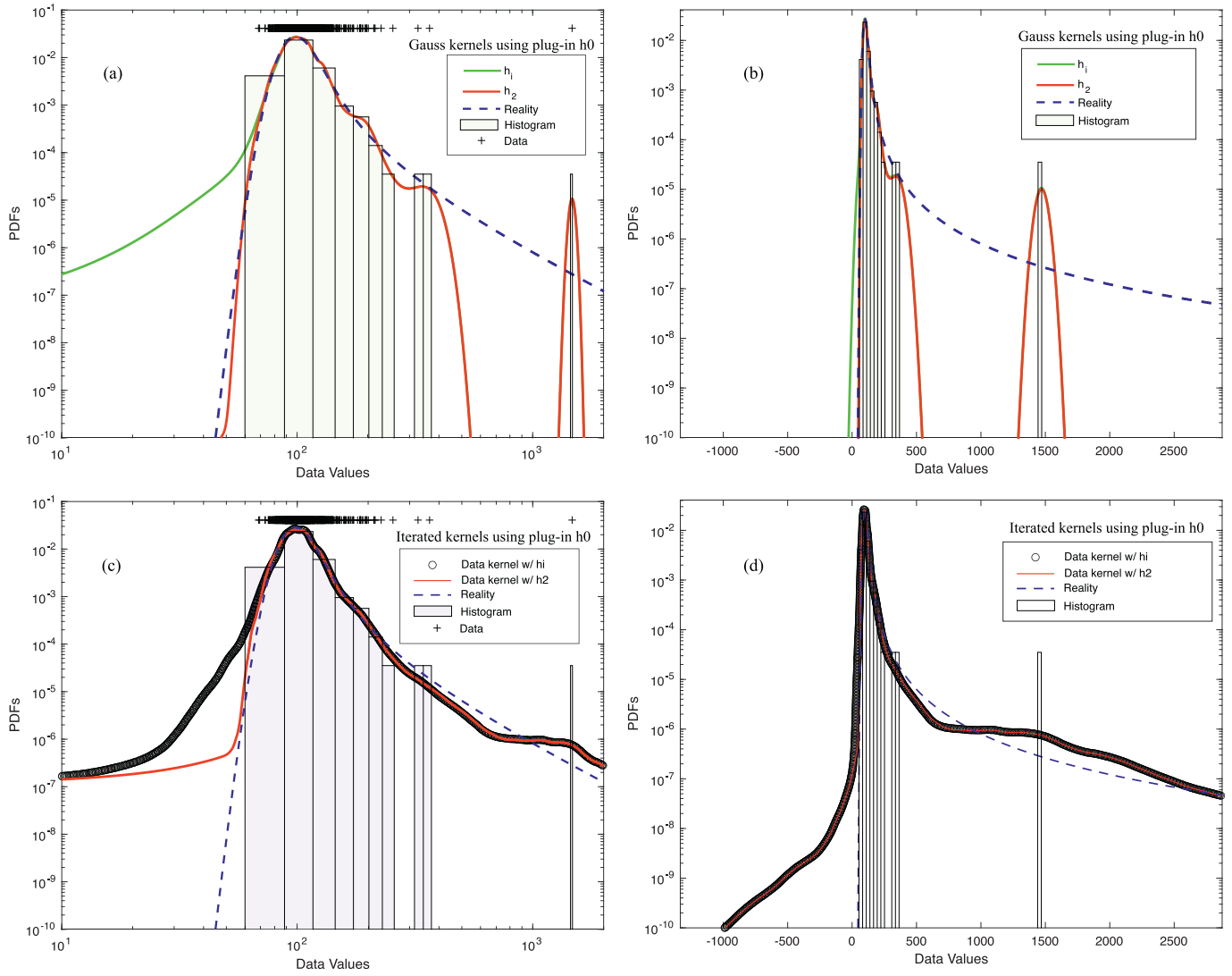

Fig. 7. Plots of density estimates from a single realization of 1000 maximally-skewed, 1.5-stable data points (a) Log-log and (b) Semi-log plots using Gaussian kernels and weights given by Eq. (7) and UAB of Pedretti and Fernàndez-Garcia (2013); (c) log-log and (d) semilog plots using iterated, data-based kernels developed in section, also using weights given by Eq. (7) and UAB. The value of global bandwidth h_0 is held the same and given by plug-in formula of Engel et al. (1994).

6. Conclusions and recommendations

The application of an optimal, iterative algorithm for kernel density estimation (using an evolving, data-based kernel) is possible with a few caveats. First, because the underlying data density is, in general, unknown, a method is needed to estimate a MISE-minimizing global bandwidth h_0. We show that a Fourier-transform based method can obtain an unbiased estimate for any kernel, and the exact value for a Gaussian kernel. This Gaussian kernel "starts" the new algorithm by generating a first continuous estimate of the density. This density is then used to construct the kernel for subsequent density estimates. Second, creating a "standard" kernel based on the current iterated density estimate requires an estimate of the scale parameter of the density. We use a value based on the interquartile range divided by 1.5. This value is intermediate for several known densities and works well for a range of known densities. Third, because the final iterated version does not use a Gaussian kernel, the initial estimate of h_0 will necessarily be in error.

We show that for some common densities, the Fourier-transform estimate of h_0 will err on the large side. Furthermore, we show for a wide range of densities that the estimate of $h_0 \sim n^{-\gamma}$, with γ being a minimum for Gaussian data and increasing systematically as the tails become heavier (including exponential and power-law). Therefore, the iterative scheme allows h_0 to decrease if the algorithm fails to demonstrate convergence. As expected, for Gaussian data, the data-based kernel converges rapidly to a form similar to that given by the Gaussian kernel. For skewed and/or heavy-tailed data, convergence is slower and only occurs when h_0 is allowed to decrease toward its actual, optimal value (or range).

Overall, the data-based, iterated kernel gives significantly smaller ensemble MISE than either (1) an iterated (adaptive bandwidth) Gaussian kernel, (2) a single-pass adaptive bandwidth Gaussian kernel, and (3) a single-pass Gaussian kernel with a single global value of h_0. The new algorithm is clearly better when the "non-Gaussian" aspects of the underlying data increase, including skewness and heavy (exponential or power-law) tails. When applied to particle arrival times that are heavy-tailed, the iterated kernel kernel and h_0 provide smooth and continuous interpolation and extrapolation of widely spaced late-time arrivals even when few particles (5000) are used. The iterated kernel approach does over-smooth the early time data, and the UAB approach can be used to thin the estimated early-time tail. If a particle-tracking model is used to compare to real data (whose measurement times will not necessarily correspond to particle arrival times), the methods developed here will

be key to providing good interpolations between a simulation's widely-spaced late time arrivals.

The derivation of the optimal kernel and global bandwidth solved a minimization problem for one variable h_0 based on kernel shape and the Fourier transform of actual data (Appendix C). A more difficult problem of optimizing a separate h_i for each data point may be possible using cluster identification (Wu et al., 2007) or multi-Gaussian kernel localization techniques (Sole-Mari and Fernàndez-Garcia, 2018). These methods would eliminate the potentially dubious Taylor-series-based assumptions of the power-law weighting scheme used in Eq. (7) to adjust each data point's bandwidth. We leave this for a future paper.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

David A. Benson: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft, Writing - review & editing. **Diogo Bolster:** Formal analysis, Writing - review & editing. **Stephen Pankavich:** Formal analysis, Writing - original draft, Writing - review & editing. **Michael J. Schmidt:** Software, Writing - review & editing.

Acknowledgments

We thank the editor and reviewers, including Daniel Fernàndez-Garcia, for extremely helpful comments. We also thank Daniele Pedretti for sending source fortran code of their UAB estimator. This material is based upon work supported by, or in part by, the US Army Research Office under Contract/Grant number W911NF-18-1-0338. The authors were also supported by the National Science Foundation under awards EAR-1417145, DMS-1614586, DMS-1911145, EAR-1351625, EAR-1417264, EAR-1446236, and CBET-1705770. Sandia National Laboratories is a multi-mission laboratory managed and operated by the National Technology and Engineering Solutions of Sandia, L.L.C., a wholly owned subsidiary of Honeywell International, Inc., for the DOEs National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Matlab codes for generating all results in this paper are held in the public repository https://github.com/dbenson5225/kernel-density-estimation.

Appendix A. Mathematical background

The idea of optimal global bandwidth (Silverman, 1986) stems from using a truncated Taylor series to represent the terms in the MISE. We begin with the fact that the expectation of the density estimate constructed from a set of independent observations is the sum of the expectations of the weights associated with each observation so that

$$\mathbb{E}[\overline{f}(x)] = \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\frac{1}{h}K\left(\frac{x-X_j}{h}\right)\right] = \int \frac{1}{h}K\left(\frac{x-\xi}{h}\right)f(\xi)d\xi. \tag{11}$$

Similarly, we compute the variance as

$$\begin{aligned}\mathrm{Var}[\overline{f}(x)] &= \mathrm{Var}\left[\sum_{j=1}^{n}\frac{1}{nh}K\left(\frac{x-X_j}{h}\right)\right] = \sum_{j=1}^{n}\frac{1}{n^2}\mathrm{Var}\left[\frac{1}{h}K\left(\frac{x-X_j}{h}\right)\right]\\ &= \frac{1}{n}\int\left(\frac{1}{h}K\left(\frac{x-\xi}{h}\right)\right)^2 f(\xi)d\xi\\ &\quad - \frac{1}{n}\left(\int\frac{1}{h}K\left(\frac{x-\xi}{h}\right)f(\xi)d\xi\right)^2.\end{aligned}$$

The bias at any point is

$$\begin{aligned}B(x) = \mathbb{E}[\overline{f}(x)] - f(x) &= \int\frac{1}{h}K\left(\frac{x-\xi}{h}\right)f(\xi)d\xi - f(x)\\ &= \int K(z)f(x-hz)dz - f(x)\\ &= \int K(z)(f(x-hz)-f(x))dz.\end{aligned}$$

With this, the MISE is written as

$$\left.\begin{aligned}\mathrm{MISE} &= \int B(x)^2dx + \int\mathrm{Var}[\overline{f}(x)]dx\\ &= \int\left(\int K(z)(f(x-hz)-f(x))\right)^2 dzdx\\ &\quad + \frac{1}{n}\left(\int\int\frac{1}{h^2}K\left(\frac{x-\xi}{h}\right)^2 f(\xi)d\xi dx\right.\\ &\quad \left. - \int\left(\int\frac{1}{h}K\left(\frac{x-\xi}{h}\right)f(\xi)d\xi\right)^2 dx\right).\end{aligned}\right\} \tag{12}$$

The bias contribution is simply the effect of the kernel smoothing on the real density, which does not depend on the sample size n. The variance obviously grows smaller as n increases and (not completely obviously) as h increases. This expression is difficult to minimize exactly, although for both K and f Gaussian, the convolutions yield Gaussians and an exact result may be computed (Silverman, 1986). The vast majority of work done with KDE is to use asymptotic expansions of certain functions, with some questionable assumptions regarding their validity and application. For example, the density at $x - hz$ is typically approximated for $hz \to 0$, even though the goal is to find a finite h and z may be arbitrarily large in the integral. Still, using a truncated Taylor series, namely

$$f(x-hz) = f(x) - hzf'(x) + \frac{1}{2}h^2z^2f''(x) + \mathcal{O}(h^3)$$

gives

$$\begin{aligned}B(x) &= -hf'(x)\int zK(z)dz + \frac{1}{2}h^2f''(x)\int z^2K(z)dz + \mathcal{O}(h^3)\\ &= -hf'(x)\mu_1(K) + \frac{1}{2}h^2f''(x)\mu_2(K) + \mathcal{O}(h^3),\end{aligned}$$

where $\mu_n(K)$ denotes the nth moment of the kernel. Clearly, using a zero-mean (i.e., symmetric or properly shifted) kernel eliminates the first term on the RHS, and indeed, letting $h \to 0$ eliminates bias altogether, but at the cost of increasing the noise in the estimate. Assuming a finite mean and proper shifting, the squared bias is simply (after truncation of higher-order terms)

$$\int B(x)^2dx \approx \frac{1}{4}(h^2\mu_2(K))^2\int(f'')^2dx.$$

Silverman (1986) uses the bias approximation, the substitution $z = (x-\xi)/h$, and another application of Taylor series to reduce the local variance term to

$$\begin{aligned}\mathrm{Var}[\overline{f}(x)] &\approx \frac{1}{nh}\int K(z)^2f(x-zh)dz - \frac{1}{n}\left(f(x)+\mathcal{O}(h^2)\right)^2\\ &\approx \frac{1}{nh}\int K(z)^2\left(f(x)-hzf'(x)+\mathcal{O}(h^2)\right)dz + \mathcal{O}(n^{-1})\\ &\approx \frac{1}{nh}f(x)\int K(z)^2dz,\end{aligned}$$

which, when integrated over x yields

$$\int\mathrm{Var}[\overline{f}(x)]dx \approx \frac{1}{nh}\int K(z)^2dz.$$

All told, this gives a MISE of

$$\mathrm{MISE} \approx \frac{1}{4}(h^2\mu_2(K))^2\int f''(x)^2dx + \frac{1}{nh}\int K(z)^2dz,$$

based on the assumptions of small h, large n, and $n \gg 1/h^2$, all of which are likely to be bad assumptions in practice. Taking $d(\mathrm{MISE})/dh$ and setting this expression to zero clearly gives the global estimate Eq. (5) in one-dimension.

Appendix B. Global bandwidth estimation using Fourier methods

In this section, we implement a method based on the Fourier transform that will allow us to create an unbiased estimator for the MISE and minimize this function in order to select the optimal bandwidth h. Throughout, we will use the form of the transform common to fast Fourier transform routines, namely

$$\hat{g}(\omega) = \int e^{-2\pi i\omega x} g(x) dx$$

for any sufficiently smooth function $g(x)$. Recall that the MISE can be written as the sum of a bias and variance term as in (12) so that

$$\text{MISE} = \int B(x)^2\, dx + \int \text{Var}[\overline{f}(x)]\, dx \tag{13}$$

where the bias is

$$B(x) = \int K(z) f(x - hz)\, dz - f(x)$$

and the variance is given by

$$\text{Var}[\overline{f}(x)] = \frac{1}{n}\left[\int \frac{1}{h^2} K\Big(\frac{x-z}{h}\Big)^2 f(z)\, dz - \left(\int \frac{1}{h} K\Big(\frac{x-z}{h}\Big) f(z)\, dz\right)^2\right].$$

Using Fourier methods, we first compute the bias term. Taking the transform of the bias and making the change of variables $y = x - hz$, we find

$$\begin{aligned}
\hat{B}(\omega) &= \int\int K(z) f(x - hz) e^{-2\pi i x\omega}\, dz dx - \hat{f}(\omega) \\
&= \int\int K(z) f(y) e^{-2\pi i\omega(y+hz)}\, dy dz - \hat{f}(\omega) \\
&= \left(\int K(z) e^{-2\pi i\omega hz}\, dz - 1\right)\hat{f}(\omega) \\
&= \big(\hat{K}(h\omega) - 1\big)\hat{f}(\omega).
\end{aligned}$$

Therefore, by Plancherel's Theorem, the bias term in Eq. (13) becomes

$$\int B(x)^2\, dx = \int \hat{B}(\omega)^2\, d\omega = \int \big(\hat{K}(h\omega) - 1\big)^2 \hat{f}(\omega)^2\, d\omega.$$

To compute the associated variance term in Eq. (13), we first split it into two parts so that

$$\int \text{Var}[\overline{f}(x)] dx = \frac{1}{n}(I - II).$$

The first term is then

$$I = \frac{1}{h^2}\int\int K\Big(\frac{x-z}{h}\Big)^2 f(z)\, dz dx$$

and satisfies

$$\begin{aligned}
I &= \frac{1}{h}\int\int K(y)^2 f(x - hy)\, dy dx \\
&= \frac{1}{h}\int K(y)^2 \left(\int f(x - hy) dx\right) dy \\
&= \frac{1}{h}\left(\int K(y)^2\, dy\right)\left(\int f(\xi) d\xi\right) \\
&= \frac{1}{h}\int K(y)^2\, dy
\end{aligned}$$

due to the change of variables $y = (x - z)/h$ and then $\xi = x - hy$, as well as the fact that $f(x)$ is a pdf. To compute II, we write it as

$$II = \int P(x)^2\, dx$$

where

$$P(x) = \frac{1}{h}\int K\Big(\frac{x-z}{h}\Big) f(z)\, dz = \int K(y) f(x - hy)\, dy.$$

Of course, the transform of $P(x)$ has already been identified in the computation of the integrated bias term. In particular, it is given by

$$\hat{P}(\omega) = \hat{K}(h\omega)\hat{f}(\omega).$$

Using Plancherel's theorem as before, we find

$$II = \int \hat{P}(\omega)^2\, d\omega = \int \hat{K}(h\omega)^2 \hat{f}(\omega)^2\, d\omega.$$

With this Fourier representation of the bias and variance integrals, we may explicitly write the MISE in terms of integrals of transformed functions, namely

$$\text{MISE}_n(h) = \frac{1}{nh}\int \hat{K}(\omega)^2\, d\omega + \int \left((\hat{K}(h\omega) - 1)^2 - \frac{1}{n}\hat{K}(h\omega)^2\right)\hat{f}(\omega)^2\, d\omega. \tag{14}$$

Note that we have used $\int K(y)^2\, dy = \int \hat{K}(\omega)^2\, d\omega$ in the first term to write the MISE depending upon $\hat{K}$ rather than K. This derivation is similar to previous spectral representations of the MISE (Chiu, 1991; Wu et al., 2007; Wu and Tsai, 2004).

Unfortunately, this expression still requires knowledge of the Fourier transform, $\hat{f}(\omega)$, of the unknown pdf and thus cannot be used to choose the optimal bandwidth h. Instead, we will rely on an empirical distribution to approximate f, and thus $\hat{f}$. Given n observations of the distribution $f(x)$, which are denoted $X_1, \ldots, X_n$, we define the empirical (or observed) distribution

$$f_n(x) = \frac{1}{n}\sum_{j=1}^{n} \delta(x - X_j)$$

so that the corresponding transform of this function is

$$\hat{f}_n(\omega) = \frac{1}{n}\sum_{j=1}^{n}\int e^{-2\pi i x\omega}\delta(x - X_j)\, dx = \frac{1}{n}\sum_{j=1}^{n} e^{-2\pi i\omega X_j}. \tag{15}$$

Now, as $n \to \infty$, we find $f_n \to f$ and $\hat{f}_n \to \hat{f}$. In fact, we have an asymptotic estimate for the expected value of $\hat{f}_n$, which implies

$$\mathbb{E}\big[\hat{f}_n(\omega)^2\big] \approx \left(1 - \frac{1}{n}\right)\hat{f}(\omega)^2 + \frac{1}{n} \tag{16}$$

as $n \to \infty$.

Therefore, by using the empirical distribution, we can define and utilize an unbiased estimator for the MISE. For fixed $n \in \mathbb{N}$ and any $h \geq 0$, define

$$\begin{aligned}
\varepsilon_n(h) &= \frac{2}{n}\int K(h\omega) d\omega + \int \left[\left(1 - \frac{1}{n}\right)\hat{K}(h\omega)^2 - 2\hat{K}(h\omega)\right]\hat{f}_n(\omega)^2\, d\omega \\
&= \frac{2}{nh}K(0) + \int \left[\left(1 - \frac{1}{n}\right)\hat{K}(h\omega)^2 - 2\hat{K}(h\omega)\right]\hat{f}_n(\omega)^2\, d\omega.
\end{aligned} \tag{17}$$

Then, $\varepsilon_n(h)$ and $\text{MISE}_n(h)$ must attain their minimum values at the same h. Therefore, given a sample $X_1, \ldots, X_n$ of n draws from $f(x)$, we define the optimal bandwidth by

$$h_\varepsilon = \underset{h \geq 0}{\arg\min}\ \varepsilon_n(h).$$

Computationally approximating the global bandwidth using this value of h_ε is instrumental to the algorithm proposed in Section 2.

Finally, we justify the claim that $\varepsilon_n(h)$ is an unbiased estimator of the MISE. We first note that by the Fourier inversion property we have

$$\frac{1}{h}K(0) = \frac{1}{h}\int \hat{K}(\omega)e^0\, d\omega = \int \hat{K}(h\omega)\, d\omega.$$

Then, taking the expectation of $\varepsilon_n(h)$ and inserting the convergence result (16), we find

$$\begin{aligned}
\mathbb{E}\big[\varepsilon_n(h)\big] &= \frac{2}{n}\int \hat{K}(h\omega) d\omega + \int \left[\left(1 - \frac{1}{n}\right)\hat{K}(h\omega)^2 - 2\hat{K}(h\omega)\right]\mathbb{E}\big[\hat{f}_n(\omega)^2\big]\, d\omega \\
&= \frac{2}{n}\int \hat{K}(h\omega) d\omega \\
&\quad + \int \left[\left(1 - \frac{1}{n}\right)\hat{K}(h\omega)^2 - 2\hat{K}(h\omega)\right]\left(\left(1 - \frac{1}{n}\right)\hat{f}(\omega)^2 + \frac{1}{n}\right) d\omega
\end{aligned}$$

$$
\begin{aligned}
&= \left(1-\frac{1}{n}\right)\int \hat{K}(h\omega)^2\left(\left(1-\frac{1}{n}\right)\hat{f}(\omega)^2+\frac{1}{n}\right)d\omega \\
&\quad -2\left(1-\frac{1}{n}\right)\int \hat{K}(h\omega)\hat{f}(\omega)^2\,d\omega \\
&= \left(1-\frac{1}{n}\right)\int \left[\hat{K}(h\omega)^2\left(\left(1-\frac{1}{n}\right)\hat{f}(\omega)^2+\frac{1}{n}\right)-2\hat{K}(h\omega)\hat{f}(\omega)^2\right]d\omega \\
&= \left(1-\frac{1}{n}\right)\int \left[\left(\hat{K}(h\omega)^2-2\hat{K}(h\omega)-\frac{1}{n}\hat{K}(h\omega)^2\right)\hat{f}(\omega)^2\right. \\
&\quad \left.+\frac{1}{n}\hat{K}(h\omega)^2\right]d\omega \\
&= \left(1-\frac{1}{n}\right)\left[\int \left((\hat{K}(h\omega)-1)^2-\frac{1}{n}\hat{K}(h\omega)^2-1\right)\hat{f}(\omega)^2 d\omega\right. \\
&\quad \left.+\frac{1}{nh}\int \hat{K}(\omega)^2\,d\omega\right] \\
&= \left(1-\frac{1}{n}\right)\left(\mathrm{MISE}_n(h)-\int \hat{f}(\omega)^2\,d\omega\right) \\
&= \left(1-\frac{1}{n}\right)\left(\mathrm{MISE}_n(h)-\int f(x)^2\,dx\right).
\end{aligned}
$$

This implies that, modulo a shifting and scaling factor that are both independent of h, the expectation of our estimator is exactly $\mathrm{MISE}_n(h)$. Additionally, it becomes clear that this function must attain its minimum at the same value of h as $\mathrm{MISE}_n(h)$, and modulo a shift we have $\mathbb{E}\left[\varepsilon_n(h)\right] \sim \mathrm{MISE}_n(h)$ as $n \to \infty$.

Appendix C. Numerical bandwidth estimation

Next, we outline a numerical approach based on our use of the Fourier transform. Implementing the iterative algorithm of Section 3 to compute the approximate distributon, let us assume that the algorithm converges. Then, due to the relationship between successive iterates and the previous kernel, namely $\widetilde{f}_{\ell+1}(x)$ and $\widetilde{f}_{\ell}(x) = K_{\ell}(x)$, the final density and the kernel must converge to the same function as $\ell \to \infty$, while the bandwidth must also converge to some value $h_\infty > 0$. Then, denoting the converged iteratively-estimated kernel (based on data) by $K_\infty(x)$, this function must satisfy the interesting self-similar property (using Eq. (1))

$$
K_\infty(x) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{h_\infty}K_\infty\left(\frac{x-X_j}{h_\infty}\right).
$$

Additionally, its Fourier transform then satisfies the relationship

$$
\begin{aligned}
\hat{K}_\infty(\omega) &= \frac{1}{nh_\infty}\int e^{-2\pi i\omega x}\sum_{j=1}^{n}K_\infty\left(\frac{x-X_j}{h_\infty}\right)dx \\
&= \frac{1}{nh_\infty}\int \sum_{j=1}^{n} e^{-2\pi i\omega(zh_\infty+X_j)}K_\infty(z)h_\infty dz \\
&= \frac{1}{n}\sum_{j=1}^{n} e^{-2\pi i\omega X_j}\int e^{-2\pi i\omega z h_\infty}K_\infty(z)dz \\
&= \frac{1}{n}\sum_{j=1}^{n} e^{-2\pi i\omega X_j}\hat{K}_\infty(h_\infty\omega) \\
&= \hat{f}_n(\omega)\hat{K}_\infty(h_\infty\omega)
\end{aligned}
$$

due to (15). With this and the asymptotic approximation (16), we have

$$
\hat{K}_\infty(\omega) = \hat{f}(\omega)\hat{K}_\infty(h_\infty\omega)+\mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \tag{18}
$$

for n suitably large. Evaluating the MISE (14) with the approximation $K(x) = K_\infty(x)$ and $h = h_\infty$ yields

$$
\mathrm{MISE}_n(h_\infty) = \int \left(\hat{K}_\infty(h_\infty\omega)-1\right)^2\hat{f}(\omega)^2\,d\omega+\mathcal{O}\left(\frac{1}{n}\right)
$$

for n suitably large. Finally, expanding this expression and using the relationship (18) satisfied by the Fourier transforms of the limiting kernel and unknown pdf, we find

$$
\begin{aligned}
\mathrm{MISE}_n(h_\infty) &= \int \left(\hat{K}_\infty(h_\infty\omega)^2\hat{f}(\omega)^2-2\hat{K}_\infty(h_\infty\omega)\hat{f}(\omega)^2+\hat{f}(\omega)^2\right)d\omega \\
&\quad +\mathcal{O}\left(\frac{1}{n}\right) \\
&= \int \left(\hat{K}_\infty(\omega)^2-2\hat{K}_\infty(\omega)\hat{f}(\omega)+\hat{f}(\omega)^2\right)d\omega+\mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\
&= \int \left|\hat{K}_\infty(\omega)-\hat{f}(\omega)\right|^2 d\omega+\mathcal{O}\left(\frac{1}{\sqrt{n}}\right).
\end{aligned}
$$

Therefore, we see that for large n the iterative algorithm guarantees that the MISE is minimized precisely when the kernel $K(x)$ converges to the unknown distribution in the L^2 sense. This suggests that when the algorithm converges (as $\ell \to \infty$) it must converge to the unknown pdf $f(x)$ because an unbiased estimator for the MISE is minimized at every step.

Furthermore, our analysis now demonstrates the appropriate range of Taylor series estimates of h, because the exact result can be derived from the Fourier transform. Assume a Gaussian for the kernel and also assume *a priori* that the underlying data are Gaussian (with zero mean and variance σ^2), so that $\hat{K}(\omega) = \exp(-(2\pi\omega)^2/2)$ and $\hat{f}(\omega) = \exp(-\sigma^2(2\pi\omega)^2/2)$. The first integral in Eq. (14) can be computed in several ways, but is easily performed by recognizing the form of a Gaussian, so that

$$
\frac{1}{nh}\int e^{-(2\pi\omega)^2}d\omega = \frac{1}{nh}\frac{1}{\sqrt{4\pi}}\int \frac{1}{\sqrt{2\pi/(8\pi^2)}}e^{\left(\frac{-\omega^2}{2/(8\pi^2)}\right)}d\omega = \frac{1}{2\sqrt{\pi}nh},
$$

owing to the fact that the last integral is of a density in ω. Similarly, the second integral in Eq. (14) is

$$
\begin{aligned}
&\int \left((1-1/n)\hat{K}^2(h\omega)-2\hat{K}(h\omega)\right)\hat{f}^2(\omega)d\omega \\
&= \int \left((1-1/n)e^{-(2\pi h\omega)^2}-2e^{-(2\pi h\omega)^2/2}\right)e^{-\sigma^2(2\pi\omega)^2}d\omega \\
&= \int \left((1-1/n)e^{-(\sigma^2+h^2)(2\pi\omega)^2}-2e^{-(\sigma^2+h^2/2)(2\pi\omega)^2}\right)d\omega \\
&= \frac{(1-1/n)}{2\sqrt{\pi(h^2+\sigma^2)}}-\frac{1}{\sqrt{\pi(\sigma^2+h^2/2)}}
\end{aligned}
$$

where we have rearranged as before to make Gaussian densities (in ω) for each term. Therefore, the resulting quantity to be minimized (Silverman, 1986) is now

$$
\mathrm{MISE}_n(h) = \frac{1}{2\sqrt{\pi}nh}+\frac{1}{2\sqrt{\pi(\sigma^2+h^2)}}-\frac{1}{2n\sqrt{\pi(\sigma^2+h^2)}}-\frac{1}{\sqrt{\pi(\sigma^2+h^2/2)}}. \tag{19}
$$

It suffices to approximate the h_0 that minimizes $\mathrm{MISE}_n(h)$ to any numerical tolerance, by taking $d(\mathrm{MISE}_n(h))/dh$, setting it to zero, and finding the root of the resulting equation. As expected, the estimate of h_0 based on Taylor series is worse for smaller data sets (i.e., $n \lesssim 100$), but as n grows large, the Taylor series solution converges to the exact solution (Fig. 8). However, it is important to note that these quantities are the optimal bandwidth when both the kernel *and the underlying data density* are known to be Gaussian. If the underlying density is unknown, then the data are used to construct the quantity to be minimized $\varepsilon_n(h)$ in Eq. (17). To see how this differs, we can imagine that perfectly Gaussian data is generated. Then Eq. (17) evaluates to

$$
\varepsilon_n(h) = \frac{2}{\sqrt{2\pi}nh}+\frac{1}{2\sqrt{\pi(\sigma^2+h^2)}}-\frac{1}{2n\sqrt{\pi(\sigma^2+h^2)}}-\frac{1}{\sqrt{\pi(\sigma^2+h^2/2)}}, \tag{20}
$$

which has a root approximately $4^{1/5} = 1.32$ larger for large n (Fig. 8). The fact that data are imperfect means that the global bandwidth must be about 32% to 70% larger (depending on n) to achieve additional smoothing when compared to a completely "perfect" realization of data.

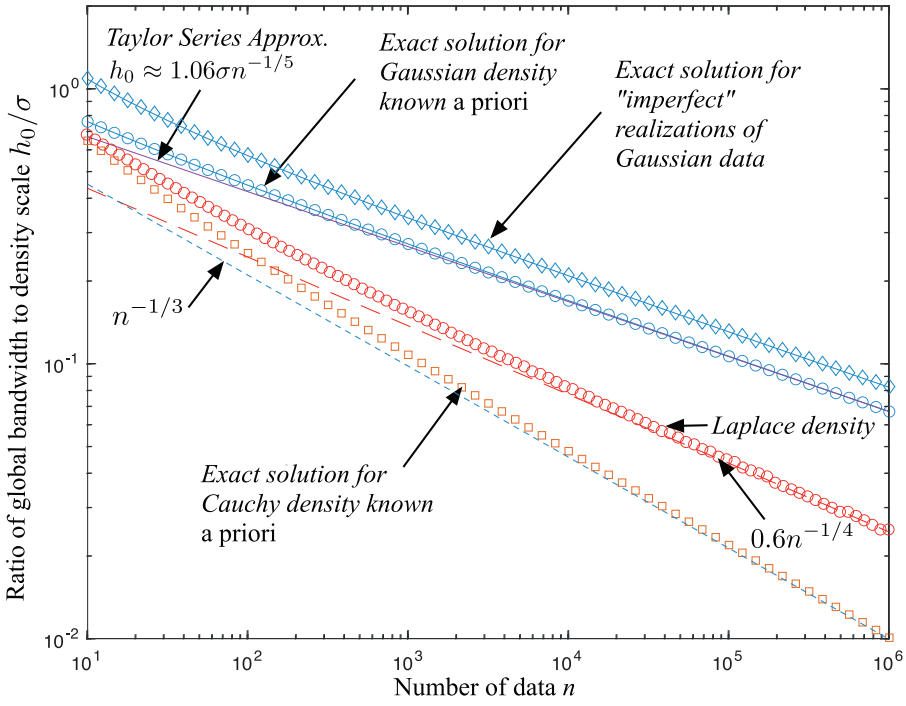

Fig. 8. Log-log plots of global bandwidth over data scale parameter (h_0/σ) versus number n of data using exact solution for Gaussian data density known a priori (Eq. (20)) versus Taylor series approximate solution (Eq. (6)) and numerical estimation of Gaussian data density (Eq. (17)). Also shown are the lower values of h_0/σ for Cauchy data estimated with Cauchy kernel.

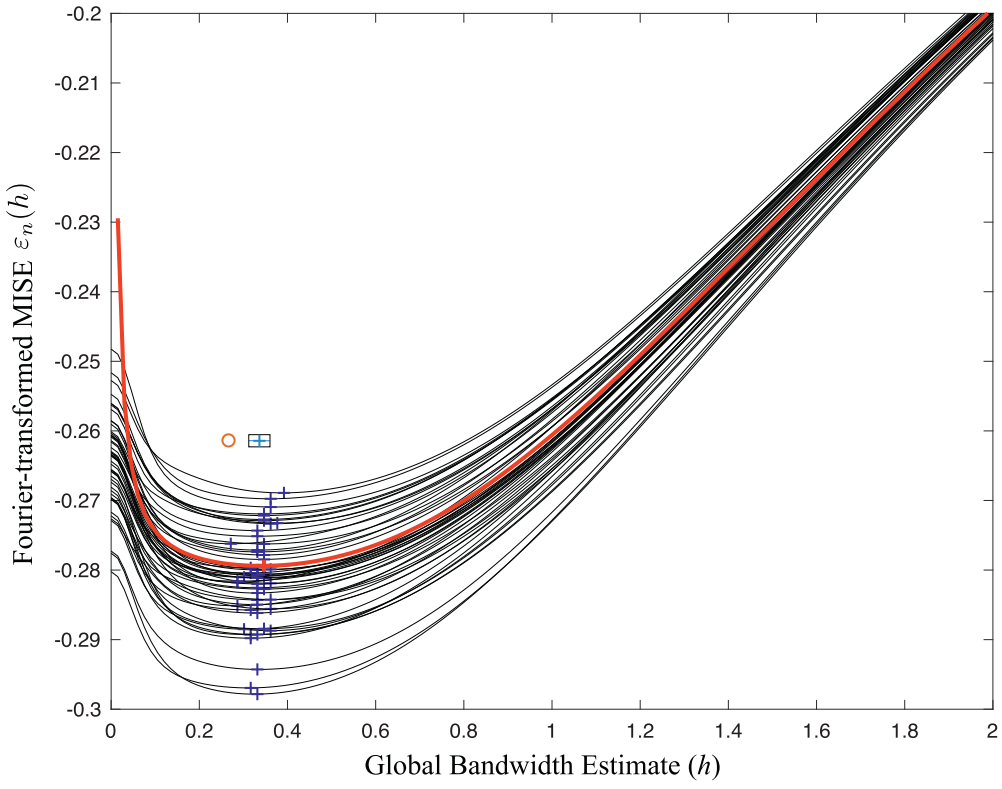

Fig. 9. Plots of unbiased estimator for Fourier-transformed MISE (denoted $\varepsilon_n(h)$) as a function of global bandwidth parameter h. Plots either assume or use Gaussian data with $\sigma^2 = 1$ and 1000 data points. The exact expression Eq. (20) is plotted with a thick red curve; the minimum (shown with a + sign) is found at $h_0 = 0.3406$. The estimate of h_0 using Taylor series is 0.266 and is denoted by a circle above the curves. Also shown is an ensemble of 50 curves (in black) wherein for each curve 1000 IID Gaussian data are generated and the density function is estimated by Fourier transform Eq. (15). The ensemble statistics of the estimated h_0 were calculated with mean $h_0 = 0.335$, with standard deviation of $\sigma_{h_0} = 0.0238$ (box above curves denotes mean $\pm\sigma_{h_0}$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For a specific example, we set $n = 1000$ and $\sigma^2 = 1$, which produces an exact optimal global bandwidth (for imperfect data using $\varepsilon_n(h)$ in Eq. (17)) of $h_0 = 0.341$ (Fig. 9), whereas the estimate based on Taylor expansions gives $h_0 = 1.06\sigma n^{-1/5} = 0.266$. It is important to see how well a numerical estimate of the data density gives an estimate of h_0, rather than simply assuming a Gaussian density function. We may now compare the values of h that are estimated using the Fourier-transformed data to form an estimate of the density function (i.e., using Eq. (15) in Eq. (17)) instead of assuming the Gaussian form. Here we show the results for 50 independent runs in which 1000 IID Gaussian data are generated and the experimental curve generated and h taken at the curve minimum (black curves in Fig. 9). While there is a large vertical spread in the curves, the locations of the minima are fairly tightly constrained. The mean of 50 values of h_0 is 0.335 (compared to the exact value of 0.341), and the estimated h_0 have a standard deviation of 0.0238.

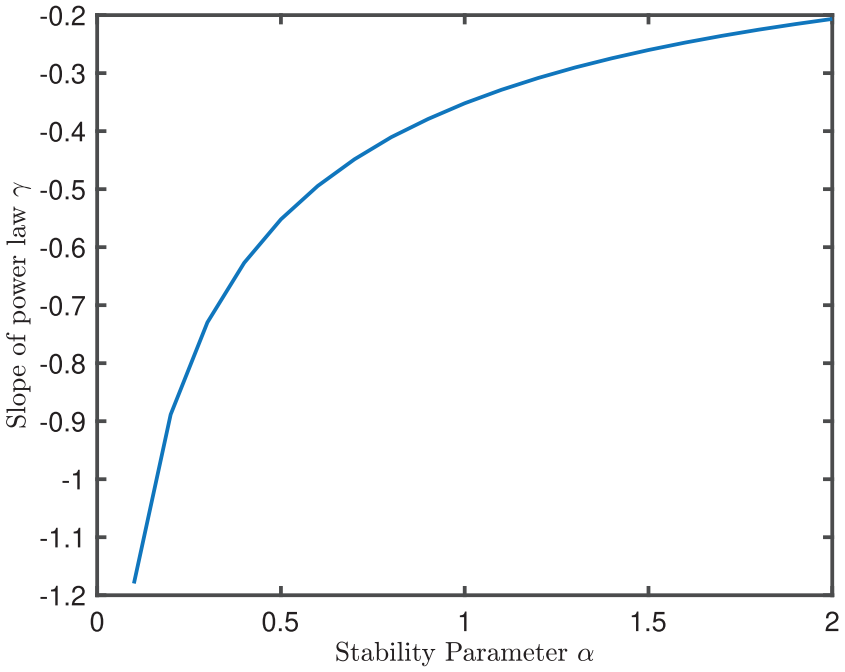

Fig. 10. Slope of power law decline $\sim n^{-\gamma}$ of optimal kernel bandwidth with particle number for stable distributions with different stability parameter α

Several other characteristic functions (Fourier transforms of PDFs) are easily integrated and illustrate the effect of data distribution on estimation of h_0. For example, the standard Cauchy density, defined by

$$f(x) = \frac{\sigma}{\pi\left(\sigma^2 + x^2\right)},$$

has both divergent variance and mean, and its associated Fourier transform is given by

$$\hat{f}(\omega) = \exp(-2\pi\sigma|\omega|).$$

Note that the scale parameter σ commonly used for stable densities is not the standard deviation, which is infinite. Assuming that the kernel was also a perfect copy of the data density, so that

$$\hat{K}(\omega) = \exp(-2\pi|\omega|),$$

the MISE becomes (up to an additive factor independent of h)

$$\mathrm{MISE}_n(h) = \frac{1}{2\pi nh} + \left(1 - \frac{1}{n}\right)\frac{1}{2\pi(h+\sigma)} - \frac{1}{\pi\left(\frac{1}{2}h + \sigma\right)}.$$

Therefore, calculating the minimum of the MISE (14) means solving the root of

$$\frac{d(\mathrm{MISE}_n)}{dh} = -\frac{1}{2\pi n} - \left(1 - \frac{1}{n}\right)\frac{1}{2\pi\left(1 + \frac{\sigma}{h}\right)^2} + \frac{2}{\pi\left(1 + \frac{2\sigma}{h}\right)^2}. \tag{21}$$

These values of $h_0(n)$ are significantly smaller than those found for Gaussian data (Fig. 8) and also decline for large n approximately like $\sim n^{-1/3}$. This suggests a numerical procedure for simultaneous estimation of the data density and the global bandwidth. The FT estimate of h_0 based on Gaussian data is the largest of the estimates (Fig. 8), so we begin with that value. If the iterated kernel—based on the estimated density and using this h_0—fails to converge, then we reduce h_0 systematically down to a minimum given by the Cauchy h_0. In this procedure, the specifics of the data distribution need not be known. Simply start with an assumption of Gaussian-like smoothness and data density, but allow for Cauchy-like sparcity of data (i.e., few very large data).

We may also consider the Laplace (or double exponential) density defined by

$$f(x) = \frac{1}{2\sigma}\exp\left(-\frac{|x|}{\sigma}\right),$$

which has mean zero and variance $2\sigma^2$, but does not possess a continuous derivative at $x = 0$. The Fourier transform of this function is given by

$$\hat{f}(\omega) = \frac{1}{1 + 4\pi^2\sigma^2\omega^2}.$$

If the kernel is similarly distributed so that

$$\hat{K}(\omega) = \frac{1}{1 + 4\pi^2\omega^2},$$

then the MISE becomes

$$\mathrm{MISE}_n(h) = \frac{1}{4nh} + \left(1 - \frac{1}{n}\right)\frac{h^2 + 3h\sigma + \sigma^2}{4(h+\sigma)^3} - \frac{2h+\sigma}{2(h+\sigma)^2}.$$

As before, the minimum of the MISE (14) can be computed by finding the root of the derivative of this expression, namely

$$\frac{d(\mathrm{MISE}_n)}{dh} = -\frac{1}{4n}\left(1 + \frac{\sigma}{h}\right)^2 + \left(\frac{3}{4} + \frac{1}{4n}\right)\frac{\frac{h}{\sigma}}{\frac{h}{\sigma} + 1} - \left(\frac{3}{4} - \frac{1}{4n}\right)\frac{\frac{h}{\sigma}}{\left(\frac{h}{\sigma} + 1\right)^2}. \tag{22}$$

The resulting values of $h_0(n)$ are again significantly smaller than those found for Gaussian data (Fig. 8) and also decline for large n approximately like $\sim n^{-1/4}$.

Finally, we consider the family of stable distributions, whose density may be defined by

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-|ck|^\alpha(1 - i\beta sgn(k)\tan(\frac{\pi\alpha}{2}))}e^{-ikx}dk$$

where $0 < \alpha \leq 2$ is the stability parameter, $-1 < \beta < 1$ is a skewness parameter and c is the scale parameter. The Fourier transform is given by

$$\hat{f}(\omega) = e^{-|2\pi c\omega|^\alpha(1 - i\beta sgn(-\omega)\tan(\frac{\pi\alpha}{2}))}.$$

As before, if the kernel is similarly distributed so that

$$\hat{K}(\omega) = e^{-|2\pi\omega|^\alpha(1 - i\beta sgn(-\omega)\tan(\frac{\pi\alpha}{2}))}$$

then the MISE becomes

$$\mathrm{MISE}_n(h) = C\left[\frac{2^{-1/\alpha}}{nh} + \left(1 - \frac{1}{n}\right)\left(2h^\alpha + 2c^\alpha\right)^{-1/\alpha} - 2\left(h^\alpha + 2c^\alpha\right)^{-1/\alpha}\right]$$

where

$$C = \int_{-\infty}^{\infty} e^{-|2\pi\omega|^{\alpha}(1-i\beta sgn(-\omega)\tan(\frac{\pi\alpha}{2}))} d\omega$$

As before this can be minimized and, similar to the distributions explored so far, we find that there is a power law decline $\sim n^{-\gamma}$ where γ depends on α as depicted in Fig. 10. Note that the magnitude of MISE depends on α and β through the constant C, but that this does not impact the minimized value. We also note that these calculations may be made for other densities but are not shown. Additionally, some of the integrations must be performed numerically as it may be the case that no closed-form expression for the antiderivative exists.

References

Abramson, I.S., 1982. On bandwidth variation in kernel estimates–a square root law. Ann. Stat. 10 (4), 1217—1223.

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Autom. Control AC-19 (6), 716–723.

Benson, D.A., Meerschaert, M.M., Revielle, J.R., 2013. Fractional calculus in hydrologic modeling: a numerical perspective. Adv. Water Resour. 51, 479–497. https://doi.org/10.1016/j.advwatres.2012.04.005. 35th Year Anniversary Issue

Benson, D.A., Pankavich, S., Bolster, D., 2019. On the separate treatment of mixing and spreading by the reactive-particle-tracking algorithm: an example of accurate upscaling of reactive Poiseuille flow. Adv. Water Resour. 123, 40–53. https://doi.org/10.1016/j.advwatres.2018.11.001.

Benson, D.A., Pankavich, S., Schmidt, M.J., Sole-Mari, G., 2020. Entropy: (1) the former trouble with particle-tracking simulation, and (2) a measure of computational information penalty. Adv. Water Resour. 137, 103509. https://doi.org/10.1016/j.advwatres.2020.103509.

Brockwell, P.J., Davis, R.A., 2016. Introduction to Time Series and Forecasting. Springer Texts in Statistics, third ed. Springer.

Carrel, M., Morales, V.L., Dentz, M., Derlon, N., Morgenroth, E., Holzner, M., 2018. Pore-scale hydrodynamics in a progressively bioclogged three-dimensional porous medium: 3-D particle tracking experiments and stochastic transport modeling. Water Resour. Res. 54 (3), 2183–2198. https://doi.org/10.1002/2017WR021726.

Chakraborty, P., Meerschaert, M.M., Lim, C.Y., 2009. Parameter estimation for fractional transport: a particle-tracking approach. Water Resour. Res. 45 (10). https://doi.org/10.1029/2008WR007577.

Chiu, S.-T., 1991. Bandwidth selection for kernel density estimation. Ann. Stat. 19 (4), 1883–1905. https://doi.org/10.1214/aos/1176348376.

Ding, D., Benson, D., Paster, A., Bolster, D., 2012. Modeling bimolecular reactions and transport in porous media via particle tracking. Adv. Water Resour. 53, 56–65. https://doi.org/10.1016/j.advwatres.2012.11.001.

Ding, D., Benson, D.A., Fernández-Garcia, D., Henri, C.V., Hyndman, D.W., Phanikumar, M.S., Bolster, D., 2017. Elimination of the reaction rate “scale effect”: Application of the Lagrangian reactive particle-tracking method to simulate mixing-limited, field-scale biodegradation at the Schoolcraft (MI, USA) site. Water Resour. Res. https://doi.org/10.1002/2017WR021103.

Engdahl, N.B., Benson, D.A., Bolster, D., 2017. Lagrangian simulation of mixing and reactions in complex geochemical systems. Water Resour. Res. 53 (4), 3513–3522. https://doi.org/10.1002/2017WR020362.

Engdahl, N.B., Schmidt, M.J., Benson, D.A., 2019. Accelerating and parallelizing Lagrangian simulations of mixing-limited reactive transport. Water Resour. Res. 55 (4), 3556–3566. https://doi.org/10.1029/2018WR024361.

Engel, J., Herrmann, E., Gasser, T., 1994. An iterative bandwidth selector for kernel estimation of densities and their derivatives. J. Nonparametric Stat. 4 (1), 21–34. https://doi.org/10.1080/10485259408832598.

Fernàndez-Garcia, D., Sanchez-Vila, X., 2011. Optimal reconstruction of concentrations, gradients and reaction rates from particle distributions. J. Contam. Hydrol. 120-121, 99–114. https://doi.org/10.1016/j.jconhyd.2010.05.001. Reactive Transport in the Subsurface: Mixing, Spreading and Reaction in Heterogeneous Media.

Hirukawa, M., 2018. Asymmetric Kernel Smoothing: Theory and Applications in Economics and Finance. SpringerBriefs in Statistics. Springer.

Kang, P.K., Dentz, M., Le Borgne, T., Lee, S., Juanes, R., 2017. Anomalous transport in disordered fracture networks: spatial Markov model for dispersion with variable injection modes. Adv. Water Resour. 106, 80–94. https://doi.org/10.1016/j.advwatres.2017.03.024. Tribute to Professor Garrison Sposito: An Exceptional Hydrologist and Geochemist

Labolle, E.M., Fogg, G.E., Tompson, A.F.B., 1996. Random-walk simulation of transport in heterogeneous porous media: local mass-conservation problem and implementation methods. Water Resour. Res. 32 (3), 583–593.

Nolan, J. P., 2018. Stable Distributions: Models for Heavy Tailed Data. http://fs2.american.edu/jpnolan/www/stable/chap1.pdf.

Pedretti, D., Fernàndez-Garcia, D., 2013. An automatic locally-adaptive method to estimate heavily-tailed breakthrough curves from particle distributions. Adv. Water Resour. 59, 52–65. https://doi.org/10.1016/j.advwatres.2013.05.006.

Perez, L.J., Hidalgo, J.J., Dentz, M., 2019. Upscaling of mixing-limited bimolecular chemical reactions in poiseuille flow. Water Resour. Res. 55 (1), 249–269. https://doi.org/10.1029/2018WR022730.

Rizzo, C.B., Nakano, A., de Barros, F.P., 2019. PAR2: parallel random walk particle tracking method for solute transport in porous media. Comput. Phys. Commun. 239, 265–271. https://doi.org/10.1016/j.cpc.2019.01.013.

Samorodnitsky, G., Taqqu, M., 1994. Stable non-Gaussian Random Processes: Stochastic Models with Infinite Variance. Chapman and Hall, New York.

Schmidt, M.J., Pankavich, S., Benson, D.A., 2017. A kernel-based Lagrangian method for imperfectly-mixed chemical reactions. J. Comput. Phys. 336, 288–307. https://doi.org/10.1016/j.jcp.2017.02.012.

Schumer, R., Benson, D.A., Meerschaert, M.M., Baeumer, B., 2003. Fractal mobile/immobile solute transport. Water Resour. Res. 39, 1296.

Siirila-Woodburn, E.R., Fernàndez-Garcia, D., Sanchez-Vila, X., 2015. Improving the accuracy of risk prediction from particle-based breakthrough curves reconstructed with kernel density estimators. Water Resour. Res. 51 (6), 4574–4591. https://doi.org/10.1002/2014WR016394.

Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. Monographs on Statistics and Applied Probability. Chapman and Hall, London.

Sole-Mari, G., Fernàndez-Garcia, D., 2018. Lagrangian modeling of reactive transport in heterogeneous porous media with an automatic locally adaptive particle support volume. Water Resour. Res. 54 (10), 8309–8331. https://doi.org/10.1029/2018WR023033.

Sole-Mari, G., Fernández-Garcia, D., Rodríguez-Escales, P., Sanchez-Vila, X., 2017. A KDE-based random walk method for modeling reactive transport with complex kinetics in porous media. Water Resour. Res. 53 (11), 9019–9039. https://doi.org/10.1002/2017WR021064.

Sole-Mari, G., Schmidt, M.J., Pankavich, S.D., Benson, D.A., 2019. Numerical equivalence between SPH and probabilistic mass transfer methods for Lagrangian simulation of dispersion. Adv. Water Resour. 126, 108–115. https://doi.org/10.1016/j.advwatres.2019.02.009.

Taverniers, S., Bosma, S.B.M., Tartakovsky, D.M., 2020. Accelerated multilevel monte carlo with kernel-based smoothing and latinized stratification. Water Resour. Res. 56 (9). https://doi.org/10.1029/2019WR026984. e2019WR026984

Wu, T., Tsai, M., 2004. Root n bandwidths selectors in multivariate kernel density estimation. Probab. Theory Relat. Fields 129, 537–558. https://doi.org/10.1007/s00440-004-0357-8.

Wu, T.-J., Chen, C.-F., Chen, H.-Y., 2007. A variable bandwidth selector in multivariate kernel density estimation. Stat. Probab. Lett. 77 (4), 462–467. https://doi.org/10.1016/j.spl.2006.08.013.