https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions
Open Access
EGU

A diffusion-based kernel density estimator (diffKDE, version 1) with optimal bandwidth approximation for the analysis of data in geoscience and ecological research

Maria-Theresia Pelz[1,2], Markus Schartau[2], Christopher J. Somes[2], Vanessa Lampe[2], and Thomas Slawig[1]
[1]Department of Computer Science, Kiel University, 24118 Kiel, Germany
[2]GEOMAR Helmholtz Centre for Ocean Research Kiel, 24105 Kiel, Germany
Correspondence: mpelz@geomar.de

Abstract. Probability density functions (PDFs) comprise basic information about the variability of observed or simulated variables within a system of interest. In geoscience data distributions are often expressed by a parametric estimation of their PDF, such as e.g. a Gaussian distribution. At present there is a growing attention towards the analysis of non-parametric estimation of PDFs, where no prior assumptions about the type of PDF are required. A common tool for such non-parametric estimation is a kernel density estimator (KDE). Existing KDEs are valuable but incomplete, because of the difficulty of specifying optimal bandwidths for the individual kernels. A diffusion-based KDE provides a useful approach to mitigate the difficulty in identifying bandwidths that resolve desired details of multi-modal data while being insensitive to noise. Therefore we designed and developed a new implementation of a diffusion-based KDE as an open source Python tool. We tested our implementation on artificial and real marine biogeochemical data individually and against other popular KDEs. Our estimator is able to detect relevant multiple modes and resolve boundary close data while suppressing details induced by noise and individual outliers. The convergence rate is comparable to the Gaussian estimator, but with a generally smaller error, most notably for small data sets with up to around 5000 data points. We exemplify and discuss the general applicability of such KDEs for data-model comparison in geoscience, in particular for sparse data. We also provide an example for how our approach can be efficiently utilized for the derivation of plankton size spectra in ecological research.

1 Introduction

In geoscience the application of Earth system models (ESMs) has become an integral part of climate research (IPCC, 2022). Given the complexity of ESMs and the associated manifold of model solutions, there is strong demand for assessing the agreement of simulation results with observational data. In fact, such necessity is not only restricted to simulations with ESM but is transferable to other model applications as well, like in social science, in financial- or ecological research. A viable evaluation procedure is to compare probability density functions (PDFs) of the data with their simulated counterparts, which may also be quantified by some distance measure or divergence between respective PDFs (Thorarinsdottir et al., 2013). Along with the examination of the suitability of specific divergence functions for data-model assessment as done by Thorarinsdottir et al. (2013), a necessary prerequisite is the approximation of PDFs based on available data and model results.

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

Mathematically formulated, PDFs are integrable non-negative functions $f : \mathcal{A} \to [0, \infty]$ from a probability space $(\Omega, \mathcal{A}, P)$ into the non-negative real numbers. By definition, they allow to directly read the probability of the occurrence of a data value $X \in \mathbb{R}$ within a specific range $[a, b] \subseteq \mathbb{R}$ via the relationship

$$P(a < X < b) = \int_a^b f(x)\, dx \text{ for all } a < b \in \mathbb{R}. \tag{1}$$

The application of kernel density estimators (KDEs) has become a common approach for approximating PDFs in a *non-parametric* way (Parzen, 1962), which means that no probability parameters (like expectation or variance) of the data and no type of the underlying probability distribution (as, e.g., normal or log-normal) are prescribed or assumed. The general concept of KDEs takes into account information of every single data point and treats all of them equally. Consequently, every point's information weighs the same in the resulting estimate, without introducing additional assumptions.

A KDE is based on a kernel function and a smoothing parameter. The kernel function is ideally chosen to be a PDF itself. It is usually unimodal and centered around zero (Sheather, 2004). In the estimation process, the kernel function is sequentially centered around each data point. The sum of these individual kernels is standardized by the number of data points. This ensures that the final estimate is again a PDF by inheriting all properties of its kernels. The smoothing parameter, referred to as bandwidth, determines the smoothness of the estimate. If it is chosen to be small, more details of the underlying data structure become visible. If it is larger, more structure becomes smoothed out (Jones et al., 1996), and information from single data points might get lost. Hence, it is crucial to determine some kind of an optimal size of the bandwidth parameter to represent a suitable signal-to-noise ratio that allows a separation of significant distinctive features from ambiguous details. The question of optimal bandwidth selection is widely discussed in the literature (e.g., Sheather and Jones, 1991; Jones et al., 1996; Heidenreich et al., 2013). It also takes into account that there might not be one single "optimal" choice for such bandwidth (Abramson, 1982; Terrell and Scott, 1992; Chaudhuri and Marron, 2000; Scott, 2012).

The reformulation of the most common Gaussian KDEs (Sheather, 2004) into a diffusion equation provides a different view on KDE (Chaudhuri and Marron, 2000). This perspective change is possible, because the Gaussian kernel function solves the partial differential equation describing the diffusion heat process as the Green function. The time parameter of this differential equation corresponds to the smoothness of the estimate, and thus becomes tantamount to the estimate's bandwidth parameter (Chaudhuri and Marron, 2000). The initial value is typically set to include the δ-distribution of the input data. This differentiates the initial value problem from classical problems, since the δ-distribution is not a proper function itself. In specific applications this diffusion approach delivered convincing results (e.g., Botev et al., 2010; Deniz et al., 2011; Qin and Xiao, 2018). However, it tends to resolve too many details or overfit the data in others (e.g., Ma et al., 2019; Chaudhuri and Marron, 2000; Farmer and Jacobs, 2022). One main benefit of the diffusion KDE is that it provides a series of PDF estimates for a sequence of bandwidths by default (Chaudhuri and Marron, 2000). As a consequence, it offers the chance to choose between different grades of smoothness by design.

In this study, we present a new, modified diffusion-based KDE, for which we provide a Python implementation. Our aim is to retain the original idea of diffusion-based KDEs by Chaudhuri and Marron (2000) and Botev et al. (2010), but to avoid the

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions
Open Access
EGU

complex fixed-point iteration by Botev et al. (2010). The main objective of our refined approach is to achieve high performance for analyses of high variance and multimodal data sets. Our diffusion-based KDE is based on an iterative approximation that differs from others, using a default optimal bandwidth and two preliminary, so-called *pilot* estimates. This way the KDE can provide a family of estimates at different bandwidths to choose from in addition to a default solution optimally designed for data from geoscience and ecological research. Thus, an interactive investigation option of these different estimates becomes possible in an easy way.

This paper is structured as follows: At first, we will briefly recall the general concept of KDEs. Afterwards, our specific KDE approach will be introduced and described, as developed and implemented in a software package. We explain the two pilot estimation steps and the selection of the smoothing parameters. Then the performance of our refined estimator will be compared with other state-of-the-art KDEs, while considering known distributions and real marine biogeochemical data. The real test data include carbon isotope ratios of particulate organic matter ($\delta^{13}C_{POC}$) and plankton size data. Our analyses presented here involve investigations of KDE error, runtime, the sensitivity to data noise and the characteristics of convergence w.r.t. increasing sample size.

2 Theory and methods

2.1 Kernel density estimation

A kernel density estimator (KDE) is a non-parametric statistical tool for the estimation of probability density functions (PDFs). In practice, diverse specifications of KDEs exist that may improve the performance with respect to individual needs. Before we explain our specifications of the diffusion-based KDE, we will provide basic background information about KDEs.

For all following let $\Omega \subseteq \mathbb{R}$ be a domain, $X_j \in \Omega$, $j \in \{1, ..., N\}$, be $N \in \mathbb{N}$ independent real random variables.

2.1.1 The general kernel density estimator

The most general form of a KDE approximates the true density f of the input data $(X_j)_{j=1}^N$ by

$$\hat{f} : \mathbb{R} \times \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}, \quad (x; h) \mapsto \frac{1}{n\,h} \sum_{j=1}^{n} K\left(\frac{x - X_j}{h}\right). \tag{2}$$

In this formula the *kernel function* $K : \mathbb{R} \to \mathbb{R}_{>0}$ has to satisfy the following conditions (Parzen, 1962):

$$\sup_{y \in \mathbb{R}} |K(y)| < \infty, \quad \int_{\mathbb{R}} |K(y)|\, dy < \infty, \quad \lim_{y \to \infty} |yK(y)| = 0, \quad \int_{\mathbb{R}} K(y) dy = 1. \tag{3}$$

The parameter h determines the smoothness of the estimate calculated by Eq. 2 and is called the *bandwidth parameter* (Silverman, 1986). In the following we will exclusively deal with the squared bandwidth (h^2) and therefore adapt a notation where some t is defined as $h^2 =: t \in \mathbb{R}$. An optimal choice for the bandwidth parameter is regarded as the minimizer of the asymptotic mean squared error between the true density of $(X_j)_{j=1}^N$ and their KDE (Sheather and Jones, 1991). The mean

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023


Geoscientific Model Development Discussions

integrated squared error is defined as

$$\mathbf{MISE}\left(\hat{f}\right) : \mathbb{R}_{>0} \to \mathbb{R}, t \mapsto \mathbf{E}\left(\int_{\mathbb{R}} \left(\hat{f}\left(x;t\right) - f\left(x\right)\right)^2 dx\right) \tag{4}$$

for all PDFs f and respective KDEs $\hat{f}$ (Scott, 1992). If now $\hat{f}$ is a KDE and there exists a $t^* \in \mathbb{R}_{>0}$ with

$$\mathbf{AMISE}\left(\hat{f}\right)(t^*) = \min_{t \in \mathbb{R}_{>0}} \mathbf{AMISE}\left(\hat{f}\right)(t), \tag{5}$$

we call t^* the *optimal bandwidth* of $\hat{f}$ by $(X_j)_{j=1}^N$ (Scott, 1992). For the general KDE from Equation 2, this can be calculated according to Parzen (1962) as

$$t^* = \left(\frac{f\left(x\right)\int K^2\left(y\right)dy}{N4\left\|f^2\right\|^2}\right)^{\frac{2}{5}}. \tag{6}$$

As we see from Eq. 6, the true density f is involved in the calculation of the optimal bandwidth t^*, which is in turn needed for the approximation of f by a KDE. Thus, a direct derivation of an optimal bandwidth is precluded. One possibility of how this implicit relation can be solved is the calculation of pilot estimation steps. Our specific approach to this is shown in Sec. 2.1.3 and Sec. 2.1.4.

There exists a variety of available choices for the type of kernel function K, which all have their individual benefits and shortcomings. Amongst them are for example the uniform, triangle, or the Epanechnikov kernel (Scott, 1992):

$$K_E : \mathbb{R} \to \mathbb{R}_{\geq 0}, w \mapsto \frac{3}{4}\left(1 - w^2\right). \tag{7}$$

A common choice for K is the Gaussian kernel (Sheather, 2004):

$$\Phi : \mathbb{R} \to \mathbb{R}_{\geq 0}, w \mapsto \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}w^2}. \tag{8}$$

The standard KDE from Eq. 2 – despite being widely applied and investigated – comes with several disadvantages in practical applications (Khorramdel et al., 2018). For example, severe boundary bias can occur when applied on a compact interval (Marron and Ruppert, 1994). It means that a kernel function with a specified bandwidth, attributed to a single point nearby the boundary, may actually exceed the boundary. Furthermore, it can lack a proper response to variations in the magnitude of the true density f (Breiman et al., 1977). The introduction of a parameter that depends on the respective data region can address the latter (Breiman et al., 1977). Unfortunately, no true independent local bandwidth strategy exists (Terrell and Scott, 1992), meaning that in all local approaches there is still an influence of neighboring data points on each locally chosen bandwidth.

2.1.2 The diffusion-based kernel density estimator

The diffusion-based KDE provides a different approach to Eq. 2. It solves the partial differential equation describing the diffusion heat process, starting from an initial value based on the input data $(X_j)_{j=1}^N$, that progresses up to an estimate at a final time $T \in \mathbb{R}_{>0}$. An advantageous connection to Eq. 2 is that the widely applied Gaussian kernel is a fundamental solution of this

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023


Geoscientific Model Development Discussions
Open Access
EGU

differential equation. Precisely, the Gaussian kernel from Eq. 8 as applied in the construction of a Gaussian KDE depends on the location $x \in \mathbb{R}$ and the smoothing parameter $h \in \mathbb{R}_{>0}$ and has the form $(x;t) \mapsto \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-X_j}{\sqrt{t}}\right)^2}$ for any $j \in \{1,...,N\}$. This function solves

$$\frac{\partial}{\partial t} u(x;t) = \frac{1}{2} \frac{d^2}{dx^2} u(x;t), x \in \Omega, t \in \mathbb{R}_{>0} \tag{9}$$

as the Green's function, where the time parameter $t \in \mathbb{R}_{>0}$ equals the squared bandwidth parameter h^2 (Chaudhuri and Marron, 2000). This idea to use the diffusion heat equation to calculate a KDE was first proposed by Chaudhuri and Marron (2000). Its benefits were widely explored in Botev et al. (2010).

Our implementation of the diffusion KDE is based on Chaudhuri and Marron (2000) and is extended by some advancements proposed by Botev et al. (2010): We included a parameter function $p \in C^2(\Omega, \mathbb{R}_{>0})$ with $\|p''\|_\infty < \infty$ into Eq. 9, acting inversely to a diffusion quotient. Boundary conditions are set to be Neumann and the initial value being a normalized sum of the δ-distributions centered around the input data points. In the following, we call a function $u \in C^{2,1}(\Omega \times \mathbb{R}_{>0}, \mathbb{R}_{\geq 0})$ the *diffusion kernel density estimator* (diffKDE), if it solves the diffusion partial differential equation

$$\frac{\partial}{\partial t} u(x;t) = \frac{1}{2} \frac{d^2}{dx^2} \left(\frac{u(x;t)}{p(x)} \right), \qquad x \in \Omega, t \in \mathbb{R}_{>0}, \tag{10}$$

$$\frac{\partial}{\partial x} \left(\frac{u(x;t)}{p(x)} \right) = 0, \qquad x \in \partial\Omega, t \in \mathbb{R}_{>0}, \tag{11}$$

$$u(x;0) = \frac{1}{N} \sum_{j=1}^{N} \delta(x - X_j), \qquad x \in \Omega. \tag{12}$$

In Eq. 12, the data are incorporated as initial values via the Dirac δ-distribution, i.e., a generalized function which takes the value infinity at its argument and zero anywhere else. Regarded as PDF, it puts all probability in the corresponding data point. The δ-distribution can be defined exactly as a limit of functions, the so-called Dirac sequence. In actual implementations, it has to be approximated, see Sec. 2.2.3.

The final iteration time $T \in \mathbb{R}_{>0}$ of the solution process of Eq. 10 is called the squared *bandwidth* of the diffKDE.

This specific type of KDE has several advantages. First of all, it naturally provides a sequence of estimates for different smoothing parameters (Chaudhuri and Marron, 2000). This obliterates identifying one single optimal bandwidth whose existence is questioned (e.g., Abramson, 1982; Terrell and Scott, 1992; Chaudhuri and Marron, 2000; Scott, 2012). Even more, such a sequence allows a specification of the estimate's smoothness that is most appropriate for the analysis. The parameter function p introduces adaptive smoothing properties (Botev et al., 2010). Thus, setting p properly solves the prior problem of having to locally adjust the bandwidth to the respective region to prevent oversmoothing of local data structure (Breiman et al., 1977; Terrell and Scott, 1992; Pedretti and Fernàndez-Garcia, 2013). In contrast to local bandwidth adjustments, local variations of the smoothing intensity can be applied to resolve multimodal data as well as values close to the boundary.

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

2.1.3 Bandwidth selection

According to the relationship $T = h^2$ between final iteration time T of the diffKDE and bandwidth parameter h (Chaudhuri and Marron, 2000), we from now on focus on the selection of the optimal squared bandwidth $T \in \mathbb{R}_{>0}$ and refer to this as the *bandwidth selection* for simplicity.

In Eq. 6 we stressed that the optimal choice of the bandwidth parameter depends on the true density f. In our setup of the diffKDE, the analytical solution for T of Eq. 5 depends not only on the true density f, but also on the parameter function p. It can be calculated as

$$T^* = \left(\frac{\mathbf{E}\left(\sqrt{p(X)}\right)}{2N\sqrt{\pi}\left\| \left(\frac{f}{p}\right)'' \right\|_{L^2}^2} \right)^{\frac{2}{5}} \tag{13}$$

(Botev et al., 2010). The role of the parameter function p is in detail described in Sec. 2.1.4.

In the simplified setup with $p = 1$ as in Eq. 9, the analytical optimal solution of Eq. 6 becomes

$$T^*_{(p=1)} = \left(\frac{1}{2\,N\,\sqrt{\pi}\,\|f''\|_{L^2}^2} \right)^{\frac{2}{5}}. \tag{14}$$

Still, the smoothing parameter depends on the unknown density function f and its derivatives. So we will need to find a suitable approximation of f, which might again be dependent of f and so on. Botev et al. (2010) use an iterative scheme to solve this implicit dependency. This additional effort is avoided in our approach.

The prior claimed possibility of no existence of one single optimal bandwidth for complicated densities (e.g. Scott, 2012) is, by default, no problem for the diffKDE. A solution to this problem is to create a family of estimates from different bandwidth parameters (Breiman et al., 1977), ranging from oversmoothed estimates to those with beginning oscillations (Sheather, 2004). For the diffKDE the progression of the time t up to a final iteration time T is equivalent to the creation of such a family of estimates. For the diffKDE we thus only need to find a suitable optimal final iteration time T^*. Then, the temporal solution of Eq. 10 provides solutions for the diffKDE for the whole sequence of the temporal discretization time steps smaller than T^*, which we can then use as the requested family of estimates.

2.1.4 Pilot estimation

A crude first estimate of the true density f can serve as a pilot estimation step for several purposes (Abramson, 1982; Sheather, 2004). The most obvious in our case is to obtain an estimate of f for the calculations of the optimal bandwidth in Eq. 13. The second purpose is its usage for the definition of the parameter function p in Eq. 10. Setting this as an estimate of the true density itself introduces locally adaptive smoothing properties (Botev et al., 2010). Since p appears in the denominator in the diffusion equation, it operates conversely to a classical diffusion coefficient. Choosing p to be a function allows for a spatially dependent influence on the smoothing intensity: at points where the function p is small, the smoothing becomes more pronounced, whereas if p is larger, the smoothing is less intense. This resolves the expected structure in data dense areas,

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

but expands in sparsely sampled areas. Eventually, we calculate two pilot estimates – one for p and one for f – to support the calculation of the diffKDE. We set both pilot estimates to be the solution of Eq. 9 with an optimal smoothing parameter approximating Eq. 14. This approach combines Gaussian and diffKDE interchangeably to make best use of both of their benefits (Chung et al., 2018).

2.2 Discretization of diffusion kernel density estimator

Equation 10 is solved numerically, using a spatial and temporal discretization. The discretization is based on finite differences and sparse matrices in Python. A similar approach can be found in a diffusion-based kernel density estimator for linear networks implemented in R by McSwiggan et al. (2016).

2.2.1 Spatial discretization

We start with the description of the discretization of the spatial domain $\Omega \subseteq \mathbb{R}$. This will reduce the partial differential equation in Eq. 10 into a system of linear ordinary differential equations.

Let $n \in \mathbb{N}$ and $(x_i)_{i=0}^n \subseteq \bar{\Omega}$, an equidistant discretization of Ω with $x_{i-1} < x_i$ and $\mathbb{R}_{>0} \ni h := x_i - x_{i-1}$ for all $i \in \{1,\dots,n\}$. For the following calculations, we set $x_{-1} := x_0 - h \in \mathbb{R}$ and $x_{n+1} := x_n + h \in \mathbb{R}$. Let u be the solution of the diffKDE and p its parameter function, both as defined in Sec. 2.1.2. We assume that u and p are both defined on x_{-1} and x_{n+1} and we set $u_i = u(x_i)$ and $p_i = p(x_i)$ for all $i \in \{-1,\dots,n+1\}$.

Let $t \in \mathbb{R}_{>0}$. We approximate Eq. 11 at $x = x_0$ by applying a first order central difference quotient as

$$0 = \frac{\partial}{\partial x}\left(\frac{u(x_0;t)}{p(x_0)}\right) = \frac{1}{2h}\left(\frac{u_1(t)}{p_1} - \frac{u_{-1}(t)}{p_{-1}}\right).$$

This implies

$$\frac{u_{-1}(t)}{p_{-1}} = \frac{u_1(t)}{p_1}.$$

We approximate Eq. 10 at $x = x_0$ by applying a second order central difference quotient

$$u_0'(t) = \frac{1}{2}\frac{1}{h^2}\left(\frac{u_1(t)}{p_1} - 2\frac{u_0(t)}{p_0} + \frac{u_{-1}(t)}{p_{-1}}\right) = \frac{1}{2}\frac{1}{h^2}\left(2\frac{u_1(t)}{p_1} - 2\frac{u_0(t)}{p_0}\right). \tag{15}$$

Analogously, we approximate Eq. 11 and Eq. 10 at $x = x_n$ again by first and second order central difference quotients, respectively. This gives

$$u_n'(t) = \frac{1}{2}\frac{1}{h^2}\left(\frac{u_{n+1}(t)}{p_{n+1}} - 2\frac{u_n(t)}{p_n} + \frac{u_{n-1}(t)}{p_{n-1}}\right) = \frac{1}{2}\frac{1}{h^2}\left(2\frac{u_{n-1}(t)}{p_{n-1}} - 2\frac{u_n(t)}{p_n}\right). \tag{16}$$

Finally, we derive from Eq. 10 by applying a second order central difference quotient for all $i \in \{1,\dots,n-1\}$:

$$u_i'(t) = \frac{1}{2}\frac{1}{h^2}\left(\frac{u_{i+1}(t)}{p_{i+1}} - 2\frac{u_i(t)}{p_i} + \frac{u_{i-1}(t)}{p_{i-1}}\right). \tag{17}$$

Now, we identify $\boldsymbol{p} := (p_0,\dots,p_n) \in \mathbb{R}^{n+1}$, $\boldsymbol{u}'(t) := (u_0'(t),\dots,u_n'(t)) \in \mathbb{R}^{n+1}$ and $\boldsymbol{u}(t) := (u_0(t),\dots,u_n(t)) \in \mathbb{R}^{n+1}$ with their spatial discretizations. Furthermore, we define $\boldsymbol{v}_{\text{upper}} := (2,1,\dots,1) \in \mathbb{R}^n$, $\boldsymbol{v}_{\text{main}} := (-2,\dots,-2) \in \mathbb{R}^{n+1}$ and $\boldsymbol{v}_{\text{lower}} :=$

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

$(1,...,1,2) \in \mathbb{R}^n$ to be the upper, main and lower diagonal of the tridiagonal matrix $\mathbf{V} \in \mathbb{R}^{(n+1)\times(n+1)}$. Now, we set

$$\frac{1}{2}\frac{1}{h^2}\mathbf{V}\frac{1}{\boldsymbol{p}} =: \mathbf{A} \in \mathbb{R}^{(n+1)\times(n+1)}, \tag{18}$$

where the division by $\boldsymbol{p}$ is meant to be column-wise. Then, Eq. 15, Eq. 16 and Eq. 17 can be summarized as a linear system of ordinary differential equations:

$$\boldsymbol{u}'(t) = \frac{1}{2}\frac{1}{h^2}\mathbf{V}\frac{\boldsymbol{u}(t)}{\boldsymbol{p}} = \mathbf{A}u(t). \tag{19}$$

By these calculations the partial differential equation from Eq. 10 becomes a system of ordinary differential equations:

$$\boldsymbol{u}'(t) = \mathbf{A}\boldsymbol{u}(t), \quad t \in \mathbb{R}_{>0}. \tag{20}$$

2.2.2 Temporal discretization

The time-stepping applied to solve the ordinary differential equation from Eq. 20 and Eq. 12 is again built on equidistant steps forward in time. Let $\Delta \in \mathbb{R}_{>0}$ small and set $t_0 := 0$ and $t_k := t_{k-1} + \Delta$ for all $k \in \mathbb{N}$. Set $u_{k,i} := u(x_i, t_k) \in \mathbb{R}$ for all $i \in \{0,\dots,n+1\}$ and $k \in \mathbb{N}_0$ and identify $\boldsymbol{u_k} := (u_{k,i})_{i=0}^n \in \mathbb{R}^{n+1}$ for all $k \in \mathbb{N}_0$ with their discretizations.

We use an implicit Euler method to approximate Eq. 20 for all $k \in \mathbb{N}_0$

$$\boldsymbol{u_{k+1}} = \Delta\mathbf{A}\boldsymbol{u_{k+1}} + \boldsymbol{u_k} \tag{21}$$

from which we obtain

$$\boldsymbol{u_k} = (\mathbf{I}_{n+1} - \Delta\mathbf{A})\,\boldsymbol{u_{k+1}} \text{ for all } k \in \mathbb{N}_0. \tag{22}$$

Together with the initial value Eq. 12 this describes an implementation-ready time stepping procedure. The linear equation for $\boldsymbol{u_{k+1}}$ will be solved in every time step $k \in \mathbb{N}_0$.

2.2.3 Initial value

The initial value in Eq. 12 depends on the δ-distribution (Dirac, 1927). The δ-distribution is not a proper function, but can be calculated as a limit of a suitable function sequence. A common approximation for the δ-distribution is to use a Dirac sequence (Hirsch and Lacombe, 1999). Such is a sequence $(\Phi_n)_{n\in\mathbb{N}}$ of integrable functions that are non-negative and satisfy

$$\int \Phi_n(x)\,dx = 1 \text{ for all } n \in \mathbb{N} \tag{23}$$

and

$$\lim_{n\to\infty} \int_{\mathbb{R}\setminus\mathcal{B}_\rho(0)} \Phi_n dx = 0 \text{ for all } \rho \in \mathbb{R}_{>0}. \tag{24}$$

For our implementation we define a Dirac sequence $(\Phi_h)_{h\in\mathbb{R}_{>0}}$ depending on the spatial discretization fineness $h \in \mathbb{R}_{>0}$ as an approximation of δ in Eq. 12. The relationship $\frac{|\Omega|}{n} = h$ provides the dependency of Φ_h on $n \in \mathbb{N}$ and the equivalence of the

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

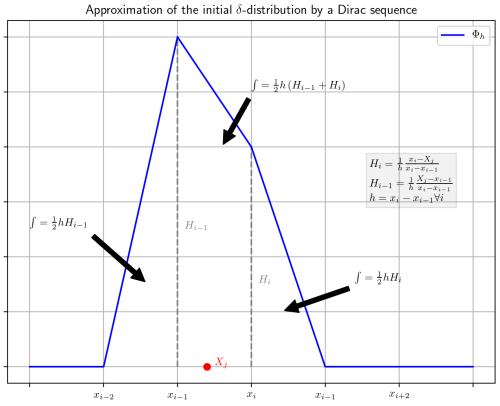

Figure 1. Dirac sequence $(\Phi_h)_{h\in\mathbb{R}_{>0}}$ for the approximation of the δ-distribution in the initial value in Eq. 12. The function Φ_h is depending on the spatial discretization fineness h and converges to δ for $h \to 0$. The function Φ_h is piecewise linear with a peak at each data point $X_j, j \in \{1, ..., N\}$ integrating to 1.

limits $n \to \infty$ and $h \to 0$ in this framework. In the following we give the specific definition of our function sequence of choice and proof that this indeed defines a proper Dirac sequence.

We assume $0 \in \Omega$. Then there exists an $i \in \{0, \dots, n\}$ with $0 \in [x_{i-1}, x_i)$. If not readily defined, we set $x_{i-2} := x_{i-1} - h \in \mathbb{R}$ and $x_{i+1} := x_i + h \in \mathbb{R}$. We define (see also Fig. 1)

$$\Phi_h : \Omega \to \mathbb{R}, x \mapsto \begin{cases} \frac{x_i}{h^3}x + \frac{x_i|x_{i-2}|}{h^3}, & x \in [x_{i-2}, x_{i-1}) \\ \frac{x_i+x_{i-1}}{h^3}x + x_i\frac{x_i+x_{i-1}}{h^3} - \frac{x_{i-1}}{h^2}, & x \in [x_{i-1}, x_i) \\ \frac{x_{i-1}}{h^3}x + \frac{x_{i+1}|x_{i-1}|}{h^3}, & x \in [x_i, x_{i+1}] \\ 0 & \text{else.} \end{cases} \tag{25}$$

Then $\Phi_h \in L^1(\mathbb{R})$ is non-negative for all $h \in \mathbb{R}_{>0}$ and $\int \Phi_h(x)\,dx = 1$ (see Appendix).

Now, let $\rho \in \mathbb{R}_{>0}$ and set $h = \frac{\rho}{2} \in \mathbb{R}_{>0}$. Then we have by Eq. 25

$$\int_{\mathbb{R}\setminus\mathcal{B}_\rho(0)} \Phi_h dx = \int_{-\infty}^{\rho} \Phi_h dx + \int_{\rho}^{\infty} \Phi_h dx = \int_{-\infty}^{\rho} 0 dx + \int_{\rho}^{\infty} 0 dx = 0,$$

and it follows

$$\lim_{h\to 0} \int_{\mathbb{R}\setminus\mathcal{B}_\rho(0)} \Phi_h dx = 0 \text{ for all } \rho \in \mathbb{R}_{>0}. \tag{26}$$

Hence Eq. 25 defines a Dirac sequence. We use Φ_h for the approximation of the δ-distribution in our implementation of Equation 12.

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023


Geoscientific Model Development Discussions

The concept of the Dirac sequence also provides the justification to generally rely on the δ-distribution in the construction of the initial value of the diffKDE. The Gaussian kernel defined in Eq. 8 that solves the diffusion equation as a fundamental solution is again a Dirac-sequence (Boccara, 1990). This link connects the diffKDE directly back to the δ-distribution.

2.3 Implementation of the diffusion kernel density estimator

The selected implementation is a straight forward approach using equidistant finite differences in space and time and a direct solution of the diffusion equation by an implicit Euler. It is build on the three times sequent solution of the diffusion equation, providing two pilot estimates for the calculation of the final diffKDE. The three chosen bandwidths increase in complexity and accuracy over this iteration. The implementation is realized in Python 3.

2.3.1 Selection of pilot function and optimal bandwidth

For the optimal bandwidth T^* from Eq. 13 we need the parameter function p as well as the true density f. We approximate them both by a simple KDE, each as pilot estimation steps. We use for both cases the simplified diffKDE defined in Eq. 9, without additional parameter functions. We denote the bandwidths for p and f as $T_p, T_f \in \mathbb{R}_{>0}$, respectively. We use a simple bandwidth as variants of the *rule of thumb* by Silverman (1986) for both of them.

We begin to estimate T_p, which is the bandwidth for the KDE that serves as p. It shall be the smoothest of the three estimates, since p limits the resolution fineness of the diffKDE as a lower boundary. This is, because the diffKDE converges to this parameter function and hence never resolves less details than p itself (Botev et al., 2010).

As seen in Eq. 14, the optimal bandwidth for the approximation of p is depending on the second derivative of f. We therefore need to make some initial assumption about f. For a first simplification, we assume that f belongs to the normal distribution family. Then the variance can be estimated by the standard deviation of the data. This leads us to the parametric approximation of the bandwidth T_P (Silverman, 1986)

$$T_p = \left(\frac{1}{2N\sqrt{\pi}\,\|f''\|_{L^2}^2}\right)^{\frac{2}{5}} = \left(\frac{1}{2N\sqrt{\pi}\sigma^{-5}\,\|\Phi''\|_{L^2}^2}\right)^{\frac{2}{5}} = \left(\frac{1}{2N\sqrt{\pi}\sigma^{-5}\frac{3}{8}\frac{1}{\sqrt{\pi}}}\right)^{\frac{2}{5}} = \sigma^2\left(\frac{4}{3}N\right)^{-\frac{2}{5}}, \tag{27}$$

whose estimate is known to be overly smooth on multimodal distributions.

To calculate the bandwidth T_f for the approximation of f in Eq. 13 we choose a refined approximation of Eq. 14, which has been proposed by Silverman (1986) as

$$T_f = \left(0.9min\left(\sigma, \frac{iqr\,(data)}{1.34}\right)\right)^2 N^{-\frac{2}{5}}. \tag{28}$$

We approximate optimal bandwidth T^* from Eq. 13 by calculating p and f by Eq. 9, based on Eq. 27, and Eq. 28 respectively. The nominator is approximated by the unbiased estimator (Botev et al., 2010)

$$\mathbf{E}\,(p(X)) = \frac{1}{n+1}\sum_{i=0}^{n+1}\sqrt{p(x_i)} =: E_\sigma \tag{29}$$

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023


Geoscientific Model Development Discussions

and the second derivative in the denominator by finite differences (McSwiggan et al., 2016)

$$\left(\frac{f}{p}\right)''(x_i) = \frac{1}{h^2}\left(\frac{f}{p}(x_{i+1}) - 2\frac{f}{p}(x_i) + \frac{f}{p}(x_{i-1})\right) =: q_i \tag{30}$$

for all $i \in \{1, ..., n\}$. For the boundary values we set

$$\left(\frac{f}{p}\right)''(x_0) = \frac{1}{h^2}\left(2\frac{f}{p}(x_1) - 2\frac{f}{p}(x_0)\right) =: q_0 \tag{31}$$

and

$$\left(\frac{f}{p}\right)''(x_{n+1}) = \frac{1}{h^2}\left(2\frac{f}{p}(x_{n-1}) - 2\frac{f}{p}(x_n)\right) =: q_{n+1}. \tag{32}$$

We set the finite differences approximation from Eq. 30, Eq. 31 and Eq. 32 as a discrete function with image $\boldsymbol{q} := (q_0, ..., q_{n+1})$. In this way we derived an already discrete formula for approximation of the optimal squared bandwidth $T^* \in \mathbb{R}_{>0}$ of the diffKDE on the discretization Ω as

$$T^* = \left(\frac{E_\sigma}{2N\sqrt{\pi}\,\|\boldsymbol{q}\|_{L^2}^2}\right)^{\frac{2}{5}}. \tag{33}$$

The L^2-norm is calculated on the discretized versions of f and p by array operations. The integration is performed by the *trapz* function of the SciPy integrate package (Gommers et al., 2022), the square root is part of the math package (Van Rossum, 2020).

2.3.2 The diffKDE algorithm with optimized bandwidth

The implementation is realized in Python and its concept shown in Alg. 1. We use the Python libraries Numpy (Harris et al., 2020) and SciPy (Virtanen et al., 2020; Gommers et al., 2022) and the Python Math module (Van Rossum, 2020) for data preprocessing, calculation of the bandwidths, setup of the differential equations and their solution. The algorithm iteratively calculates three KDEs: first the two for the approximations of p and f as the pilot estimation steps described in Sec. 2.3.1 and the last one being u the solution of the diffKDE built on the two prior. All three KDEs are calculated by solving the diffusion equation up to the respective final iteration time. The solution is realized in *while*-loops solving Eq. 22. The two pilot estimation steps can be calculated simultaneoulsy, since they are independent of each other and only differ in their final iteration times T_p and T_f. All input variables are displayed in Tab. 1 the return values listed in Tab. 2.

The spatial grid Ω is setup according to the description in Sec. 2.2.1 in lines 1 and 2 of Alg. 1. It consists of $n \in \mathbb{N}$ intervals, where n can be set by the user. The boundary values are $x_{min} := \min X \in \mathbb{R}$ and $x_{max} := \max X \in \mathbb{R}$ by default, but can also be chosen individually. Setting the boundary values to an individually chosen interval in the function call results in a clipping of the used data to this smaller one before KDE calculation. Outside the interval boundaries, the diffKDE adds two additional discretization points to make it applicable for the case of a data point X_i, $i \in \{0, ..., n+1\}$ being directly located at one of the boundaries. This way it is possible to construct the initial value defined in Eq. 25, which takes into account the two neighbouring discretization points in each direction. This leads to a full set of $n+1$ equidistant discretization points saved

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions
Open Access
EGU

Algorithm 1 Finite differences based algorithm for the implementation of the diffusion KDE.

Note: the routine solve$(\mathbf{M}, \mathbf{b})$ means that the system $\mathbf{Mx} = \mathbf{b}$ is solved.

Require: $\boldsymbol{X} \in \mathbb{R}^N$, $n \in \mathbb{N}$, $timesteps \in \mathbb{N}$, $x_{min} \in \mathbb{R}$, $x_{max} \in \mathbb{R}$

1: $h \leftarrow (x_{max} - x_{min}) \,/\, (n-4)$
2: $\boldsymbol{\Omega} \leftarrow (x_{min} - 2h, x_{min} - h, ..., x_{max} + h, x_{max} + 2h) \in \mathbb{R}^{n+1}$
3: $\boldsymbol{p}, \boldsymbol{f}, \boldsymbol{u} \leftarrow \Phi_h$
4: $T_p \leftarrow \sigma^2 \left(\frac{4}{3}N\right)^{-\frac{2}{5}}$
5: $T_f \leftarrow \left(0.9min\left(\sigma, \frac{iqr(data)}{1.34}\right)\right)^2 N^{-\frac{2}{5}}$
6: $t \leftarrow 0$, $\Delta_p \leftarrow T_p/timesteps$, $\Delta_f \leftarrow T_f/timesteps$
7: **while** $t < T_p$ **do**
8: $\quad \boldsymbol{p} \leftarrow \text{solve}\left(\mathbf{I}_{n+1} - \Delta_p \mathbf{A}_{pilot}, \boldsymbol{p}\right)$
9: $\quad \boldsymbol{f} \leftarrow \text{solve}\left(\mathbf{I}_{n+1} - \Delta_f \mathbf{A}_{pilot}, \boldsymbol{f}\right)$
10: $\quad t \leftarrow t + \Delta_p$
11: **end while**
12: $\boldsymbol{q} \leftarrow \sqrt{\int_\Omega \left(\left(\frac{\boldsymbol{f}}{\boldsymbol{p}}\right)''\right)^2 dh}$
13: $E_\sigma \leftarrow \frac{1}{n+1}\sum_{i=0}^{n+1}\sqrt{p(x_i)}$
14: $T \leftarrow \left(\frac{E_\sigma}{2N\sqrt{\pi}q^2}\right)^{\frac{2}{5}}$
15: $t \leftarrow 0$, $\Delta \leftarrow T/timesteps$
16: **while** $t < T$ **do**
17: $\quad \boldsymbol{u} \leftarrow \text{solve}\left(\mathbf{I}_{n+1} - \Delta\mathbf{A}, \boldsymbol{u}\right)$
18: $\quad t \leftarrow t + \Delta$
19: **end while**
20: **return** $\boldsymbol{u}, \boldsymbol{\Omega}, \boldsymbol{\Phi_h}, \boldsymbol{p}, \boldsymbol{stages}, \boldsymbol{times}$

in the variable Ω. The spatial discretization Ω includes an inner discretization between the handed in (or default set) interval endpoints x_{min} and x_{max} of $n-4$ equally sized inner discretization intervals.

The Dirac sequence Φ_h for the implementation of the initial value is defined in Eq. 25 and we use the same for initialization of all three approximations of the the PDF (p, f, u) in line 3 of Alg. 1. In its calculation, the algorithm searches for each $j \in \{1, ..., N\}$ for the $i \in \{1, ..., n+1\}$ with x_i being the closest right neighbour of X_j. Then the initial value is constructed by assigning the values $\frac{1}{h}\frac{x_i - X_j}{x_i - x_{i-1}}$ and $\frac{1}{h}\frac{X_j - x_{i-1}}{x_i - x_{i-1}}$ at grid point x_i and x_{i-1}, respectively, and zero elsewhere. These values are corresponding to the weighed heights H_i and H_{i-1} displayed in Fig. 1. The final initial value is the normalized sum of all these individual approximations of the δ-distribution. All three used KDEs (p, f, u) are initialized with this initial value.

In the pilot estimation steps, we calculate the KDEs for p and for f required for the set up of the bandwidth T^* for the diffKDE. The bandwidths T_p and T_f for p and f, respectively, are calculated based on the input data X in lines 4 and 5 of Alg. 1 as described in Sec. 2.3.1. Then, the KDEs are calculated by solving a linear ordinary differential equation by an implicit Euler in the first *while*-loop in lines 7 to 9 of Alg. 1. For the pilot estimation steps calculating p and f the matrix $\mathbf{A}$ defined in

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

Table 1. Input variables: The only required input variable for the calculation of the diffKDE is a one dimensional data set as an array like type. All other variables are optional, with some prescribed defaults. On demand the user can set individual lower and upper bounds for their data evaluation under the diffKDE as well as the number of used spatial and temporal discretization intervals. The individual selection of the final iteration time provides the opportunity to choose the specific smoothing grade on demand.

index	name	type	default	description
0	$data$	array like	required	input data $X \in \mathbb{R}$
1	$xmin$	float	$\min X$	lower data boundary for KDE calculation
2	$xmax$	float	$\max X$	upper boundary for KDE calculation
3	n	integer	1004	number of spatial discretization intervals in Ω
4	$timesteps$	integer	20	number of temporal discretization intervals
5	T	float	T^*	final iteration time for diffKDE

Eq. 18 does not incorporate a parameter function and reduces to

$$\frac{1}{2}\frac{1}{h^2}\mathbf{V} =: \mathbf{A}_{pilot} \in \mathbb{R}^{(n+1)\times(n+1)}. \tag{34}$$

Apart from this, the solutions for the pilot KDEs are the same as for the final diffKDE. The two pilot KDEs can be solved simultaneously, since they share their matrix $\mathbf{A}_{pilot}$ and have independent pre-computed bandwidths. The difference in their bandwidths is implemented in different time step sizes Δ_p and Δ_f for p and f, respectively, which are initialized in line 6 of Alg. 1 directly before this first *while*-loop. The time forward is calculated $timesteps \in \mathbb{N}$ times in equidistant time steps until each individual final iteration time derived by the respective bandwidths. Since we solve implicitly, there is no restriction to the time step size. But a larger $timesteps$ parameter reduces the numerical error proportional to the step size parameters Δ_p and Δ_f. In this temporal solution we rely on the fact that the involved matrices are sparsely covered. The applied solver is part of the SciPy Python library and designed for efficient solution of linear systems including sparse matrices (Virtanen et al., 2020; Gommers et al., 2022).

The final bandwidth T for the diffKDE solution u is calculated after the calculations of p and f, using them both as described in Sec. 2.3.1 in lines 12 to 14 of Alg. 1. For the diffKDE u the differential equation is given in Eq. 20 and the solution approach by the implicit Euler in Eq. 22. This is implemented in a second *while*-loop described in lines 16 to 18 in Alg. 1 and apart from the final iteration time T^* and the matrix $\mathbf{A}$ identical to the calculations in the pilot step.

The return value is a vector providing the user the diffKDE, along with some main parameters and the opportunity to also evaluate different approximation stages. It provides in the first and second entry the diffKDE and the spatial discretization Ω. The third entry is the initial value Φ_δ and the fourth pilot estimate p that influences the adaptive smoothing. The last return values two vectors are handed back: $stages$ and $times$. These include the approximation stages of the diffKDE and the respective times exceeding the default optimal solution stored in the diffKDE and providing also some oversmoothed solution

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

Table 2. Return values of the diffKDE: The return variable of the diffKDE is a vector. Its first entry is the diffKDE evaluated on the spatial grid. Its second entry is the spatial grid Ω.

index	name	type	size	description
0	u	Numpy array	$n+1$	diffKDE values on Ω
1	Ω	Numpy array	$n+1$	spatial discretization

for individual evaluations. The times are the 20 timesteps used for the calculation of u and 10 additional with doubled stepsize reaching up to the doubled approximated optimal final iteration time $2T^*$.

Possible problems are caught in *assert* and *if* clauses. First of all, the data is reshaped to a Numpy array for the case of a list handed in and it is made sure that this is non-empty. For the case of numerical issues leading to a pilot estimate including zero values, the whole pilot is set back equal to 1 to ensure numerical convergence. Similar is done for the case of NaN value being delivered for the optimal bandwidth for the diffKDE, in which case this is also set to the bandwidth chosen for f in Eq. 28.

2.4 Pre-implemented visual outputs

Besides the standard use to calculate a diffKDE at an approximated optimal final iteration time for direct usage, we also included three possibilities to generate a direct visual output, one of them being interactive. Matplotlib (Hunter, 2007) provides the software measures for creating the plots. Most methods are part of the submodule Pyplot, the interactive plot is based on the submodule Slider.

The function call *evol_plot* opens a plot showing the time evolution of the diffKDE. The plot includes drawings of the data points on the x-axis. In the background the initial values are drawn, but cut off at 20 % above the global maximum of the diffKDE to preserve focus of the graphic on the diffKDE and evolution. The evolution is presented by drawings of the individual time evolution stages using the sequential color map Viridis. In the front the diffKDE is drawn. This visualization of the evolution provides the user insight into the data distribution and their respective influence on the final form of the diffKDE.

The function call *pilot_plot* opens that shows the diffKDE together with its pilot estimate p, showing the intensity of local smoothing. With this the user has the possibility to gain insight to the influence of this pilot estimator on the performance of the diffKDE. This plot also includes the data points on the x-axis.

The function call *custom_plot* opens an interactive plot, allowing the user to slide through different approximation stages of the diffKDE. This feature is based on the Slider module from the Matplotlib library (Hunter, 2007) and opens a plot showing the diffKDE. On the bottom of this plot is a scale that shows the time, initially being set to the optimal iteration time derived from Eq. 13 in the middle of the scale. By clicking to the scale, the user can display the evolution stages at the respective (closest) iteration time. This reaches down to the initial value and up to the doubled optimal iteration time. This interactive tool provides the user a simple tool to follow the estimate at different bandwidths, the intensity of smoothing at different localizations. With

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023


Geoscientific Model Development Discussions

the help of such plot it is possible to decide on whether the diffKDE is desired to be applied with a final iteration time that is different from the default.

3 Results and Discussion

In the following we document the performance of the diffKDE on artificial and real marine biogeochemical data. Different data sources are chosen to best show possibilities and performance of the diffKDE. Additionally, snapshots of the pre-implemented plot routines are given as examples. Whenever not stated otherwise, we used the default values of the input variables stated in Tab. 2 in the calculation of the diffKDE.

For testing our implementation against a known true PDF we first constructed a three-modal distribution. The objective is to assess the diffKDE's resolution and to exemplify the pre-implemented plot routines. The distribution was constructed from three Gaussian kernels centered around $\mu_1 = 3$, $\mu_2 = 6.5$ and $\mu_3 = 9$ and with variances $\sigma_1^2 = 1$, $\sigma_2^2 = 0.7^2$ and $\sigma_3^2 = 0.5^2$, each of them with a relative contribution of 30 %, 60 % and 10 %, respectively:

$$f(x) = 0.3\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(x-3)^2} + 0.6\frac{1}{0.7\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-6.5}{0.7}\right)^2} + 0.1\frac{1}{0.5\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-9}{0.5}\right)^2}. \tag{35}$$

The performance of the diffKDE is then illustrated with real data of a) measurements of carbon isotopes (Verwega et al., 2021; Verwega et al., 2021) and b) of plankton size (equivalent spherical diameter) (Lampe et al., 2021). We chose these data because we propose to apply the diffKDE for the analysis of field data for assessment and optimization of marine biogeochemical- as well as size-based ecosystem models. The carbon isotope data have been collected to constrain model parameter values of a marine biogeochemical model that incorporates this tracer as a prognostic variable (Schmittner and Somes, 2016).

3.1 Pre-implemented outputs

As described in Sec. 2.4, we included three plot functions in the diffKDE implementation. All of them open pre-implemented plots, to give an impression of the special features that come with the diffKDE. An overview of the three possible direct visual outputs of the diffKDE software is described below.

First we outline the possibility to display the diffKDE's evolution. By calling the *evol_plot* function, a plot opens that shows all temporal evolution stages of the solution of Eq. 22. The temporal progress is visualized by a sequential colorscheme progressing from light yellow over different shades of green to dark blue. On the x-axis, all used data points are drawn and in the background a cut-off part of the initial value in light yellow as the beginning of the temporal evolution. The final diffKDE is plotted as a bold blue line in front of the evolution process. This gives the user an insight in the distribution of the initial data and their influence on the shape of the estimate. As an example of the default setting, we created an evolution plot from 100 random samples of Eq. 35 visualized in Fig. 2. The second example shows the possibility of displaying the diffKDE together with the pilot estimate p by the function *pilot_plot*. This is the parameter function in Eq. 12 responsible for the adaptive smoothing. Where this function is larger, the smoothing is less intense and allows more structure in the estimate of the diffKDE. Contrarily where it is smaller, the smoothing becomes more pronounced and data gaps are better smoothed out. The result of the diffKDE

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions
Open Access
EGU

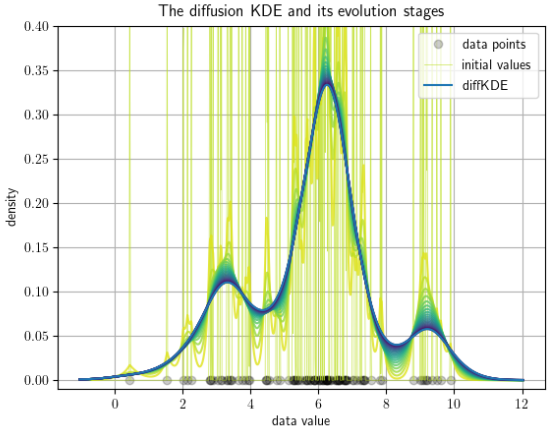

Figure 2. Pre-implemented direct visual output of the evolution process of the diffKDE. The input data are 100 samples randomly collected from Eq. 35. The individual data points are drawn on the x-axis. The y-axis represents the estimated probability density. The light yellow vertical lines in the background are the initial value of the the diffKDE. The temporal evolution of the solution of Eq. 22 is visualized by the sequent color scheme from light yellow over green to the bold blue graph in the front. The final diffKDE at the approximated optimal final iteration time represents as this graph the end of the time evolution.

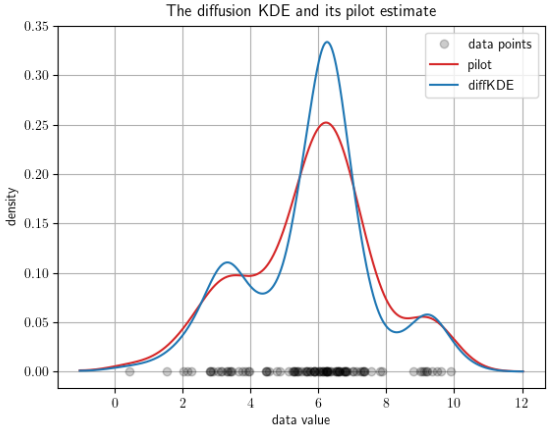

Figure 3. The diffKDE and its pilot estimate p. The input data are 100 samples randomly collected from Eq. 35. The data points are drawn on the x-axis. The y-axis represents the estimated density of the diffKDE in blue and the pilot estimate in red.

is shown together with its parameter function p in figure Fig. 3 on the same random sample of the distribution from Eq. 35 as before.

Lastly, we illustrate example snapshots of the interactive option to investigate different smoothing stages of the diffKDE by the function. We chose simpler and smaller example data for this demonstration, because these are better suited for visualization

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

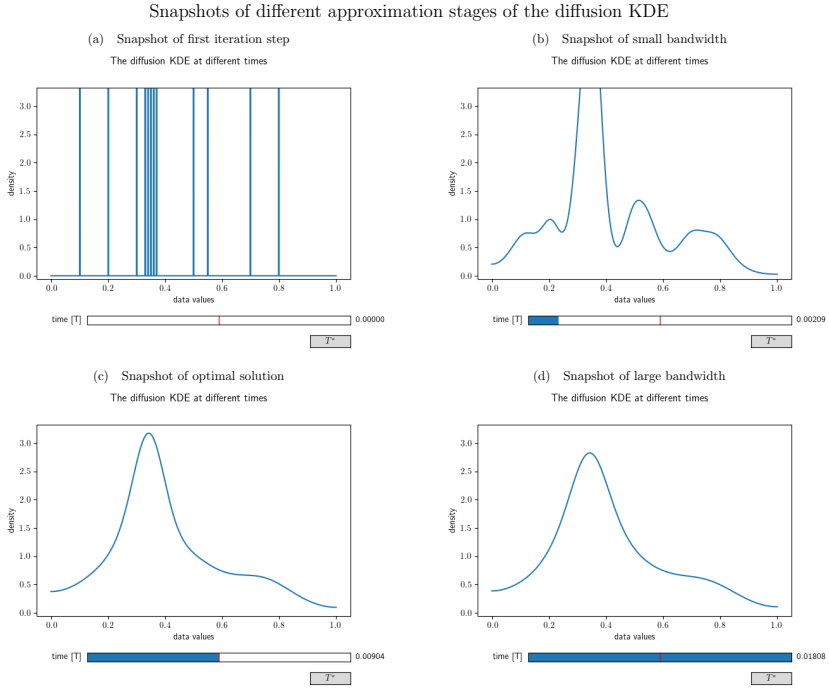

Figure 4. Different snapshots from the interactive visualization of the diffusion KDE generated from the artificial data set $(0.1, 0.2, 0.3, 0.33, 0.34, 0.35, 0.36, 0.37, 0.5, 0.55, 0.7, 0.8)$. (a) shows the output at $time = 0$ and hence the initial value. (b) shows an intermediate smoothing stage of the diffKDE. (c) shows the diffKDE of the input data at the approximated optimal iteration time T^*. This is the initial stage of the interactive graphic. By clicking the button on the lower right, the graphic can be reset to this stage. (d) shows an oversmoothed version of the diffKDE at the doubled approximated optimal iteration time.

of this tool's possibilities. The function *custom_plot* opens an interactive graphic, starting with a plot of the approximated optimal default solution of the diffKDE at T^*. In this graphic the user is able to individually choose, by a slider, the iteration time at which the desired approximation stage of the diffKDE can be seen. The time can be chosen from 0, where the initial value is shown, up until the doubled approximated optimal time ($2 \times T^*$). A reset button sets the graphic back to its initial stage of the diffKDE at T^*. Four snapshots of this interactive experience are drawn in Fig. 4.

3.2 Performance analyses on known distributions and in comparison to other KDEs

In this section we present results obtained by random samples of the trimodal distribution from Eq. 35 and lognormal distributions with differing parameters. Wherever suitable, the results are compared to other commonly used KDEs. These include the most common Gaussian KDE with the kernel function from Eq. 8 (Gommers et al., 2022), the Epanechnikov KDE with the kernel function from Eq. 7 (Pedregosa et al., 2012) and an improved implementation of the Gaussian KDE by Botev et al. (2010) in a Python implementation by Hennig (2021). We begin with an example of how the user may choose individually

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions
Open Access
EGU

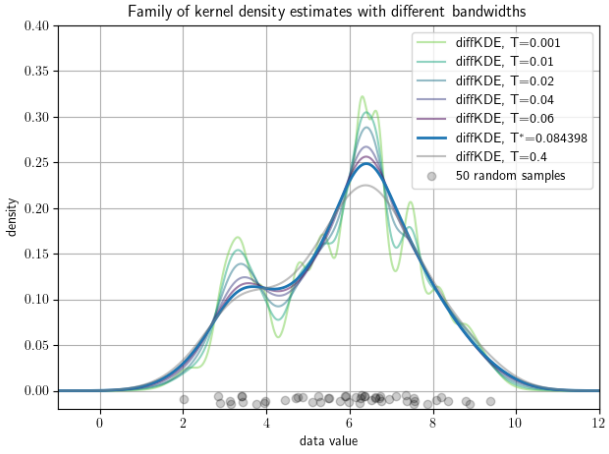

Figure 5. Family of diffKDEs evaluated at different bandwidths: A data set of 50 random samples drawn as grey circles on the x-axis serve to show the possibility to investigate a whole family of estimates by the diffDKE. The bold blue line represents the default solution of the diffKDE by solving the diffusion equation up to the approximated optimal final iteration time T^*. The other colors depict more detailed prior approximation stages with smaller bandwidth, i.e. earlier iteration times, and a smoother estimate with a far larger iteration time.

different smoothing grades of the diffKDE, then compare the different KDEs with the true distribution, followed by investigating the influence of noise on different KDEs, and finally show the convergence of different KDEs to the true distribution with increasing sample size.

We start with an individual selection of the approximation stages. This is one of the main benefits of the diffKDE compared to standard KDEs by providing naturally a family of approximations. This family can be observed by the function *custom_plot*. Individual members can be produced by setting the bandwidth parameter T in the function call of the diffKDE. This gives the user the chance to choose among more and less smooth approximations. A selection of such approximations along with the default solution are shown in Fig. 5 on a random sample of 50 data points from the trimodal distribution in Eq. 35. The plot shows how smaller iteration times resolve more structure in the estimate, while a substantially larger iteration time has only little influence on the increased smoothing of the diffKDE.

From now on we only work with the default solution of the diffKDE at T^*. We start with comparisons of the diffKDE and the three other popular KDEs directly to the underlying true distribution. The three other KDEs are the Gaussian KDE in an implementation from SciPy (Gommers et al., 2022), the Epanechnikov KDE in an implementation from Scikit-learn (Pedregosa et al., 2012) and an improved Gaussian KDE by Botev et al. (2010) in a Python implementation by Hennig (2021).

We use differently sized random samples of the known distribution from Eq. 35 and the standard lognormal distribution both over $[-1, 12]$, for a direct comparison of the accuracy of the KDEs. The random samples are 50 and 100 data points of each distribution and all four KDEs are calculated and plotted together in Fig. 6. The underlying true distribution is plotted in the background to visually assess the approximation accuracy. In general, the diffKDE resolves more of the details of the structure of the true distribution, while not being too sensitive to patterns introduced by the selection of the random sample and

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

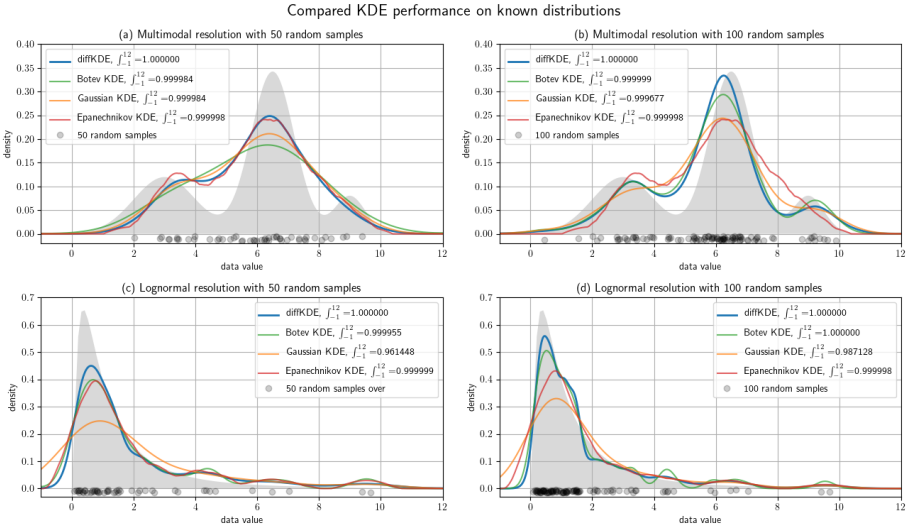

Figure 6. Test cases with known distributions: The plots (a) and (b) show KDEs of random samples of the trimodal distribution defined in Eq. 35, (c) and (d) the same for a lognormal distribution. The left figure column is constructed from 50 random samples, the right from 100. In all plots the true distribution is drawn in grey in the background and the random data sample as grey dots on the x-axis. Each subfigure shows four KDEs: the diffKDE, the Botev KDE, the Gaussian KDE and the Epanechnikov KDE. In the labels of the KDEs are also the integrals over the interval $[-1, 12]$ given for each of the KDEs

individual outliers. For the 50 random samples test of the trimodal distribution, all KDEs do not detect the third mode and only the diffKDE and the Epanechnikov KDE detect the second. The magnitude of the main mode is also best resolved by these two. In the 100 random samples test of the trimodal distribution, the diffKDE and the Botev KDE are able to detect all three modes. The main mode is best resolved by the diffKDE, whereas the third mode best by the Botev KDE. In both test cases for the trimodal distribution, the Gaussian KDE is the smoothest and the Epanechnikov KDE provides the least smooth graph. For 50 as well as for 100 random samples drawn from the lognormal distribution the magnitude and the steep decline to 0 is best reproduced by the diffKDE. The Gaussian KDE always performs the worst. The Botev KDE is generally also close to the diffKDE, but resolves in the tail of the distribution too much influence of individual outliers. An analysis of the the integral of the KDEs over the observed domain reveals that the diffKDE is the only one that integrates to 1 in all test cases.

We refined the test cases from Fig. 6 by investigating a lognormal distribution with different parameters and a restriction to the interval $[0, 12]$ in Fig. 7. We varied mean and variances of the normal distribution and used two different means and three different variances resulting in six test cases. All of them are run with 300 random samples and again with all four KDEs. The larger the variance becomes, the more structure of individual data points is resolved by the Botev KDE. The Gaussian KDE fails for increasing variance too, resulting in intense oversmoothing. The Epanechnikov KDE performs well for smaller variances and larger means, but also oversmoothes in the other cases. The diffKDE is generally one of the closest to the true distribution, while not resolving too much of the structure introduced by the choice of the random sample, especially for

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

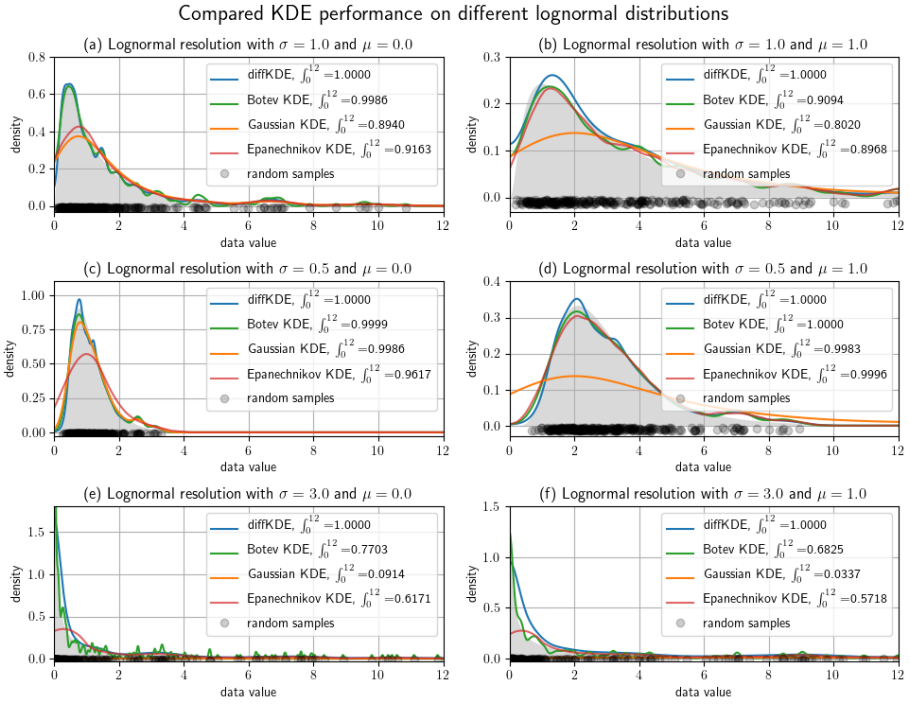

Figure 7. Lognormal test cases with different mean and variance parameters. Of each distribution 300 random samples were taken and the diffKDE, the Botev KDE, the Gaussian KDE and the Epanechnikov KDE calculated and plotted together with the true distribution. The random data sample is drawn as gray circles on the x-axis. (a) and (b) use $\sigma = 1$, (c) and (d) $\sigma = 0.5$ and (e) and (f) $\sigma = 3$ in their underlying normal distributions. The means are in the left column $\mu = 0$, the right column $\mu = 1$ of the underlying normal distribution.

increased variances. But this too tends to resolve too much structure in the vicinity of the mode for smaller variances. The integral of our implementation is again always exactly 1.

Now, we show the performance of the diffKDE on increasingly large data sets. We still use the trimodal distribution from Eq. 35. We start with four larger random data samples ranging from 100 to 10 million data points of the trimodal distribution and then being restricted to our core area of interest $[-1, 12]$. We calculate the diffKDE from all of them as well as the respective runtime on a consumer laptop from 2020. We compare the results again to the true distribution in Fig. 8. All of the estimates could be calculated in less than one minute. For 100 data points there is still an offset to the true distribution visible in the estimate. For the larger data samples the estimate only shows some minor uneven areas, which smooth out until the largest test case.

Furthermore, we investigated the convergence of the diffKDE to the true distribution, again in comparison to the three other KDEs. The error between the respective KDE and the true distribution is calculated by the Wasserstein distance (Panaretos and Zemel, 2019) with $p = 1$ by a SciPy function. We used increasingly large random samples from the trimodal distribution starting with 10 and reaching up to 1 million. The errors calculated for each of the KDEs on each of the random samples are

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

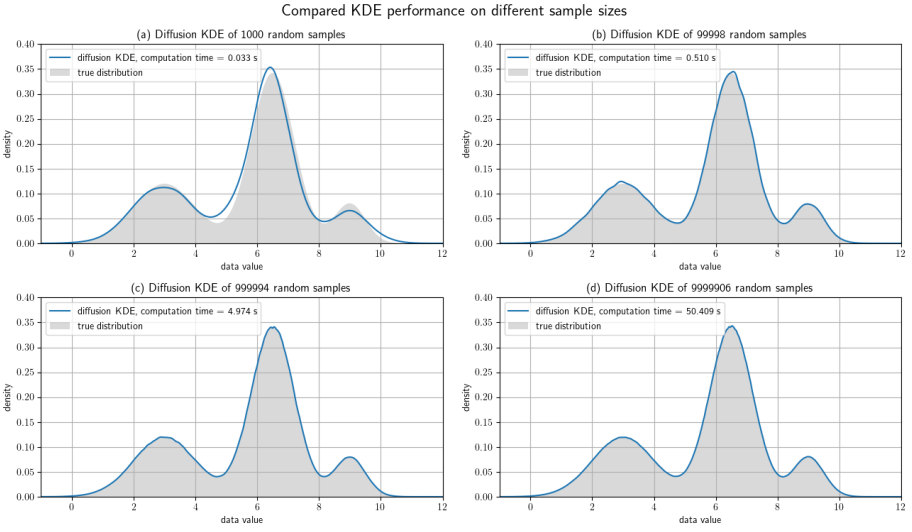

Figure 8. Test cases with different sample sizes: All four plots show the diffKDE of random samples of the known trimodal distribution defined in Eq. 35. (a) is calculated from a subsample of 100 data points, (b) $100,000$, (c) $1,000,000$ and (d) $10,000,000$, all cut to the interval $[-1,12]$ and hence lacking a few data points. The true numbers of incorporated data points in the four test cases are given in the respective sub-headings. The measured computing time on a 2020 MacBook Air is also drawn in the respective label.

listed in Tab. 3. The values from Tab. 3 are visualized in Fig. 9 on a log-scale and with a linear regression for each KDE's error values. The diffKDE, the Gaussian and the Botev show a similar steep decline, while the Epanechnikov KDE far slower decreases its error with increased sample size. The diffKDE and the Botev KDE generally show similar error values, the diffKDE relatively smaller ones on smaller data samples, the Botev KDE relatively smaller ones on data samples larger than around 5000.

Finally, we investigated the noise sensitivity of the diffKDE compared to the three other KDEs on data containing artificially introduced noise. We again used the trimodal distribution from Eq. 35 and 1000 random samples. From this, we created noised data $X_\theta \in \mathbb{R}^N$ by

$$(X_\theta)_i = (X)_i + (-1)^\tau \, rand 10^{-2} \theta \sigma \text{ for all } i \in \{1, ..., 1000\}, \tag{36}$$

where $\theta \in \{0, 1, 5, 15, 30\}$ defines the percentage of noise with respect to the standard deviation $\sigma \in \mathbb{R}$. $\tau \in \{1, 2\}$ was chosen randomly as well as $rand \in [0, 1]$. The error is again expressed by the Wasserstein distance between the original probability density and the respective KDE. The results are visualized in Fig. 10 with an individual panel for each KDE. The error of the Epanechnikov KDE is overall the largest and also increases to the largest. The Gaussian KDE produces the second largest error, but this even decreases with increased noise. The Botev KDE produces the smallest errors, but for increased noise this increases and approaches the magnitude of the one from the diffKDE. The error of the diffKDE only minimally responds to increased noise in the data. Visually, all four KDEs follow a similar pattern of a shift to the left of the graph. The Botev KDE additionally resolves more structure of the noised data as the noise increases.

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

Table 3. Error convergence

sample size	$error_{diffKDE}$	$error_{BKDE}$	$error_{GKDE}$	$error_{EKDE}$
10	0.02354	0.03618	0.02662	0.0273
50	0.01813	0.02484	0.02182	0.02017
100	0.00422	0.00724	0.01371	0.00702
150	0.00664	0.00937	0.01526	0.00933
200	0.00787	0.00894	0.01522	0.00967
300	0.0053	0.00621	0.01385	0.00849
400	0.00368	0.00484	0.01147	0.0081
500	0.0027	0.00324	0.01057	0.00761
750	0.00361	0.00343	0.00999	0.00807
1000	0.00321	0.00238	0.00933	0.00785
2000	0.00235	0.00171	0.00743	0.00771
5000	0.00154	0.00187	0.00578	0.00802
10000	0.00199	0.00188	0.00437	0.00791
50000	0.00113	0.00093	0.00234	0.00811
100000	0.00074	0.00059	0.00181	0.00822
500000	0.00048	0.00038	0.00108	0.00835
1000000	0.00046	0.00034	0.00084	0.00838

3.3 Performance analyses on biogeochemical data

In this final part, we show the diffKDE's performance on real marine biogeochemical field data. We chose two example data: A set of δ^{13}C in particulate organic carbon (POC) (Verwega et al., 2021) data and a set of plankton size spectra data (Lampe et al., 2021). Both data sets were already analyzed using KDEs in their original publications (Verwega et al., 2021; Lampe et al., 2021). Here we expand these analyses by a comparison of the KDEs used in the respective publications to the new implementation of the diffKDE. For the $\delta^{13}C_{POC}$ data, the Gaussian KDE was the one used in the data description publication. Since we have done this in the previous chapter, we furthermore added the Epanechnikov and the Botev KDE to these graphics. For the plankton size spectra data, we only compared the diffKDE to the two Gaussian KDEs used in the respective publication to preserve the clarity of the resulting figures.

The $\delta^{13}C_{POC}$ data (Verwega et al., 2021) was collected to serve for direct data analyses as well as for future model assessments (Verwega et al., 2021). We show here the Gaussian KDE as it was used in the data publication in a direct comparison to the diffKDE. Furthermore, we added the Epanechnikov and the Botev KDE. Since in this case no true known PDF is available, we have to compare the four estimates and subjectively judge their usefulness. In Fig. 11 we show the KDEs on four different

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions
Open Access
EGU

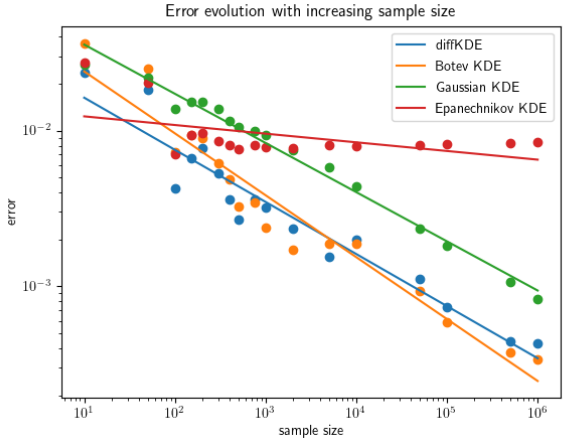

Figure 9. The evolution of the errors of the diffKDE, the Gaussian KDE, the Epanechnikov KDE and the Botev KDE are drawn on log-scale against the increasing sample size on the x-axis. The error has been calculated with the Wasserstein distance. A linear regression line on the log-scale is constructed from the discrete values of the individual errors for all four KDEs.

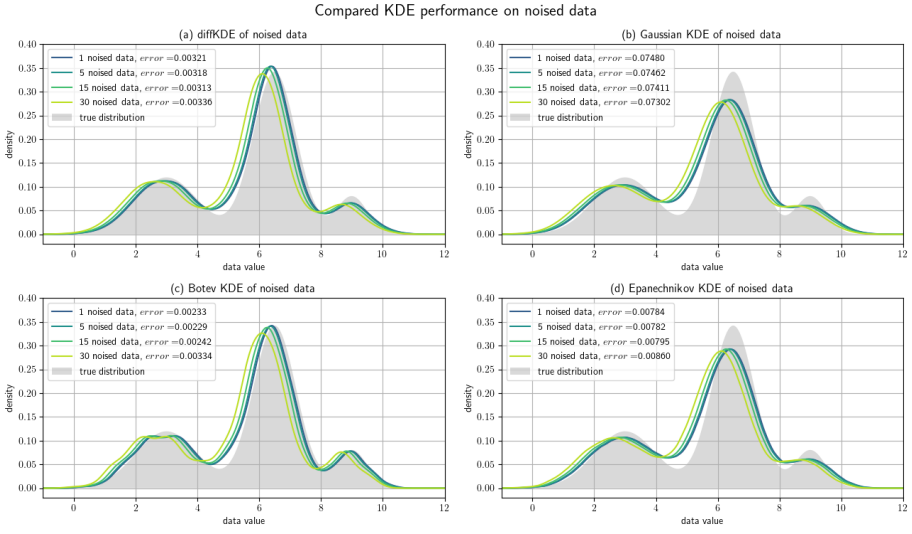

Figure 10. Noised data experiments: A random sample of 1000 data points of the trimodal distribution is artificially noised by differing amounts of the standard deviation. (a) shows the resulting diffKDEs of the differently noised data, (b) the Gaussian KDE, (c) the Botev KDE and (d) the Epanechnikov KDE. In all four panels the original true distribution is drawn in grey in the background. The values of the error between the KDEs and the original true distribution are also part of the respective labels.

subsets of the $\delta^{12}C_{POC}$ data: a) the full data set, b) a restriction to the core data interval of $[-35, -15]$, where 98.65 % of the data is located, and then even further restricted to c) the euphotic zone and d) only data sampled in the 1990s. In all three cases that

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions
Open Access
EGU

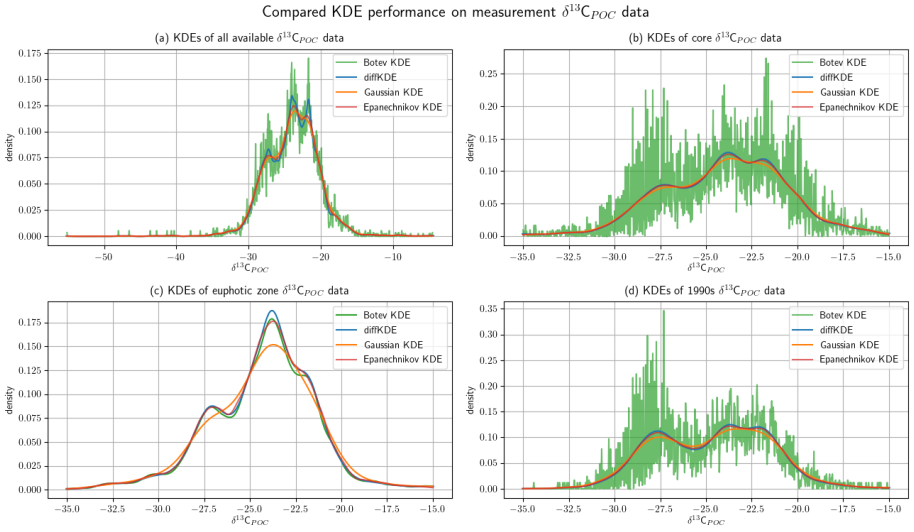

Figure 11. Comparison of KDE performance on marine biogeochemical field data: The $\delta^{13}C_{POC}$ data (Verwega et al., 2021) is in detail described in Verwega et al. (2021) and is covering all major world oceans, the 1960s to 2010s and reaches down into the deep ocean. In all four panels the diffKDE is plotted together with the Gaussian, the Epanechnikov and the Botev KDE. (a) Shows KDEs from all available data, (b) shows the KDEs of the data restricted to the core data values of $[-35, -15]$, (c) shows the KDEs from only euphotic zone data with values in $[-35, -15]$ and (d) the KDEs from all 1990s data with values in $[-35, -15]$.

involve deep ocean measurements, the Botev KDE produces strong oscillations while the Gaussian KDE strongly smoothes the dip between the modes at around $\delta^{13}C_{POC} = -24$ and $\delta^{13}C_{POC} = -22$ and mostly the one between around $\delta^{13}C_{POC} = -28$ and $\delta^{13}C_{POC} = -24$. The Epanechnikov KDE resolves more structure than the Gaussian, but still less pronounced than the diffKDE. Especially in the full data analysis, the diffKDE reveals the most structure while not resolving smaller data features of individual data points. The KDEs from the euphotic zone data are all reasonably smooth. The Gaussian KDE is again the smoothest and missing the mode at $\delta^{13}C_{POC} = -22$ completely. The other three KDEs resolve a similar amount of data structure. The Botev KDE reveals a better distinction between the modes at around $\delta^{13}C_{POC} = -24$ and $\delta^{12}C_{POC} = -22$ while the diffKDE shows the first one more pronounced. These observations are consistent with those from the experiments from Fig. 7 and Fig. Figure 10, where especially the Gaussian and the Botev KDE struggle with the resolution of data with increasing variances or noise. From the four here observed $\delta^{12}C_{POC}$ data sets the euphotic zone data shown in panel (c) in Fig. 11 has with 7.78 the smallest standard deviation. The other shown data has variances 13.91, 10.96 and 9.61 for panels (a), (b) and (d), respectively.

Another example demonstrates the performance of the diffKDE if applied to plankton size data (Lampe et al., 2021). The data of size, abundance of protist plankton was originally collected for resolving changes in plankton community size-structure, providing complementary insight for investigations of plankton dynamics and organic matter flux (e.g., Nöthig et al., 2015). In the study of Lampe et al. (2021) a KDE was applied for the derivation of continuous size spectra of phytoplankton and

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023


Geoscientific Model Development Discussions

microzooplankton that can potentially be used for the calibration and assessment of size-based plankton ecosystem models. In their study they used a Gaussian KDE, as proposed in Schartau et al. (2010), but with two different approaches for generating plankton size spectra. Uncertainties, also with respect to optimal bandwidth selection, were accounted for in both approaches by analyzing ensembles of pseudo-data resampled from original microscopic measurements. Smooth plankton spectra were obtained using the *combined* approach, where all phytoplankton and all zooplankton data were lumped together respectively and single bandwidths were calculated for every ensemble member (set of resampled data). This procedure avoided over-fitting but was also prone to over-smoothing. More structured size spectra were obtained with the *composite* approach, where individual size spectra were calculated for each species or genus and then pieced together. Since the variance within species or genus groups is smaller than within the large groups 'phytoplankton' or 'zooplankton', resulting bandwidths and therefore the degree of smoothing were considerably smaller than in the combined approach. This computationally expensive method revealed many details in the spectra, but at the same time tended to resolve narrow peaks that were either clearly insignificant or remained difficult to interpret (see supplemental material in Lampe et al. (2021)).The here proposed diffKDE is tested with resampled data used for the simpler *combined* approach. The objective is to identify details in the size spectra that remained previously unresolved while insignificant peaks, as found in the composite approach, become smoothed out. Figure 12 shows the performance of the diffKDE in comparison to the original combined and composite spectra that were derived as ensemble means of estimates obtained with a Gaussian KDE. The spatial discretization of the diffKDE was set to $n = 600$ to be comparable to the other already published KDEs in this case. The diffKDE seems to meaningfully combine the advantages of the two Gaussian KDE approaches in both spectra, of the phytoplankton and microzooplankton respectively. With the diffKDE it is possible to generate estimates that display more detailed structure of the composite KDE for cell sizes smaller than 10 μm, in particular in the microzooplankton spectrum. Concurrently, detailed variations, as caused by overfitting in the composite spectra, become suppressed for cell sizes larger than 10 μm. Thus, with the diffKDE it is possible to generate a single robust estimate that otherwise is only achieved by analyzing a series of estimates of a Gaussian KDE.

3.4 Future application to model calibration

The robustness of Earth system models is crucial for providing reliable climate projections for a sustainable development into Earth's future. Such models can assist the understanding of past and present and predict future conditions in the Earth system. Earth system models simulate the ocean's element cycling (e.g., Ilyina et al., 2013) and with this the ocean's carbon uptake capacity (e.g., Frölicher et al., 2015). They serve to assess the current and future state of our climate system and provide projections for different mitigation scenarios. This information can be used to support a sustainable development in our climate system (IPCC, 2022). As a consequence, political decisions depend on reliable projections to construct a safe pathway into Earth's future.

Calibration can increase the reliability of Earth system models (e.g., Oliver et al., 2022). For this purpose, a metric calculates the difference between simulated model output and measured field data. This metric defines the target or cost function in an optimization process, where unknown or uncertain model parameters are identified or estimated by numerical algorithms. This

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions
Open Access
EGU

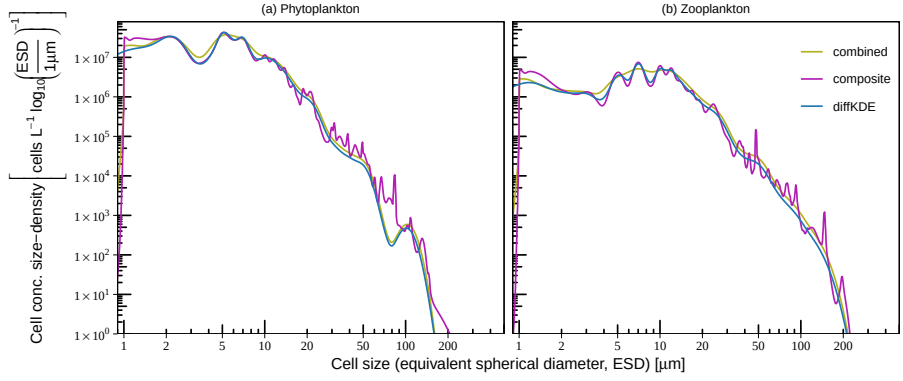

Figure 12. Comparison of KDE performance on (a) phytoplankton and (b) microzooplankton size spectra. The construction of composite and combined size spectra is described in Lampe et al. (2021) and based on Gaussian KDEs.Smoother combined spectra are the result of one KDE with a common bandwidth for all data. More structured composite spectra were assembled from taxon-specific spectra with individual, hence smaller, bandwidths.

process is sometimes also called "tuning" of the model. The result is usually a single or multiple sets of "optimal" parameters. They provide the model configuration with results closest to the incorporated field data.

Comparison of model and field data requires additional processing to account for spatial-temporal differences between collected samples and model resolution. Typically, simulation results are available at every single spatial grid point and in every time step. In comparison, field data are usually sparsely available only. Interpolating such sparse field data can introduce high uncertainty (e.g., Oliver et al., 2022). PDFs provide a useful approach to investigate data independent of the number of data points available (Thorarinsdottir et al., 2013). A comparison of two such functions can easily resolve the issue of non-equal field observations and simulation results. Histograms are commonly used as an approach to compare and ultimately constrain the distribution of model data to observations. However, many issues arrive including the subjective selection of intervals and histograms not being proper PDFs themselves.

The presented diffKDE provides a non-parametric approach to estimate PDFs with typical features of geoscientific data. Being able to resolve typical patterns such as multiple or boundary close modes, while being insensitive to noise and individual outliers makes the diffKDE a suitable tool for future work in the calibration and optimization of Earth system models.

4 Summary and conclusions

In this study we constructed and tested an estimator (KDE) of probability density functions (PDFs) that can be applied for analysing geoscientifc and ecological data. KDEs allow the investigation of data with respect to their probability distribution, and PDFs can be derived even for sparse data. To be well suited for geoscientific data, the KDE must work fast and reliably on differently sized data sets, while revealing multimodal details as well as features nearby data boundaries. A KDE should not

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

be overly sensitive to noise introduced by measurement errors or by numerical uncertainties. Such an estimator can be applied for direct data analyses or can be used to construct a target function for model assessment and calibration.

We presented a novel implementation of a KDE based on the diffusion heat process (diffKDE). This idea was originally proposed by Chaudhuri and Marron (2000) and its benefits in comparison to traditional KDE approaches were widely investigated by Botev et al. (2010). Our approach combines the solution of the diffusion equation with with two pilot estimation steps that correspond to the Gaussian KDE. We used an approximation of the optimal bandwidth for the diffKDE by a central differential quotient and plug-in of the pilot estimates. For their bandwidths we used variations of the *rule of thumb* by Silverman (1986). Our approach results in three subsequent estimations of the PDF, each of them chosen with a finer bandwidth approximation.

Finite differences build the fundamentals of our discretization. The spatial discretization are equidistant finite differences. The δ-distribution in the initial value is discretized by piecewise linear functions along the spatial discretization points constructing a Dirac-sequence. For the timestepping we applied an implicit Eulerian algorithm on an ordinary differential equation set up by a tridiagonal matrix corresponding to the diffusion equation on the spatial equidistant grid.

Our diffKDE implementation includes pre-implemented default output options. The first is the visualization of the diffusion time evolution showing the sequence of all solution steps from the initial values to the final diffKDE. This lets a user see the influence of individual data points and outlier accumulations on the final diffKDE and how this decreases over time. The second is the visualization of the pilot estimate that is also included in the partial differential equation to introduce adaptive smoothing properties. This provides the user an easy insight into the adaptive smoothing as well as the lower boundary of structure resolution given by this parameter function. Finally, an interactive plot provides a simple opportunity to explore all of these time iterations and look even beyond the optimal bandwidth and see smoother estimates.

Our implementation is fast and reliable on differently sized and multimodal data sets. We tested the implementation for up to 10 million data points and obtained acceptably fast results. A comparison of the diffKDE on known distributions together with classically employed KDEs showed reliable and often superior performance. For comparison we chose a SciPy implementation (Gommers et al., 2022) of the most classical Gaussian KDE (Sheather, 2004), an Scikit implementation (Pedregosa et al., 2012) of an Epanechnikov KDE (Scott, 1992) and a Python implementation (Hennig, 2021) of the improved Gaussian KDE developed by Botev et al. (2010). We designed multimodal and different boundary-close distributions and found our implementation to generate the most reliable estimates across a large range of sample sizes (Fig. 9). The diffKDE was neither prone to oversmoothing nor overfitting of the data, which we could observe in the other tested KDEs. A noise sensitivity test in comparison to the other KDEs also showed a good stability of the diffKDE against noise in the data.

An assessment of the diffKDE on real marine biogeochemical field data in comparison to usually employed KDEs reveals superior performance of the diffKDE. We used carbon isotope and plankton size spectra data and compared the diffKDE to the KDEs that were used to explore the data in the respective original data publications. On the carbon isotope data, we furthermore applied all previous KDEs for comparison. In both cases we were able to show that the diffKDE resolves relevant features of the data while not being sensitive to individual outliers or uncertainties (noise) in the data. We were able to obtain a best possible and reliable represantation of the true data distribution, better than those derived with other KDEs.

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

In future studies the diffKDE may potentially be used for the assessment, calibration and optimization of marine biogeochemical- and Earth system models. Already a plot of PDFs, of field data and simulation results respetcively, may provide visual insight into some shortcomings of the applied model. A target function can be constructed by adding a distance like the Wasserstein distance (Panaretos and Zemel, 2019) or other useful metrics for the calibration of climate models that can be investigated (Thorarinsdottir et al., 2013). Thus, KDE applications such as our diffKDE can greatly simplify comparisons of differently sized field and simulation data sets.

Code availability. The exact version of the diffKDE implementation (Pelz and Slawig, 2023) used to produce the results used in this paper is archived on Zenodo: https://doi.org/10.5281/zenodo.7594915.

Appendix A

Here, we briefly give the proof of the integral property of the used Dirac sequence Φ_h defined in Equation 25. Let $h \in \mathbb{R}_{>0}$. Then we obtain

$$\begin{aligned}
\int \Phi_h(x)\,dx &= \int_{x_{i-2}}^{x_{i-1}} \Phi_h(x)\,dx + \int_{x_{i-1}}^{x_i} \Phi_h(x)\,dx + \int_{x_i}^{x_{i+1}} \Phi_h(x)\,dx \\
&= \frac{1}{2}(x_{i-2}-x_{i-1})\frac{1}{x_{i-2}-x_{i-1}}\frac{x_i}{x_i-x_{i-1}} + \frac{1}{2}(x_i-x_{i-1})\left(\frac{1}{x_{i-2}-x_{i-1}}\frac{x_i}{x_i-x_{i-1}} + \frac{1}{x_{i+1}-x_i}\frac{-x_{i-1}}{x_i-x_{i-1}}\right) \\
&\quad + \frac{1}{2}(x_{i+1}-x_i)\frac{1}{x_{i+1}-x_i}\frac{-x_{i-1}}{x_i-x_{i-1}} \\
&= \frac{1}{2}h\frac{1}{h}\frac{x_i}{h} + \frac{1}{2}h\left(\frac{1}{h}\frac{x_i}{h} + \frac{1}{h}\frac{-x_{i-1}}{h}\right) + \frac{1}{2}h\frac{1}{h}\frac{-x_{i-1}}{h} \\
&= \frac{1}{2}\frac{x_i}{h} + \frac{1}{2}\frac{x_i}{h} - \frac{1}{2}\frac{x_{i-1}}{h} - \frac{1}{2}\frac{x_{i-1}}{h} \\
&= \frac{x_i - x_{i-1}}{h} \\
&= 1.
\end{aligned} \tag{A1}$$

Author contributions. MTP set up the manuscript and developed the implementation of the diffusion-based kernel density estimator. VL conducted the comparison experiments with the plankton size spectra. CJS edited the manuscript. MS and TS edited the manuscript and supported the development of the implementation of the diffusion-based kernel density estimator.

Competing interests. The contact author has declared that neither they nor their co-authors have any competing interests.

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions
Open Access
EGU

Acknowledgements. The first author is funded through the Helmholtz School for Marine Data Science (MarDATA), Grant No. HIDSS-0005.

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Geoscientific Model Development Discussions

References

Abramson, I. S.: On bandwidth variation in kernel estimates-a square root law, The annals of Statistics, pp. 1217–1223, 1982.

Boccara, N.: Functional Analysis - An Introduction for Physicists, Academic Press, Inc., 1990.

Botev, Z. I., Grotowski, J. F., and Kroese, D. P.: Kernel density estimation via diffusion, The annals of Statistics, 38, 2916–2957, https://doi.org/10.1214/10-AOS799, 2010.

Breiman, L., Meisel, W., and Purcell, E.: Variable kernel estimates of multivariate densities, Technometrics, 19, 135–144, 1977.

Chaudhuri, P. and Marron, J.: Scale space view of curve estimation, ANNALS OF STATISTICS, 28, 408–428, https://doi.org/10.1214/aos/1016218224, 2000.

Chung, Y.-W., Khaki, B., Chu, C., and Gadh, R.: Electric Vehicle User Behavior Prediction Using Hybrid Kernel Density Estimator, in: 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), pp. 1–6, https://doi.org/10.1109/PMAPS.2018.8440360, 2018.

Deniz, T., Cardanobile, S., and Rotter, S.: A PYTHON Package for Kernel Smoothing via Diffusion: Estimation of Spike Train Firing Rate, Front. Comput. Neurosci. Conference Abstract: BC11 : Computational Neuroscience & Neurotechnology Bernstein Conference & Neurex Annual Meeting 2011, 5, https://doi.org/10.3389/conf.fncom.2011.53.00071, 2011.

Dirac, P. A. M.: The physical interpretation of the quantum dynamics, Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 113, 621–641, https://doi.org/10.1098/rspa.1927.0012, 1927.

Farmer, J. and Jacobs, D. J.: MATLAB tool for probability density assessment and nonparametric estimation, SoftwareX, 18, 101 017, https://doi.org/10.1016/j.softx.2022.101017, 2022.

Frölicher, T. L., Sarmiento, J. L., Paynter, D. J., Dunne, J. P., Krasting, J. P., and Winton, M.: Dominance of the Southern Ocean in Anthropogenic Carbon and Heat Uptake in CMIP5 Models, Journal of Climate, 28, 862–886, https://doi.org/10.1175/jcli-d-14-00117.1, 2015.

Gommers, R., Virtanen, P., Burovski, E., Weckesser, W., Oliphant, T. E., Cournapeau, D., Haberland, M., Reddy, T., alexbrc, Peterson, P., Nelson, A., Wilson, J., endolith, Mayorov, N., Polat, I., van der Walt, S., Laxalde, D., Brett, M., Larson, E., Millman, J., Lars, peterbell10, Roy, P., van Mulbregt, P., Carey, C., eric jones, Sakai, A., Moore, E., Kai, and Kern, R.: scipy/scipy: SciPy 1.8.0, https://doi.org/10.5281/zenodo.5979747, 2022.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E.: Array programming with NumPy, Nature, 585, 357–362, https://doi.org/10.1038/s41586-020-2649-2, 2020.

Heidenreich, N.-B., Schindler, A., and Sperlich, S.: Bandwidth selection for kernel density estimation: a review of fully automatic selectors, AStA Advances in Statistical Analysis, 97, 403–433, https://doi.org/10.1007/s10182-013-0216-y, 2013.

Hennig, J.: John-Hennig/KDE-diffusion: KDE-diffusion 1.0.3, https://doi.org/10.5281/zenodo.4663430, 2021.

Hirsch, F. and Lacombe, G.: Elements of Functional Analysis, Springer, 1999.

Hunter, J. D.: Matplotlib: A 2D graphics environment, Computing in science & engineering, 9, 90–95, https://doi.org/10.1109/mcse.2007.55, 2007.

Ilyina, T., Six, K. D., Segschneider, J., Maier-Reimer, E., Li, H., and Núñez-Riboni, I.: Global ocean biogeochemistry model HAMOCC: Model architecture and performance as component of the MPI-Earth system model in different CMIP5 experimental realizations, Journal of Advances in Modeling Earth Systems, 5, 287–315, https://doi.org/10.1029/2012ms000178, 2013.

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

IPCC: Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [P.R. Shukla, J. Skea, R. Slade, A. Al Khourdajie, R. van Diemen, D. McCollum, M. Pathak, S. Some, P. Vyas, R. Fradera, M. Belkacemi, A. Hasija, G. Lisboa, S. Luz, J. Malley, (eds.)], https://doi.org/10.1017/9781009157926, 2022.

Jones, M. C., Marron, J. S., and Sheather, S. J.: A Brief Survey of Bandwidth Selection for Density Estimation, Journal of the American Statistical Association, 91, 401–407, https://doi.org/10.1080/01621459.1996.10476701, 1996.

Khorramdel, B., Chung, C. Y., Safari, N., and Price, G. C. D.: A Fuzzy Adaptive Probabilistic Wind Power Prediction Framework Using Diffusion Kernel Density Estimators, IEEE Transactions on Power Systems, 33, 7109–7121, https://doi.org/10.1109/tpwrs.2018.2848207, 2018.

Lampe, V., Nöthig, E.-M., and Schartau, M.: Spatio-Temporal Variations in Community Size Structure of Arctic Protist Plankton in the Fram Strait, Frontiers in Marine Science, 7, https://doi.org/10.3389/fmars.2020.579880, 2021.

Ma, S., Sun, S., Wang, B., and Wang, N.: Estimating load spectra probability distributions of train bogie frames by the diffusion-based kernel density method, International Journal of Fatigue, 132, 105 352, https://doi.org/10.1016/j.ijfatigue.2019.105352, 2019.

Marron, J. S. and Ruppert, D.: Transformations to reduce boundary bias in kernel density estimation, Journal of the Royal Statistical Society: Series B (Methodological), 56, 653–671, https://www.jstor.org/stable/2346189, 1994.

McSwiggan, G., Baddeley, A., and Nair, G.: Kernel Density Estimation on a Linear Network, Scandinavian Journal of Statistics, 44, 324–345, https://doi.org/10.1111/sjos.12255, 2016.

Nöthig, E.-M., Bracher, A., Engel, A., Metfies, K., Niehoff, B., Peeken, I., Bauerfeind, E., Cherkasheva, A., Gäbler-Schwarz, S., Hardge, K., Kilias, E., Kraft, A., Mebrahtom Kidane, Y., Lalande, C., Piontek, J., Thomisch, K., and Wurst, M.: Summertime plankton ecology in Fram Strait - a compilation of long- and short-term observations, Polar Research, 34, 23 349, https://doi.org/10.3402/polar.v34.23349, 2015.

Oliver, S., Cartis, C., Kriest, I., Tett, S. F. B., and Khatiwala, S.: A derivative-free optimisation method for global ocean biogeochemical models, Geoscientific Model Development, 15, 3537–3554, https://doi.org/10.5194/gmd-15-3537-2022, 2022.

Panaretos, V. M. and Zemel, Y.: Statistical Aspects of Wasserstein Distances, Annual Review of Statistics and Its Application, 6, 405–431, https://doi.org/10.1146/annurev-statistics-030718-104938, 2019.

Parzen, E.: On estimation of a probability density function and mode, The annals of mathematical statistics, 33, 1065–1076, 1962.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine Learning in Python, https://doi.org/10.48550/ARXIV.1201.0490, 2012.

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

Pelz, M.-T. and Slawig, T.: Diffusion-based kernel density estimator (diffKDE), https://doi.org/10.5281/ZENODO.7594915, 2023.

Qin, B. and Xiao, F.: A Non-Parametric Method to Determine Basic Probability Assignment Based on Kernel Density Estimation, IEEE Access, 6, 73 509–73 519, https://doi.org/10.1109/ACCESS.2018.2883513, 2018.

Schartau, M., Landry, M. R., and Armstrong, R. A.: Density estimation of plankton size spectra: a reanalysis of IronEx II data, Journal of Plankton Research, 32, 1167–1184, https://doi.org/10.1093/plankt/fbq072, iSBN: 0142-7873, 2010.

Schmittner, A. and Somes, C. J.: Complementary constraints from carbon (^{13}C) and nitrogen (^{15}N) isotopes on the glacial ocean's soft-tissue biological pump, Paleoceanography, pp. 669–693, https://doi.org/10.1002/2015PA002905, 2016.

https://doi.org/10.5194/gmd-2023-17
Preprint. Discussion started: 13 February 2023

Scott, D. W.: Multivariate density estimation: theory, practice, and visualization, John Wiley & Sons, 1992.

Scott, D. W.: Multivariate density estimation and visualization, in: Handbook of computational statistics, pp. 549–569, Springer, https://doi.org/10.1007/978-3-642-21551-3-19, 2012.

Sheather, S. J.: Density Estimation, Statistical Science, 19, 588–597, https://doi.org/10.1214/088342304000000297, 2004.

Sheather, S. J. and Jones, M. C.: A reliable data-based bandwidth selection method for kernel density estimation, Journal of the Royal Statistical Society: Series B (Methodological), 53, 683–690, 1991.

Silverman, B.: Density estimation, Monographs on Statistics and Applied Probability, 1986.

Terrell, G. R. and Scott, D. W.: Variable kernel density estimation, The Annals of Statistics, pp. 1236–1265, https://www.jstor.org/stable/2242011, 1992.

Thorarinsdottir, T. L., Gneiting, T., and Gissibl, N.: Using Proper Divergence Functions to Evaluate Climate Models, SIAM/ASA Journal on Uncertainty Quantification, 1, 522–534, https://doi.org/10.1137/130907550, 2013.

Van Rossum, G.: The Python Library Reference, release 3.8.2, Python Software Foundation, 2020.

Verwega, M.-T., Somes, C. J., Schartau, M., Tuerena, R. E., Lorrain, A., Oschlies, A., and Slawig, T.: Description of a global marine particulate organic carbon-13 isotope data set, Earth System Science Data, 13, 4861–4880, https://doi.org/10.5194/essd-13-4861-2021, 2021.

Verwega, M.-T., Somes, C. J., Tuerena, R. E., and Lorrain, A.: A global marine particulate organic carbon-13 isotope data product, https://doi.org/10.1594/PANGAEA.929931, 2021.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods, 17, 261–272, https://doi.org/10.1038/s41592-019-0686-2, 2020.