Citation
Baddeley, A. and Davies, T.M. and Rakshit, S. and Nair, G. and McSwiggan, G. 2022. Diffusion Smoothing for Spatial Point Patterns. Statistical Science. 37 (1): pp. 123-142.
http://doi.org/10.1214/21-STS825

APPENDICES
for the article
Diffusion Smoothing for Spatial Point Patterns

A. Baddeley T.M. Davies S. Rakshit G.M. Nair G. McSwiggan

Contents

A Heat kernel in a rectangle

On a a one-dimensional interval $[A, B]$, the heat kernel has the expansion

$$\kappa_t^{[A,B]}(x \mid y) = \sum_{k\in\mathbb{Z}} [\varphi_t(x - y + 2kL) + \varphi_t(-x - y + 2kL + 2A)]$$

where $L = B - A$ and φ_t is the probability density of the standard normal distribution. See, e.g. Boyce & DiPrima (1969, pp. 456–458).

For the two-dimensional rectangle $W = [A, B] \times [C, D]$, the heat kernel is the product of 1-dimensional heat kernels: for a source point $y = (u_y, v_y)$ and query point $x = (u_x, v_x)$,

$$\kappa_t(x \mid y) = \kappa_t^{[A,B]}(u_x \mid u_y)\kappa_t^{[C,D]}(v_x \mid v_y)$$

It can be verified directly (i.e. by differentiating the expression above) that this satisfies the heat equation (4) and the Neumann boundary condition (5) in the main paper.

B Behaviour of the heat kernel in special cases

B.1 Behaviour near the boundary

Suppose W is the infinite half-plane $H = \{(u, v) : \ u \geq 0\}$. For data location $x_0 = (u_0, v_0)$ and query location $x = (u, v)$, the heat kernel is

$$\kappa_t(x \mid x_0) = \boldsymbol{\phi}_t(x - x_0) + \boldsymbol{\phi}_t(x - x_0^*)$$

where $\boldsymbol{\phi}_t$ is the 2-dimensional isotropic Gaussian density, and $x_0^* = (-u_0, v_0)$ is the mirror image of x_0 reflected in the y axis.

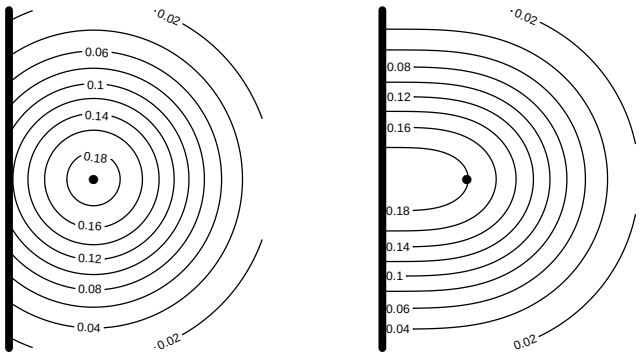

Figure A1: Behaviour of kernels near an infinite straight line boundary. *Left:* Gaussian kernel renormalised as in Jones-Diggle correction. *Right:* heat kernel. The domain W is the infinite half-plane to the right of the vertical axis (thick solid line). The source point ($\bullet$) is at location $(1, 0)$. Bandwidth is $\sigma = 1$.

The corresponding contribution to the Jones-Diggle corrected fixed-bandwidth estimate is $\boldsymbol{\phi}_t(x - x_0)/c_W(x_0)$, where $c_W(x_0) = \int_W \boldsymbol{\phi}_t(x - x_0)\,\mathrm{d}x = 1 - \Phi_t(-u_0)$ in which Φ_t is the c.d.f. of the 1-dimensional Gaussian. These expressions coincide when $u_0 = 0$, that is, when the data location lies exactly on the boundary of the window, when they both equal $2\boldsymbol{\phi}_t(x - x_0)$. They also coincide when the data location is far away from the window, when they both approach $\boldsymbol{\phi}_t(x - x_0)$. At intermediate locations there are discrepancies: an example is shown in Figure A1.

B.2 Performance in a square

The heat kernel can be evaluated rapidly when the domain W is a rectangle, as shown in Appendix A. This allows us to compare the performance of the edge-corrected Gaussian kernel estimators (2)–(3) and the diffusion estimator (7) of the main paper.

Lemmas 3, 5 and 6 of the main paper were stated for the diffusion estimator but generalise to the edge-corrected Gaussian estimators (2)–(3), because each estimator takes the form $\widehat{\lambda}(u) = \sum_i G(u, x_i)$ where $G(u, v)$ is a known function. Namely, $G(u, v) = e(u)\varphi_\sigma(u - v)$ for the uniform correction, and $G(u, v) = e(v)\varphi_\sigma(u - v)$ for the Jones-Diggle correction, where φ is the Gaussian kernel and $e(u) = 1/\int_W \varphi_\sigma(v - u)\,\mathrm{d}v$. See Baddeley et al. (2015, p. 173).

Consider a homogeneous Poisson process with intensity $\lambda > 0$. The pointwise bias $B_\lambda(x) = \mathbb{E}[\widehat{\lambda}(x)] - \lambda$ and pointwise variance $V_\lambda(x) = \mathsf{var}[\widehat{\lambda}(x)]$ are proportional to λ, so that $B_\lambda(x) = \lambda B_1(x)$ and $V_\lambda(x) = \lambda V_1(x)$. The functions $B_1(x)$ and $V_1(x)$ can be evaluated numerically for each of the estimators mentioned. The pointwise mean square error is $M_\lambda(x) = V_\lambda(x) + B_\lambda(x)^2 = \lambda V_1(x) + \lambda^2 B_1(x)$.

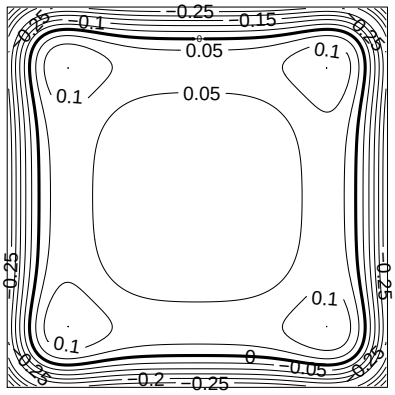

Figure A2: Relative bias $B_\lambda(x)/\lambda = B_1(x)$ of the Jones-Diggle-corrected Gaussian kernel estimator of point process intensity for a homogeneous Poisson process in the unit square. Bandwidth $\sigma = 0.1$. Thickened line is contour for zero bias.

Figure A2 shows the relative bias function B_1 for the Jones-Diggle corrected Gaussian kernel estimator, and Figure A3 shows the normalised variance functions V_1 for each estimator, calculated for $W = [0, 1]^2$ using bandwidth $\sigma = 0.1$.

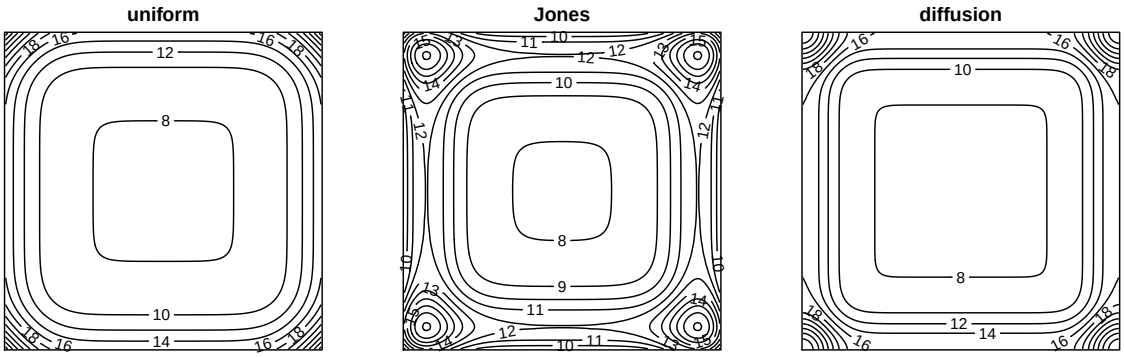

Figure A3: Normalised variance $V_\lambda(x)/\lambda = V_1(x)$ for the uniform-corrected Gaussian (*Left*), Jones-Diggle-corrected Gaussian (*Middle*) and diffusion kernel (*Right*) estimators. Unit square domain. Bandwidth $\sigma = 0.1$. Homogeneous Poisson point process.

The uniform corrected Gaussian kernel estimator and the diffusion estimator have qualitatively similar performance, with a steep increase in pointwise variance close to the corners of the domain. The Jones-Diggle correction has a different, more complicated pattern of performance. The *range* of the variance term $V_1(x)$ is $[7.96, 30.69]$ for the uniform correction, $[7.96, 18.13]$ for the Jones-Diggle correction and $[7.96, 31.78]$ for the diffusion estimator. The uniform corrected estimator and the diffusion estimator have zero bias, while the Jones-Diggle correction has bias term $B_1(x)$ with range $[-0.49, 0.15]$. For $\sigma = 0.1$,

Table A1: Terms involved in the Integrated Mean Squared Error (IMSE) for each estimator. Poisson process with uniform intensity in the unit square.

Estimator	Term	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$
Uniform	$I(V_1)$	10.599	3.406	1.908
Jones–Diggle	$I(V_1)$	10.599	3.406	1.908
	$I(B_1^2)$	0.011	0.024	0.036
Diffusion	$I(V_1)$	11.029	3.645	2.075

the *maximum* value of pointwise mean square error over the domain is smallest for the Jones-Diggle estimator when $\lambda < 80$, and for the uniform correction when $\lambda > 80$.

If instead we compare performance using the Integrated Mean Square Error

$$\text{IMSE} = \int_W M_\lambda(x)\,\mathrm{d}x = \int_W V_\lambda(x)\,\mathrm{d}x + \int_W B_\lambda(x)^2\,\mathrm{d}x = \lambda \int_W V_1(x)\,\mathrm{d}x + \lambda^2 \int_W B_1(x)^2\,\mathrm{d}x,$$

then the uniform correction is the most efficient for all values of λ. Table A1 shows the terms $I(V_1) = \int_W V_1(x)\,\mathrm{d}x$ and $I(B_1^2) = \int_W B_1^2(x)\,\mathrm{d}x$ for each estimator. It can be verified directly that the integrated variances for the uniform and Jones-Diggle corrections are equal in this case.

Overall, the diffusion estimator combines the advantages of the Gaussian kernel estimator with uniform correction (namely unbiasedness for the uniform) and Jones-Diggle correction (namely conservation of mass); its performance is closest to the former.

C Algorithms and software implementation

This section gives details of the algorithms described in Sections 4 and 6 of the main paper for computing the fixed-bandwidth, variable-rate and lagged-arrival diffusion estimates. The specifications should be sufficient for users wishing to code their own implementation.

We have implemented these algorithms in the R language using the package `spatstat` (Baddeley & Turner 2005, Baddeley et al. 2015) for spatial data. The transition matrix $\boldsymbol{P}$ is represented efficiently as a sparse matrix (Pissanetzky 1984), using the `Matrix` package (Bates & Maechler 2017).

C.1 Fixed-bandwidth algorithm

Algorithm 1 below specifies the procedure for discretising continuous data, and Algorithm 2 specifies the calculation of the diffusion estimator.

These algorithms are given for rectangular grids. For a regular hexagonal grid, the essential modifications are that, for hexagons of side length Δc, the maximum permitted value of Δt is $((1-\epsilon)/2)(\Delta c)^2$, and the corresponding value of the transition probability is $q = (\Delta t)/(3(\Delta c)^2)$.

Algorithm 1: Discretisation procedure onto a 4-connected or 8-connected rectangular grid.

Given the spatial domain $W \subset \mathbb{R}^2$, the point pattern dataset $\boldsymbol{x} = \{x_1, \ldots, x_n\}$, pixel dimensions $\Delta x, \Delta y$ and grid connectivity $v \in \{4, 8\}$:

1. Let $G = \{(i\Delta x, j\Delta y) : i, j \in \mathbb{Z}\}$ be the infinite rectangular grid with the desired spacing. Find the set $\boldsymbol{C} = W \cap G$ of grid nodes inside W. Let $N = \#(\boldsymbol{C})$ be the total number of nodes and $A = (\Delta x)(\Delta y)$ the area of one grid cell.
2. For each data point x_i, define its discretised counterpart x_i^* to be the element of C closest to x_i. Construct the $N \times 1$ column vector $\boldsymbol{s} = \left[s_c\right]_{c \in \boldsymbol{C}}$ where $s_c = \sum_i \mathbb{1}\{x_i^* = c\}$ is the number of discretised data points located at node c.
3. Declare two points $a = (i\Delta x, j\Delta y)$ and $b = (i'\Delta x, j'\Delta y)$ to be *horizontal neighbours* if $j = j'$ and $|i - i'| = 1$; *vertical neighbours* if $i = i$ and $|j - j'| = 1$; and *diagonal neighbours* if $|i - i'| = |j - j'| = 1$. Construct the sparse matrix $\boldsymbol{M}_{\text{horz}} = \left[m_{a,b}^{(\text{horz})}\right]_{a,b \in C}$ with binary entries $m_{a,b}^{(\text{horz})}$ equal to 1 whenever a and b are horizontal neighbours. Similarly construct $\boldsymbol{M}_{\text{vert}}$ for vertical neighbors, and if $v = 8$, construct $\boldsymbol{M}_{\text{diag}}$ for diagonal neighbors.

Algorithm 2: Constant-rate diffusion smoothing on a 4-connected or 8-connected rectangular grid.

Given the inputs $W, \boldsymbol{x}, \Delta x, \Delta y, v$ required for Algorithm 1, together with a smoothing bandwidth $\sigma > 0$ and minimum stayput probability $\epsilon > 0$:

1. Discretise the data as described in Algorithm 1.
2. Determine the largest permitted value of Δt as $\Delta t_{\max} = (1-\epsilon)\frac{(\Delta x)^2(\Delta y)^2}{(\Delta x)^2+(\Delta y)^2}$ if $v = 4$, and $\Delta t_{\max} = (1-\sqrt{\epsilon})\min\{\Delta x, \Delta y\}^2$ if $v = 8$.
3. Calculate the number of iterations $\tau = \left\lceil \frac{h^2}{\Delta t_{\max}} \right\rceil$ where $\lceil x \rceil$ is the smallest integer $m \geq x$.
4. Calculate the single-step variance $\Delta t = \sigma^2/\tau$.
5. Infer the jump probabilities $q_x = (\Delta t)/(2(\Delta x)^2)$ and $q_y = (\Delta t)/(2(\Delta y)^2)$.
6. Construct the transition matrix $\boldsymbol{P}$ as follows. First set

$$\boldsymbol{U} = \begin{cases} q_x \boldsymbol{M}_{\text{horz}} + q_y \boldsymbol{M}_{\text{vert}} & \text{if } v = 4, \\ q_x(1-2q_y)\boldsymbol{M}_{\text{horz}} + q_y(1-2q_x)\boldsymbol{M}_{\text{vert}} + q_x q_y \boldsymbol{M}_{\text{diag}} & \text{if } v = 8. \end{cases}$$

   Then set $\boldsymbol{P} = \boldsymbol{U} + I - \text{diag}(U\mathbf{1}_N)$ where $\text{diag}(z)$ is the diagonal matrix with diagonal entries $z_1, \ldots, z_N$. That is, $p_{a,b} = u_{a,b}$ for $a \neq b$, and $p_{a,a} = 1 - \sum_{k \in \boldsymbol{C}} u_{ak}$.
7. Define the initial state as the $N \times 1$ vector $\boldsymbol{f}_0 = \boldsymbol{s}/A$. Recursively compute $\boldsymbol{f}_{k+1} = \boldsymbol{f}_k \boldsymbol{P}$ for $k = 1, \ldots, \tau - 1$. Return $\boldsymbol{f}_\tau$ as the intensity estimate.

A slight modification of Algorithm 2 makes it possible to compute the general solution of the heat conduction problem in equation (11) of the main paper, simply by initialising $\boldsymbol{f}_0 = \boldsymbol{\lambda}$.

C.2 Adaptive estimator, variable rate

Algorithm 3 below specifies our algorithm for the variable-rate diffusion estimator. This is a modification of Algorithm 2 in which transition probabilities are adjusted to produce the required spatial variation in diffusion speed while still satisfying the constraints of a probability distribution. Numerical underflow can occur if the smoothing bandwidth function $\sigma(x)$ has a wide range of values.

Algorithm 3: Variable-rate diffusion smoothing on a 4-connected or 8-connected rectangular grid.

Given the inputs $W, \boldsymbol{x}, \Delta x, \Delta y, v$ required for Algorithm 1, together with a smoothing bandwidth function $\sigma(x),\ x \in W$ and minimum stayput probability $\epsilon > 0$:

1. Discretise the data as described in Algorithm 1.
2. Evaluate the bandwidth function at each grid node, giving the bandwidth vector $\boldsymbol{\sigma} = \big[\sigma(c)\big]_{c\in\boldsymbol{C}}$. Find the maximum $\sigma_{\max} = \max\{\sigma(c) : c \in \boldsymbol{C}\}$.
3. Compute the required number of steps $\tau = \lceil(\sigma_{\max})^2/D\rceil$ where $D = (1-\epsilon)\frac{(\Delta x)^2(\Delta y)^2}{(\Delta x)^2+(\Delta y)^2}$ if $v = 4$ and $D = (1-\sqrt{\epsilon})\min\{\Delta x, \Delta y\}^2$ if $v = 8$.
4. Calculate the column vector of single-step variances $\Delta\boldsymbol{t} = \big[\sigma(\boldsymbol{c})^2/\tau\big]_{\boldsymbol{c}\in\boldsymbol{C}}$.
5. Calculate column vectors of node-specific horizontal and vertical jump probabilities $\boldsymbol{p}_x = (\Delta\boldsymbol{t})/(2(\Delta x)^2)$ and $\boldsymbol{p}_y = (\Delta\boldsymbol{t})/(2(\Delta y)^2)$.
6. Replicate $\boldsymbol{p}_x, \boldsymbol{p}_y$ across rows to obtain square matrices $\tilde{\boldsymbol{p}}_x = \boldsymbol{p}_x 1_N^\top$, $\tilde{\boldsymbol{p}}_y = \boldsymbol{p}_y 1_N^\top$.
7. Compute transition matrix $\boldsymbol{P} = \boldsymbol{U} + \boldsymbol{I} - \mathrm{diag}(\boldsymbol{U}\boldsymbol{1}_N)$ where

$$\boldsymbol{U} = \begin{cases} \tilde{\boldsymbol{p}}_x \circ \boldsymbol{M}_{\mathrm{horz}} + \tilde{\boldsymbol{p}}_y \circ \boldsymbol{M}_{\mathrm{vert}} & \text{if } v = 4 \\ \tilde{\boldsymbol{p}}_x \circ (\boldsymbol{J} - 2\tilde{\boldsymbol{p}}_y) \circ \boldsymbol{M}_{\mathrm{horz}} + \tilde{\boldsymbol{p}}_y \circ (\boldsymbol{J} - 2\tilde{\boldsymbol{p}}_x) \circ \boldsymbol{M}_{\mathrm{vert}} + \tilde{\boldsymbol{p}}_x \circ \tilde{\boldsymbol{p}}_y \circ \boldsymbol{M}_{\mathrm{diag}} & \text{if } v = 8 \end{cases}$$

   where $\circ$ is the Hadamard product and $\boldsymbol{J} = 1_N 1_N^\top$ is the $N \times N$ matrix with all entries equal to 1.
8. Define the initial state as the $N \times 1$ vector $\boldsymbol{f}_0 = \boldsymbol{s}/A$. Recursively compute $\boldsymbol{f}_{k+1} = \boldsymbol{f}_k\boldsymbol{P}$ for $k = 1, \ldots, \tau - 1$. Return $\boldsymbol{f}_\tau$ as the intensity estimate.

C.3 Adaptive estimator, lagged-arrival algorithm

Algorithm 4 below specifies the implementation of the new observation-specific adaptive diffusion smoother. It is almost identical to the implementation of the fixed-bandwidth estimator described in Algorithm 2, except that data points are introduced progressively during the iterations.

Algorithm 4: Lagged-arrival algorithm for observation-specific diffusion smoothing.

Given the inputs $W, \boldsymbol{x}, \Delta x, \Delta y, v$ required for Algorithm 1, together with smoothing bandwidth values $\sigma_1, \ldots, \sigma_n$ and minimum stayput probability $\epsilon > 0$:

1. Discretise the data as described in Algorithm 1.
2. Construct the indicator vector for each discretised observation, $\boldsymbol{\zeta}_i = \big[\mathbb{1}\{x_i^* = c\}\big]_{c\in\boldsymbol{C}}$.
3. Find the maximum bandwidth $\sigma_{\max} = \max\{\sigma_1, \ldots, \sigma_n\}$.
4. Calculate the transition matrix $\boldsymbol{P}$ for fixed-bandwidth smoothing with bandwidth $\sigma = \sigma_{\max}$ by following steps 2–6 of Algorithm 2.
5. Associate each observation x_i with an arrival time $a_i \le \tau$ by $a_i = \max\left\{1, \tau - \lfloor\tau\left(\frac{\sigma_i}{\sigma_{\max}}\right)^2\rceil\right\}$ where $\lfloor x\rceil$ is the integer closest to x.
6. Define the initial state to be empty, $\boldsymbol{f}_0 = \boldsymbol{0}_N$. Find the final estimate $\widetilde{\boldsymbol{f}}_\tau$ by recursion with lagged arrivals:

$$\widetilde{\boldsymbol{f}}_\tau = \widetilde{\boldsymbol{f}}_{\tau-1}\boldsymbol{P} + A^{-1}\sum_{i=1}^{n}\mathbb{1}\{a_i = \tau\}\boldsymbol{\zeta}_i.$$

D Accuracy of discrete random walk approximation

The Euler scheme for numerical approximation of ordinary differential equations is described and analysed in Butcher (2003, Sections 20–21, pp. 55–89). It is classified as a first order method and is expected to have numerical errors of the same order as the grid step size (Butcher 2003, p. 73). Butcher (2003, p. 59) states that it is "not feasible to estimate the total accumulated error" of approximation in the Euler scheme; however, the following calculation gives an upper bound on the error for a special case.

Lemma 1 *Consider the pseudosymmetric random walk on the 8-connected infinite square grid, with space and time rescaled so that the grid spacing is Δx, the time step is Δt, and the variance per unit time is 1. Let $H_t(x, y)$ be the bivariate cumulative distribution function of the location of the random walk at time $t = \sigma^2$, given that it started at the origin. Then*

$$\sup_{x,y} |H_t(x/\sigma, y/\sigma) - \Phi(x)\Phi(y)| \leq \frac{\Delta c}{\sigma} \tag{A1}$$

where Φ is the cdf of the standard Normal distribution.

Proof 1 *Each step $(\Delta X, \Delta Y)$ in the random walk is such that ΔX and ΔY are independent random variables with the same distribution,*

$$\Delta X = \begin{cases} \Delta c & \text{with probability } p \\ -\Delta c & \text{with probability } p \\ 0 & \text{with probability } 1 - 2p \end{cases}$$

We have $\mathbb{E}(\Delta X) = 0$ and $\mathsf{var}(\Delta X) = 2p(\Delta c)^2 = a^2$, say. The assumption of unit variance per unit time implies that $a^2 = \Delta t$ so $p = (\Delta t)/(2(\Delta c)^2)$.

At time $t = \sigma^2$ there have been $n = t/\Delta t$ steps (assuming henceforth that t is an integer multiple of Δt) and the position of the random walk is $V_n = (X_n, Y_n)$ where X_n, Y_n are independent. Each of the coordinates X_n and Y_n is the sum of n independent copies of ΔX, and has mean zero and variance t. By independence $H_t(x, y) = F_n(x)F_n(y)$ where $F_n(x)$ is the cdf of the x-coordinate.

Applying the Berry-Esséen Theorem to F_n, we have

$$\sup_x |F_n(x/\sigma) - \Phi(x)| \leq C \frac{\mathbb{E}[|\Delta X|^3]}{n^{1/2}(\mathsf{var}(\Delta X))^{3/2}} \tag{A2}$$

where C is a universal constant, known to be less than $1/2$. By direct calculation $\mathbb{E}[|\Delta X|^3] = 2p(\Delta c)^3 = \Delta c\, \Delta t$ and since $\mathsf{var}(\Delta X) = \Delta t$ and $n = t/\Delta t = \sigma^2/\Delta t$, the bound (A2) *becomes*

$$\sup_x |F_n(x/\sigma) - \Phi(x)| \leq C\, \frac{\Delta c}{\sigma} \leq \frac{\Delta c}{2\sigma}. \tag{A3}$$

Returning to the bivariate distributions, we use the fact that, if $a_1, a_2, b_1, b_2, \epsilon$ are probabilities satisfying $|a_1 - a_2| < \epsilon$ and $|b_1 - b_2| < \epsilon$, then $|a_1 b_1 - a_2 b_2| < 2\epsilon$. Applying this bound with $a_1 = F_n(x/\sigma)$, $a_2 = \Phi(x)$, $b_1 = F_n(y/\sigma)$, $b_2 = \Phi(y)$ and $\epsilon = (\Delta c)/(2\sigma)$ yields the result.

This lemma provides an estimate of L_∞ error, at least in the infinite plane, and an argument why p should be made as large as possible subject to the constraints.

For example, consider approximating the cumulative distribution function of the heat kernel for the unit square W, with the source point located at the centre of the square, and bandwidth $\sigma = 0.1$, using the 8-connected grid with spacing $\Delta c = 1/128$. The rough bound (A1) above is $(\Delta c)/\sigma = (1/128)/0.1 = 0.0781$. Table SUP-3 of the supplementary material gives the corresponding actual value of L_∞ error as 0.019.

E Equations for a general diffusion

Here we collect some general results from stochastic process theory which were used (in special cases) in Section 6.2 of the main paper. Consider a diffusion $\boldsymbol{X}_t$ in d-dimensional space (for $d \geq 1$) with Itô stochastic differential equation

$$\mathrm{d}\boldsymbol{X}_t = \boldsymbol{b}(\boldsymbol{X}_t)\,\mathrm{d}t + \Sigma(\boldsymbol{X}_t)\,\mathrm{d}\boldsymbol{B}_t, \quad t \geq 0, \tag{A4}$$

where $\boldsymbol{b}(x) = (b_1(x) \dots b_d(x))$, $x \in \mathbb{R}^d$ is a d-dimensional vector valued function ("drift"), $\Sigma(x)$, $x \in \mathbb{R}^d$ is a $d \times m$ matrix-valued function and $\boldsymbol{B}_t$ is m-dimensional standard Brownian motion in $\mathbb{R}^m$, where $m \geq d$. Under conditions ensuring existence and differentiability (Gihman & Skorohod 1972, Theorems 3 and 5, pp. 288–300), the Kolmogorov forward equation for this diffusion is (Elliott 1982, eq. (18.5)) the Fokker-Planck equation

$$\frac{\partial}{\partial t} p_t(x \mid y) = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}[b_i(x) p_t(x \mid y)] + \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}[D_{ij}(x) p_t(x \mid y)], \tag{A5}$$

where p_t is the transition density, i.e. $p_t(\cdot \mid y)$ is the probability density of $\boldsymbol{X}_t$ given $\boldsymbol{X}_0 = y$. Here $D_{ij}(x)$ is the (i, j) entry of the $d \times d$ matrix $\boldsymbol{B}(x) = \frac{1}{2}\Sigma(x)\Sigma(x)^\top$. The diffusion kernel estimator with kernel $p_t(x \mid y)$ can be computed by solving (A5).

The Kolmogorov backward equation is (Elliott 1982, eq. (18.2); Gihman & Skorohod 1972, p. 297)

$$\frac{\partial}{\partial t} p_t(x \mid y) = \sum_{i=1}^{d} b_i(y) \frac{\partial}{\partial y_i} p_t(x \mid y) + \sum_{i=1}^{d}\sum_{j=1}^{d} D_{ij}(y) \frac{\partial^2}{\partial y_i \partial y_j} p_t(x \mid y). \tag{A6}$$

Bias in the diffusion kernel estimator for small bandwidth can be characterised using (A6) in the same manner as in Theorem 1 of Botev et al. (2010).

For this general diffusion, the formal definitions and results in Section 3 of the main paper are modified as follows. Redefine $\kappa_t(x \mid y)$ as the probability density of the state of this diffusion $\boldsymbol{X}_t$ at time t evaluated at query location x given the starting location was $\boldsymbol{X}_0 = y$. In Definition 1 of the main paper, the heat equation (4) is replaced by the forward equation (A5). Theorem 1 of the main paper remains true. In Lemma 1 of the main paper, properties (2)–(4) remain true.

References

Baddeley, A., Rubak, E. & Turner, R. (2015), *Spatial Point Patterns: Methodology and Applications with R*, Chapman and Hall/CRC, London.

Baddeley, A. & Turner, R. (2005), 'Spatstat: an R package for analyzing spatial point patterns', *Journal of Statistical Software* **12**(6), 1–42. URL: `www.jstatsoft.org`, ISSN: 1548-7660.

Bates, D. & Maechler, M. (2017), *Matrix: Sparse and Dense Matrix Classes and Methods*. R package version 1.2-12.
URL: *https://CRAN.R-project.org/package=Matrix*

Botev, Z., Grotowski, J. & Kroese, D. (2010), 'Kernel density estimation via diffusion', *Annals of Statistics* **38**(5), 2916–2957.

Boyce, W. & DiPrima, R. (1969), *Elementary Differential Equations and Boundary Value Problems*, second edn, Wiley, New York.

Butcher, J. (2003), *Numerical Methods for Ordinary Differential Equations*, third edn, Wiley, Chichester.

Elliott, R. (1982), *Stochastic Calculus and Applications*, Springer Verlag, New York.

Gihman, I. & Skorohod, A. (1972), *Stochastic Differential Equations*, Springer-Verlag, Berlin/New York. Trans. by K. Wickwire.

Pissanetzky, S. (1984), *Sparse Matrix Technology*, Academic Press.