On simulation of continuous determinantal point processes

Lavancier, Frédéric; Rubak, Ege

Published in:
Statistics and Computing

DOI (link to publication from Publisher):
10.1007/s11222-023-10272-w

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Lavancier, F., & Rubak, E. (2023). On simulation of continuous determinantal point processes. *Statistics and Computing*, *33*(45), Article 120. https://doi.org/10.1007/s11222-023-10272-w

Statistics and Computing (2023) 33:120
https://doi.org/10.1007/s11222-023-10272-w

ORIGINAL PAPER

On simulation of continuous determinantal point processes

Frédéric Lavancier[1,2] · Ege Rubak[3]

Received: 25 January 2023 / Accepted: 25 June 2023 / Published online: 19 August 2023

Abstract
We review how to simulate continuous determinantal point processes (DPPs) and improve the current simulation algorithms in several important special cases as well as detail how certain types of conditional simulation can be carried out. Importantly we show how to speed up the simulation of the widely used Fourier based projection DPPs, which arise as approximations of more general DPPs. The algorithms are implemented and published as open source software.

Keywords Spatial point process · Condition simulation · Ginibre process · Mercer decomposition · Prolate spheroidal functions

1 Introduction

Since the seminal work by Kulesza and Taskar (2012), discrete DPPs defined on a finite set $\{1, \ldots, N\}$ have become popular objects in the machine learning community to generate diverse random subsets, the main application being for recommandation systems. Discrete DPPs are defined through a kernel matrix of size $N \times N$, which is often assumed to be Hermitian. The default algorithm to make perfect simulation of discrete DPPs, the so-called spectral algorithm of Hough et al. (2006), starts from an orthonormal eigendecomposition of this matrix. Since this decomposition may be costly to obtain, and even unfeasible for large N, other options have been developed, from approximate algorithms (Affandi et al. 2013b; Li et al. 2016) to alternative perfect simulation strategies (Gillenwater et al. 2019; Derezinski et al. 2019; Poulson 2020; Launay et al. 2020).

On the other side, continuous DPPs generally refer to DPPs defined on the Euclidean space $\mathbb{R}^d$. They constitute the initial setting of (Macchi 1975) who introduced these processes in their current form to model the distribution of a fermion system in statistical physics. Continuous DPPs find applications in machine learning (Affandi et al. 2013a), spatial statistics (Lavancier et al. 2015), telecommunications (Deng et al. 2014; Miyoshi and Shirai 2014) and Monte Carlo approximations (Bardenet and Hardy 2020; Belhadji et al. 2019; Coeurjolly et al. 2021). They are defined through a kernel K which in this setting is a function $K(x, y)$, $x, y \in \mathbb{R}^d$, that must satisfy some properties to ensure existence of the continuous DPP as detailed in Sect. 2. The spectral algorithm of Hough et al. (2006) to simulate continuous DPPs on a compact set $S \subset \mathbb{R}^d$ is in general feasible only if we know the spectral representation (also called Mercer representation) of K on S. While this representation always exists in theory, it is in general intractable. Unfortunately, the aforementioned alternative perfect simulation strategies developed in the discrete case do not apply to the continuous case, and their extension to this setting is a difficult open challenge. For these reasons, there is still an avenue to improve simulation of continuous DPPs. This paper addresses this question for projection DPPs and other important classes of DPPs, including the β-Ginibre process, the Gaussian-type DPP and the Bessel-type DPP.

A projection DPP on a bounded subset $S \subset \mathbb{R}^d$ corresponds to the special case where K satisfies the identity $\int_S K(x, z)K(z, y)dz = K(x, y)$. This class of DPPs is of particular interest for several reasons. First, unlike general DPPs, projection DPPs generate a fixed number of points in S, equal to $n = \int_S K(x, x)dx$. Second, they correspond to the most repulsive DPPs, since their (non null) eigenvalues in

✉ Frédéric Lavancier
Frederic.Lavancier@univ-nantes.fr;
Frederic.Lavancier@ensai.fr

✉ Ege Rubak
rubak@math.aau.dk

[1] Laboratoire de Mathématiques Jean Leray, BP 92208, 2 Rue de la Houssinière, 44322 Nantes Cedex 03, France

[2] CREST-ENSAI, UMR CNRS 9194, Campus de Ker-Lann, Rue Blaise Pascal, BP 37203, 35172 Bruz cedex, France

[3] Department of Mathematical Sciences, Aalborg University, Aalborg, Denmark

their spectral representation are all identically equal to 1, the maximal possible value. Third, the simulation algorithm of Hough et al. (2006) for a general DPP, provided its spectral representation is known, boils down to the simulation of an associated projection DPP. A first contribution of this paper, carried out in Sect. 3, is to review and improve the simulation algorithms of projection DPPs. An important particularity of these processes is that they can be perfectly simulated without knowing the spectral representation of their kernel; see Algorithm 1. Nonetheless, if we have access to this representation, then the well-known spectral algorithm (Algorithm 2) is more efficient. As a special case, we give a particular account of the case when the eigenfunctions of the spectral decomposition are the Fourier basis functions on S. This case plays an important role both because it gives rise to an homogeneous projection DPP on S and because it is at the heart of the spectral approximation of general invariant-translation kernels in Lavancier et al. (2015). For this basis, we refine the spectral algorithm to decrease the computation time by up to 45% in dimension $d = 1$ and up to 25% in $d = 2$; see Algorithm 3. Finally, we show how Algorithm 1 and Algorithm 2 can be combined to perform conditional simulation of a projection DPP given a subset of points on S, including in-painting conditional simulation.

For general (non projection) DPPs, the standard simulation procedure consists in approximating the spectral representation of their kernel in order to apply the algorithm of Hough et al. (2006); see Lavancier et al. (2015); Affandi et al. (2013a). We do not discuss these kinds of approximation here, but as a second contribution, we rather focus on important classes of DPPs for which the spectral representation can be obtained explicitly on some subset S. In particular, we investigate in details in Sect. 4 the β-Ginibre process (Goldman 2010; Deng et al. 2014), a generalisation of the famous standard Ginibre process (Ginibre 1965). For this model, we provide a simulation algorithm based on the exact spectral representation of the kernel, and we compare it to the simulation based on the eigenvalues of a random matrix (sometimes referred to as the truncated β-Ginibre approximation). Moreover, we also derive in Sect. 5 the spectral representation of Gaussian kernels and Bessel-type kernels, two widely used parametric models in spatial statistics, opening the way of perfect simulation for these models.

All algorithms presented in Sects. 3 and 4 are available in an R-package (R Core Team 2022) currently available on https://github.com/rubak/dppsim and planned for submission to the Comprehensive R Archive Network (CRAN). The package includes functions for simulation and conditional simulation of any projection DPP, and simulation of the β-Ginibre process by the spectral algorithm and the truncated β-Ginibre approximation.

2 Preliminaries

Throughout the manuscript the notation $|.|$ has different meanings: for a complex number $z \in \mathbb{C}$, $|z|$ denotes its modulus; for a vector x, $|x|$ is its infinite norm; if J is a finite set, then $|J|$ stands for its cardinality; if S is an infinite set, then $|S|$ is its volume. On the other hand, $\|.\|$ will either refer to the vectorial ℓ^2 norm, or to the functional L^2 norm, depending on the context. For two vectors x and y, $x \cdot y$ denotes their inner product. For a matrix A, we denote by A' its transpose and by A^* its conjugate transpose.

2.1 DPPs and basic properties

Let X be a point process on $\mathbb{R}^d$ (we refer to Møller and Waagepetersen (2004) for background material on spatial point processes). If there exists a non-negative function $\rho_n : (\mathbb{R}^d)^n \to \mathbb{R}$ such that

$$\mathbb{E} \sum_{x_1,\ldots,x_n \in X}^{\neq} f(x_1,\ldots,x_n) = \int_{(\mathbb{R}^d)^n} f(x_1,\ldots,x_n) \rho_n(x_1,\ldots,x_n)\mathrm{d}x_1 \cdots \mathrm{d}x_n,$$

for all locally integrable functions $f : (\mathbb{R}^d)^n \to \mathbb{R}$, where the symbol $\neq$ means that the sum is done for distinct x_i, then ρ_n is called the n-th order joint intensity function of X. Intuitively, $\rho_n(x_1,\ldots,x_n)$ is the infinitesimal probability that X contains points (among others) located at $x_1,\ldots,x_n$.

We say that X is a DPP on $S \subseteq \mathbb{R}^d$ with kernel $K : S \times S \to \mathbb{C}$ if for all $n \geq 1$, the joint intensity ρ_n of X exists and is of the form

$$\rho_n(x_1,\ldots,x_n) = \det[K(x_k,x_l)]_{1\leq k,l\leq n} \qquad (1)$$

for all $\{x_1,\ldots,x_n\} \subset S$, where $[K(x_k,x_l)]_{1\leq k,l\leq n}$ denotes the matrix with entries $K(x_k,x_l)$.

Some conditions on K are necessary to ensure the existence of X, which depend on whether X is defined on all of $\mathbb{R}^d$ or a compact subset $S \subset \mathbb{R}^d$. First, assume that S is a compact set and that K is Hermitian and continuous on $S \times S$, then by Mercer's Theorem it admits the spectral representation

$$K(x,y) = \sum_{k\geq 1} \lambda_k \Phi_k(x)\overline{\Phi}_k(y), \quad x,y \in S, \qquad (2)$$

for some eigenvalues λ_k and orthonormal eigenfunctions Φ_k on S, for $k = 1,2,\ldots$. (In typical applications $k \in \mathbb{Z}^d$, but for convenience we use a natural number index here.) A necessary and sufficient condition for existence of a DPP X on S with kernel K is that $\lambda_k \in [0,1]$ for all $k = 1,2,\ldots$. We refer the reader to Hough et al. (2006) for details and more results. Second, assume that K is invariant under translation on $\mathbb{R}^d$, i.e. $K(x, y) = K_0(x - y)$ for some function K_0, then a sufficient condition for existence of the DPP with kernel K on $\mathbb{R}^d$ (and then on any subset $S \subset \mathbb{R}^d$) is that K_0 is a continuous covariance function on $\mathbb{R}^d$ whose Fourier transform is less than 1; see Lavancier et al. (2015). Note that to check the latter condition it is not necessary to know the spectral representation (2) of K on any subset S. In the following, we will assume that the considered DPPs exist (conditions will be recalled for each specific example).

Let us recall some important properties of DPPs (Lavancier et al. 2015).

(i) The (first order) intensity of a DPP X on S is by definition $\rho(x) = K(x, x)$, $x \in S$. We say that X is homogeneous if $\rho(x)$ is constant.

(ii) Its pair correlation function, defined by $g(x, y) = \rho_2(x, y)/(\rho(x)\rho(y))$, reads

$$g(x, y) = 1 - \frac{|K(x, y)|^2}{K(x, x)K(y, y)}, \quad x, y \in S,$$

whenever K is Hermitian, which we assume henceforth. The fact that $g(x, y) \leq 1$ reflects that DPPs are models for repulsiveness in a point pattern.

(iii) Suppose we apply an independent thinning of X with retention probabilities $p(x)$, $x \in S$, then the resulting process is still a DPP with kernel $K(x, y)\sqrt{p(x)p(y)}$. In particular, the restriction of X to any subset $S' \subset S$ is the DPP with kernel $K(x, y)\mathbf{1}_{x\in S'}\mathbf{1}_{y\in S'}$.

(iv) Any smooth transformation of X remains a DPP. For example, let $T(x) = Ax + b$, $x \in \mathbb{R}^d$, be an affine transformation on $\mathbb{R}^d$ with $\det(A) \neq 0$, then $T(X)$ is the DPP on $T(S)$ with kernel $K(A^{-1}(x - b), A^{-1}(y - b))/|\det(A)|$.

As mentioned in the introduction, an important special class of DPPs is the class of projection DPPs on bounded $S \subset \mathbb{R}^d$. Their kernel K satisfies $\int_S K(x, z)K(z, y)dz = K(x, y)$ and they generate a fixed number of n points in S, where $n = \int_S K(x, x)dx$. Their eigenvalues in (2) take only two possible values, 0 or 1, exactly n of them being 1, the others being 0.

Assume now that all eigenvalues of K in (2) are strictly less than 1. Then, the probability density that X has n points located at $x_1, \ldots, x_n$ (sometimes called likelihood or Janossy density of X) is proportional to $\det[L(x_k, x_l)]_{1\leq k,l\leq n}$, where L is the kernel defined by

$$L(x, y) = \sum_{k\geq 1} \frac{\lambda_k}{1 - \lambda_k} \Phi_k(x)\overline{\Phi}_k(y), \quad x, y \in S.$$

It is common in the machine learning community to define X from this L-kernel, instead of its K-kernel, meaning that X is defined through its Janossy densities instead of its joint intensities ρ_n. If $\lambda_k < 1$ for all k, both point of views are equivalent and the spectral representation of K can be deduced from that of L, and vice-versa. On one hand, the advantage of defining X from L is that the eigenvalues of L are not restricted to be less than 1 for X to be well defined and the likelihood function is easier to understand. On the other hand, defining X through L implies that the moments of X become unknown, contrary to (i) and (ii) above. Moreover some DPPs, including projection DPPs for which $\lambda_k \in \{0, 1\}$, cannot be defined through a L-kernel. Note however that in Tremblay et al. (2023), extended L-ensembles are introduced and fill this gap (the main setting of that paper concerns discrete DPPs but it is is argued in conclusion that the extension to the continuous setting is straightforward). In this paper we start from the K-kernel and we consider the simulation of X given K.

2.2 Overview of simulation methods

2.2.1 Projection DPPs

For projection DPPs on a compact set $S \subset \mathbb{R}^d$, perfect simulation algorithms are available, whatever the spectral representation of the kernel is known or not. We detail these algorithms in Sect. 3.1. The idea is to generate the first point of the DPP with respect to the unnormalised density $K(x, x)$ on S, then to generate the second point given the first one with respect to the associated conditional density on S, and so on. The difficulty in this procedure is to be able to simulate, at each step, a point with respect to the conditional density given the previous points. Figure 1 shows examples of such conditional densities at intermediate steps of the algorithm when simulating a homogeneous projection DPP having 121 points on the unit square with $K(x, y) = \sum_{|j|\leq 5} e^{ij\cdot(x-y)}$ for $x, y \in [0, 1]^2$ and $j \in \mathbb{Z}^2$ (where $i = \sqrt{-1}$ and the unit square in $\mathbb{R}^2$ is identified with the unit square in $\mathbb{C}$). The top left hand plot is the density of the 20th point given the 19 first points while the top right hand plot shows the density of the last (121st) point given the first 120 points. The bottom row plots are the same plots on log scale. The simulation with respect to these conditional densities is commonly achieved by rejection sampling where the proposal is the unnormalised density $K(x, x)$. This procedure can be extremely costly in the last steps of the algorithm where many rejections may occur due to the complicated shape of the conditional density.

Currently there seems to be no general way to avoid the costly rejection sampling step, but in Sect. 3.2 we show how to significantly accelerate it in the important particular case where the eigenfunctions of K correspond to the Fourier basis

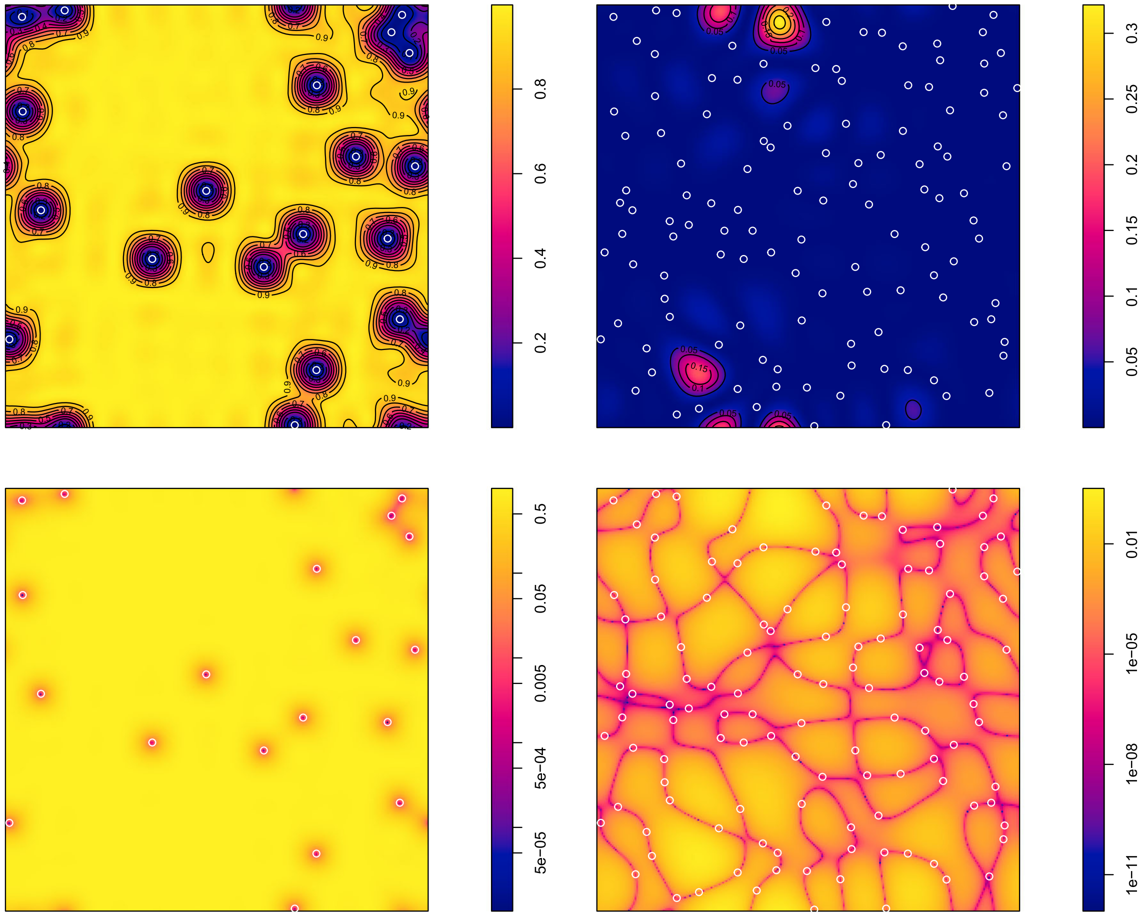

Fig. 1 Conditional density p_i at intermediate steps of Algorithms 1 and 2, for the Fourier basis and $n = 121$ on the unit square. Top row is untransformed values while the bottom row is in log scale. Left: density of the 20th point given the 19 first ones (superimposed as white circles); right: density of the last point given the preceding 120 points

functions. Moreover, the fact that it is possible to simulate a projection DPP without knowing its spectral representation opens the way to perform conditional simulation, including in-painting, as we discuss it in Sect. 3.3.

2.2.2 Non-projection DPPs

If the DPP is not a projection DPP, then the standard algorithm to generate it on S, due to Hough et al. (2006), starts from the spectral representation (2) of K on S. It consists in first, generating a sequence of Bernoulli variables B_k with respective rates λ_k (note that only a finite number of B_k's are non-zero since $\sum_{k\geq 1} \lambda_k < \infty$), and second, given $(B_k)_{k\geq 1}$, generating the projection DPP with kernel $\sum_{k\geq 1} B_k \Phi_k(x)\overline{\Phi}_k(y)$.

In Sects. 4 and 5, we show that the exact spectral representation (2) of the β-Ginibre kernel, the Gaussian kernel, and the Bessel-type kernel is available on some sets S, opening the possibility of perfect simulation for the associated DPPs on these sets (and any subset thereof). We in particular treat in detail the β-Ginibre process, for which we provide efficient simulation algorithms in Sect. 4.

However, for most kernels and sets S, the expansion (2) is not known. To overcome this issue we may:

- Consider an approximation of (2). For any translation invariant kernel on a rectangular set S, a Fourier series approximation is proposed in Lavancier et al. (2015), and when the DPP is defined through the L-kernel, a Nyström based approximation is suggested in Affandi et al. (2013a).
- Using property (iv) of the previous section, consider a smooth transformation $Y = T(X)$, where the spectral representation of Y on $T(S)$ is known or can be approximated, then generate Y on $T(S)$ and deduce $X = T^{-1}(Y)$. In particular, by this property, the simulation of a DPP on any rectangular (resp. any ellipsoid) set S boils down to the simulation on the unit square (resp. on the unit disc).

- Using property (iii), view X as an independent thinning of some DPP Y whose spectral representation is known or can be approximated, then generate Y and get X by thinning. This includes in particular the simulation of X on any subset S' of S, provided we know how to generate X on S. This strategy is also possible for any non-stationary DPP X whose kernel reads $K_0(x-y)\sqrt{\rho(x)\rho(y)}$ with $K_0(0) = 1$, provided $\rho(x)$ is uniformly bounded on S.

If X is defined through its L-kernel, the simulation challenge is similar: one needs to know or approximate the spectral representation of L in order to deduce K and use the procedure above. Let us however mention an alternative solution that does not require the spectral representation of L: knowing L on S allows one to have access to the Papangelou intensity of X on S. This opens the possibility to use a CFTP algorithm to simulate perfectly X on S, provided $\int_S L(x,x)dx < \infty$, as considered in (Decreusefond et al. 2016, Section 5.2). Unfortunately this procedure can be very slow.

3 Simulation of a projection DPP

3.1 Basic algorithms

Assume that $X \sim DPP(K)$ is a projection DPP with cardinality n on a compact set $S \subset \mathbb{R}^d$. Then we can simulate X on S with Algorithm 1 below. This algorithm is justified in Hough et al. (2006); see also (Lyons 2014, Section 3.5) and "Appendix A.1". Note that explicit knowledge of the spectral representation (2) of K is not required. This is useful in some situations where we know that $DPP(K)$ exists but no spectral representation, and even no accurate approximation of it, is available for K. This will be the case for the kernel K involved in the inpainting conditional simulation of Sect. 3.3.

Algorithm 1 Projection DPP without using the spectral representation

```
sample $X_n$ from the distribution with density $p_n(x) = K(x,x)/n$, $x \in S$
for $i = (n-1)$ to 1 do
    set $\mathbf{k}_i^*(x) = (K(x,X_n),\dots,K(x,X_{i+1}))$ and $\mathbf{K}_i = (K(X_j,X_l))_{n \geq j,l \geq i+1}$
    sample $X_i$ from the distribution with density

    $p_i(x) = \frac{1}{i}\left[K(x,x) - \mathbf{k}_i^*(x)\mathbf{K}_i^{-1}\mathbf{k}_i(x)\right], \quad x \in S$    (3)

end for
return $\{X_1,\dots,X_n\}$
```

On the other hand, if we have access to the spectral representation of K, the conditional density p_i in (3) can be recast to get the more efficient Algorithm 2. Some justification for this is given in "Appendix A.1" and more details can be found in Lavancier et al. (2015). Here, we assume without loss of generality that the spectral representation of K reads

$$K(x,y) = \sum_{k=1}^{n} \Phi_k(x)\bar{\Phi}_k(y), \quad x, y \in S, \tag{4}$$

and we set $\boldsymbol{v}(x) = (\Phi_1(x),\dots,\Phi_n(x))'$.

Algorithm 2 Projection DPP using the spectral representation (4)

```
sample $X_n$ from the distribution with density $p_n(x) = \|\boldsymbol{v}(x)\|^2/n$, $x \in S$
set $\boldsymbol{e}_1 = \boldsymbol{v}(X_n)/\|\boldsymbol{v}(X_n)\|$
for $i = (n-1)$ to 1 do
    sample $X_i$ from the distribution with density

    $p_i(x) = \frac{1}{i}\left[\|\boldsymbol{v}(x)\|^2 - \sum_{j=1}^{n-i}|\boldsymbol{e}_j^*\boldsymbol{v}(x)|^2\right], \quad x \in S$    (5)

    set $\boldsymbol{w}_i = \boldsymbol{v}(X_i) - \sum_{j=1}^{n-i}\left(\boldsymbol{e}_j^*\boldsymbol{v}(X_i)\right)\boldsymbol{e}_j$, $\boldsymbol{e}_{n-i+1} = \boldsymbol{w}_i/\|\boldsymbol{w}_i\|$
end for
return $\{X_1,\dots,X_n\}$
```

To appreciate why the computation of p_i is faster with Algorithm 2 than with Algorithm 1, take for instance the evaluation of the second term in (3). For each proposed x it requires the computation of $\mathbf{k}_i(x)$ that itself requires $n-i$ evaluations of the vector $\boldsymbol{v}(x)$ along with the $n-i$ vectors $\boldsymbol{v}(X_j)$ for $i+1 \leq j \leq n$. In contrast, only one evaluation of $\boldsymbol{v}(x)$ is required for the same term in Algorithm 2 since the $\boldsymbol{e}_j$'s are stored from the previous step.

From a practical point of view, we need to be able to generate a point with respect to p_i, whether we use Algorithm 1 or 2. This is in fact the bottleneck of these simulation methods. The standard procedure consists in rejection sampling, where the proposal can be deduced from the bound $ip_i(x) \leq K(x,x)$. Two strategies are then possible:

- Strategy 1: If we know how to simulate with respect to the distribution on S with density $K(x,x)/n$, then this distribution can be chosen as the proposal. Specifically, if Z is a point generated by this proposal and $U \sim \mathcal{U}([0,1])$ independent of Z, we accept Z if $ip_i(Z)/K(x,x) > U$. The probability of acceptance is i/n.
- Strategy 2: If we only know an upper-bound for $K(x,x)$, i.e. $K(x,x) \leq M$ for all $x \in S$, then $p_i(x) \leq M/i$ and rejection sampling can be applied where the proposal is the uniform distribution on S. Specifically, if Z is generated uniformly on S and $U \sim \mathcal{U}([0,1])$ independent

of Z, we accept Z if $ip_i(Z)/M > U$. The probability of acceptance is $i/(M|S|)$.

The second strategy is always an option in practice, provided we initially approximate the upper-bound M numerically. Note that given S and K, this bound can be found offline and once for all.

Remark 3.1 A natural and commonly used idea in spatial statistics consists in leveraging the local contribution of a point process to generate a new point given the other ones. Unfortunately, the conditional density p_i is not well localised, especially in presence of many conditional points, which corresponds to the smallest values of i or equivalently to the last (most time consuming) steps of the simulation algorithms, ruling out the idea of localisation. This is clearly illustrated in Fig. 1: While in the first steps of the algorithm (left hand plots) p_i seems to be localised around the conditional points, the picture looks different in the right hand plots, where some "empty" regions are associated to very small values of the density p_i, whereas other seemingly similar regions are associated with a high density. This effect is particularly evident in the bottom right hand plot on log scale, that shows quite stunning long range dependencies in p_i. This demonstrates that not only the neighbour points of these regions impact the values of p_i, but really the whole point pattern.

3.2 Refinement for the Fourier basis

The Fourier basis plays an important special role in continuous DPPs for its simplicity, the homogeneity and the repulsiveness it yields (Coeurjolly et al. 2021), and the fact that it is at the heart of the spectral approximation of invariant-translation kernels in Lavancier et al. (2015). Let us assume for simplicity that $S = [0, 1]^d$ (remember that the simulation on any rectangular windows boils down to this case). The kernel of a projection DPP on S based on the Fourier basis reads

$$K(x, y) = \sum_{j \in J} \mathrm{e}^{2\mathrm{i}\pi j \cdot (x-y)},$$

where $J \subset \mathbb{Z}^d$ is some finite subset of $\mathbb{Z}^d$ with cardinality $|J| = n$.

In this case, the more efficient Algorithm 2 is feasible and the two aforementioned strategies for the simulation with respect to p_i are similar since $K(x, x) = n$ (then $M = n$ in Strategy 2). The Algorithm builds on rejection sampling with the uniform distribution as a proposal. The costly part in this procedure is the evaluation of $p_i(Z)$ for any proposed point Z generated from the uniform distribution on S, which becomes all the more problematic in the last steps of the i-loop in Algorithm 2 when many rejections are expected, the probability of acceptance then being of order $1/n$.

To accelerate the procedure, we suggest to exploit the following bound; see "Appendix B":

$$ip_i(x)/n \le \min_{i+1 \le k \le n} (1, P(x - X_k)/n), \tag{6}$$

where

$$P(x) = 4\pi^2 \sum_{j \in J} (j \cdot x)^2 - \frac{4\pi^2}{n} \left(\sum_{j \in J} j \cdot x \right)^2$$

is a second-order polynomial depending on d-variables. Note that the coefficients of P only depend on the index set J and can thus be computed offline before running Algorithm 2. The inequality (6) provides a simple first bound to test against in the rejecting sampling procedure, avoiding the more costly evaluation of p_i for many proposed points. The refined rejection sampling procedure is detailed in Algorithm 3.

Algorithm 3 Refined rejection sampling for p_i with the Fourier basis

```
repeat
  sample $Z$ from the uniform distribution on $S$
  generate $U \sim \mathcal{U}([0,1])$ independent of $Z$
  if $\min_{i+1 \le k \le n} P(Z - X_k)/n < U$ then
    reject $Z$
  else if $ip_i(Z)/n > U$ then
    accept $Z$
  else
    reject $Z$
  end if
until $Z$ is accepted
```

We tested the performance of this refinement by simulation. Table 1 reports the fraction of time needed to generate different point patterns in dimension $d = 2$ by using Algorithm 3 compared to using simple rejection sampling. The entries are averaged over 500 replications. The tested models are first, the projection DPP with kernel $K(x, y) = \sum_{|j| \le \ell} e^{ij \cdot (x-y)}$, that leads to a cardinality of $n = (2\ell + 1)^2$ points, which we call "most repulsive" kernel with intensity $\rho = n$ in Table 1, and second, the Gaussian-type DPP (see Sect. 5.1) with a comparable intensity $\rho = n$ and an interaction parameter $\alpha = \alpha_{\max}$ (strong repulsion) and $\alpha = \alpha_{\max}/2$ (mild repulsion). Here $\alpha_{\max} = 1/\sqrt{\rho\pi}$ is the maximal possible value of α for a Gaussian-type DPP with intensity ρ, and the simulation exploits the Fourier approximation of (2) as introduced in Lavancier et al. (2015), making the use of Algorithm 3 relevant. Table 1 also reports the "bound rate" which summarises how often the bound (6) was used on average. This rate was empirically quite stable across the different

Table 1 For three DPP models (detailed in the text), generated 500 times on $S = [0, 1]^2$: fraction of rejections due to (6) when using Algorithm 3 (Bound rate) and fraction of time needed for the simulation using the refinement of Algorithm 3 compared with not using it, for different values of the intensity ρ of the DPP models

DPP model	Bound rate	Intensity (ρ)				
		25	81	289	625	1089
Most repulsive	0.41	1.10	1.00	0.96	0.77	0.72
Gauss ($\alpha = \alpha_{\max}$)	0.24	0.99	0.93	0.93	0.85	0.85
Gauss ($\alpha = \alpha_{\max}/2$)	0.06	1.02	0.97	0.99	0.99	0.96

values of ρ that we implemented and this is why only a single number is given. Specifically, a bound rate of 0.40 means that 40% of the rejections in the algorithm was done directly thanks to the bound (6). This rate can be viewed as the maximal theoretical gain we can expect thanks to Algorithm 3, if the only computation time was due to this rejection step in Algorithm 2.

From Table 1, we conclude that using Algorithm 3 in dimension $d = 2$ may lead to a speed improvement up to almost 30% (for the most repulsive kernel and about 1000 points), and never really slows down the simulation even when no clear improvement can be expected (as for the mild repulsive Gaussian-type DPP for which the bound rate is only 0.06). We have carried out similar simulations in dimension $d = 1$, not detailed here, where we observed a bound rate of 65% and a gain up to 45% for the most repulsive kernel with 1000 points. In dimension $d = 3$, comparable simulations showed a bound rate of 25% and a speed improvement up to 16% for the most repulsive kernel with 1000 points. We expect the improvements to be smaller in higher dimensions, making the use of Algorithm 3 superfluous in this case.

Finally, let us mention an additional possible refinement in dimension $d = 1$. In this case, the bound (6) can be manipulated to provide a more efficient proposal than the uniform distribution, that can be easily simulated by the inversion method. The details are given in "Appendix B.1". With this new proposal, the acceptance probability in the rejection sampling for the last step of Algorithm 2, i.e. for the simulation of the last point given the $n - 1$ first ones, can be 3 times higher than one associated to a uniform proposal; see "Appendix B.1". Unfortunately, this procedure does not generalise easily to higher dimensions, in the sense that it seems difficult to derive a similar efficient proposal from (6) in dimension $d > 1$.

3.3 Conditional simulation

Let again X be a projection DPP on $S \subset \mathbb{R}^d$ with cardinality n.

By construction, the last i steps of Algorithms 1 and 2 provide an exact simulation of $X \setminus \{X_n, \ldots, X_{i+1}\}$ given that $\{X_n, \ldots, X_{i+1}\} \subset X$. This conditional simulation may be useful in practice if we observe a point pattern in S with missing points and we wish to generate some realisations of the full point pattern in S. This is possible if we agree that the complete unobserved point pattern can be modelled by a projection DPP, to be chosen, and that the missing points are due to an independent thinning with known retention probability $q(x)$, $x \in S$.

As an example, assume that X is a projection DPP on $S = [0, 1]^2$ with kernel $K(x, y) = \sum_{|j| \le \ell} e^{ij\cdot(x-y)}$, so cardinality $n = (2\ell + 1)^2$, where we need to choose ℓ (or equivalently n), and we observe only m points of X, $m \le n$, coming from an independent thinning of X with known retention probability $q(x)$, $x \in S$. Using the relation $\mathbb{E}(m) = \int_S q(x)K(x, x)dx = n \int_S q(x)dx$, we deduce a plausible value of n by replacing $\mathbb{E}(m)$ by the observed value of m. Then we can use Algorithm 2 to generate the $n - m$ remaining points given the m observed ones. This example is illustrated in the leftmost plot of Fig. 2 where the red triangles represent a possible realisation of the unobserved points in $S = [0, 1]^2$ given the observed point pattern formed by the black circles. Here we assumed that $q(x) = 1/2$, so that n is twice the number of observed black points.

We chose in the above example a specific form of K, up to n, that may not be relevant for other applications. A general procedure to select a projection DPP X that fits well with the observed conditional points is discussed in "Appendix A.3".

Alternatively, we may consider in-painting conditional simulation, that is the simulation of $X \cap A$ given that $X \cap A^c = \{X_1, \ldots, X_m\}$, $m \le n$, where A and A^c constitute a partition of S (Here we agree that if $m = 0$, $X \cap A^c = \emptyset$.) Since X is a projection DPP, it is easily verified (see "Appendix A.2") that the latter conditional point process is a projection DPP with cardinality $n - m$ and kernel, when $m > 0$:

$$K_{X_1,\ldots,X_m}(x, y) = K(x, y) - \mathbf{k}^*_m(x)\mathbf{K}_m^{-1}\mathbf{k}_m(y), \quad x, y \in A, \tag{7}$$

where $\mathbf{k}^*_m(x) = (K(x, X_1), \ldots, K(x, X_m))$ and $\mathbf{K}_m = (K(X_j, X_l))_{1\le j,l\le m}$. If $m = 0$, $K_\emptyset(x, y) = K(x, y)\mathbf{1}_A(x)$ $\mathbf{1}_A(y)$. The spectral representation of the conditional kernel (7) is unlikely to be known, even if the spectral representation

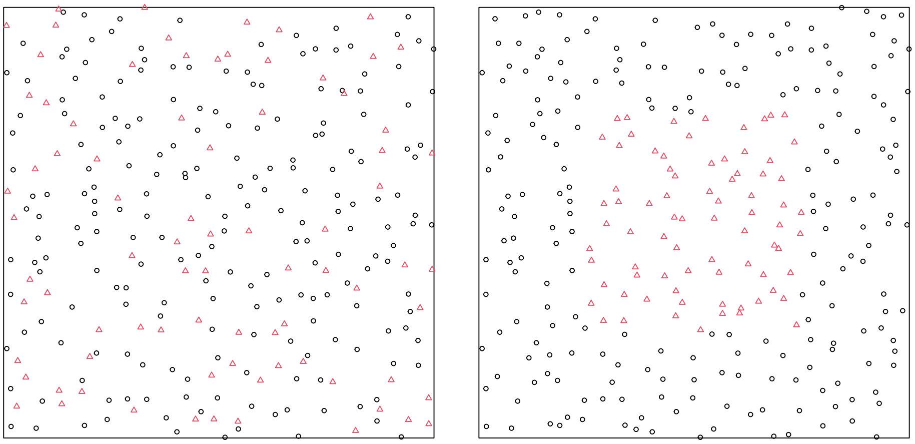

Fig. 2 Conditional simulation of the red points (shown as triangles) given the black points, assuming that the full point pattern formed by the union of the black and red points is a projection DPP. Left: simulation in the whole domain. Right: simulation in the inner square (in-painting)

of K on S is explicitly given. Nonetheless the simulation can be done by Algorithm 1.

As an illustration, we have generated in the rightmost plot of Fig. 2 the red triangles in the inner square $[1/4, 3/4]^2$ given the point pattern in black circles outside this square. The full point pattern was assumed to be the same DPP as before, where n was fixed to the number of observed black circles times $4/3$ (that is the ratio of the volume of the full region to the volume of the observed region). A more adaptive choice of K based on the observed points is discussed in "Appendix A.3".

4 β-Ginibre process

The β-Ginibre process, for $\beta > 0$, is the DPP in the complex plane $\mathbb{C}$ with kernel

$$K(x, y) = \rho \exp\left(\frac{x\bar{y}}{\beta} - \frac{|x|^2 + |y|^2}{2\beta}\right), \quad x, y \in \mathbb{C}. \qquad (8)$$

Note that we could equivalently consider this process in $\mathbb{R}^2$ but it is commonly defined in $\mathbb{C}$, a point of view which we will respect in the following. The β-Ginibre process is a stationary and isotropic point process with intensity $\rho > 0$, that exists if and only if $\rho\beta\pi \le 1$. The standard Ginibre process introduced in Ginibre (1965) corresponds to the particular case $\rho = 1/\pi$ and $\beta = 1$. The β-Ginibre process, initially considered in Goldman (2010) and Deng et al. (2014), can be viewed as an independent thinning of the Ginibre process with retention probability $\rho\beta\pi$, followed by the homothety with ratio $\sqrt{\beta}$. Given $\rho > 0$, the β-Ginibre process includes the Poisson point process with intensity ρ as a limiting case when $\beta \to 0$, and when $\beta = 1/(\rho\pi)$, it is one of the most repulsive stationary DPPs with intensity ρ in the sense of Møller and O'Reilly (2018).

We discuss and compare in this section two strategies to generate the β-Ginibre process on the ball $\mathbf{B}_R = \{|x| < R\}$ in $\mathbb{C}$, for any $R > 0$. Remember that the simulation on any compact set $S \subset \mathbf{B}_R$ can then be performed by removing the points in $\mathbf{B}_R \setminus S$. The first strategy is a standard procedure that takes advantage of the representation of the Ginibre process as the distribution of certain eigenvalues of an infinite matrix. The second strategy exploits the (unusual) fact that the spectral decomposition (2) of K is explicitly known on $\mathbf{B}_R$, and so the spectral algorithm of Hough et al. (2006) can be employed.

Remark 4.1 The second order properties of the Ginibre process, i.e. the intensity and the pair correlation function, are similar to those of the Gaussian-type DPP with parameters ρ and α (see Sect. 5.1) with the correspondence $\beta = \alpha^2/2$. Note however that the Ginibre process can reach more repulsiveness in view of the existence condition $\beta \le 1/(\rho\pi)$, while $\alpha^2/2 \le 1/(2\rho\pi)$ is required for existence of the Gaussian-type DPP.

4.1 Approximation by the truncated β-Ginibre process

The Ginibre process can be viewed as the distribution of the eigenvalues of an infinite matrix with i.i.d. standard complex Gaussian entries (Ginibre 1965). To simulate (approximately) the Ginibre process on $\mathbf{B}_R$, the standard procedure consists of generating a finite $n \times n$ matrix of i.i.d. standard complex Gaussians, where n has to be chosen according to R, and then computing its n complex eigenvalues. The simulation of the β-Ginibre process is easily deduced by the thinning and rescaling steps explained in the beginning of this section. Given the importance of this procedure, we provide some details below and we discuss the choice of n.

The distribution of the n eigenvalues above (followed by the thinning and scaling steps) corresponds to the distribution of the truncated β-Ginibre process, which is the DPP on $\mathbb{C}$ whose kernel is the truncation at $k = n - 1$ of (11) defined in Sect. 4.2 below, that is

$$K_n(x,y)=\sum_{k=0}^{n-1}\rho\beta\pi\,\Phi_k(x)\overline{\Phi}_k(y),\quad x,y\in\mathbb{C}. \tag{9}$$

We refer to Decreusefond et al. (2015) for some discussion of this process and on other modified versions of the Ginibre process. As already noticed in Ginibre (1965), this DPP is not stationary and is mostly concentrated on the disc with radius $\sqrt{n\beta}$. Inside this disc the truncated β-Ginibre process and the genuine β-Ginibre process are very similar, their differences arising mostly close to the border. In particular the intensity of the truncated β-Ginibre process reads

$$\rho(x)=K_n(x,x)=\rho e^{-|x|^2/\beta}\sum_{k=0}^{n-1}\frac{(|x|^2/\beta)^k}{k!},\quad x\in\mathbb{C}.$$

This intensity is always less than ρ (the intensity of the β-Ginibre process) and we have for any $|x|\leq\sqrt{n\beta}$ (see Ginibre (1965)):

$$\begin{aligned}0\;&\leq\frac{\rho-\rho(x)}{\rho}\leq e^{-|x|^2/\beta}\frac{(|x|^2/\beta)^n}{n!}\,\frac{n+1}{n+1-|x|^2/\beta}\\&:=M_\beta(|x|,n).\end{aligned} \tag{10}$$

To simulate approximately the β-Ginibre process on $\mathbf{B}_R$, the standard procedure consists in simulating the truncated β-Ginibre process for n large enough so that $R\leq\sqrt{n\beta}$ and so that the approximation is satisfying. To our knowledge, there is no clear recommendation for the choice of n. Choosing $n=\lceil R^2/\beta\rceil$ would lead to a poor approximation close to the border of the disc. We instead suggest to choose n so that the upper bound in (10) is uniformly less than a prescribed error ε on $\mathbf{B}_R$. Specifically, we choose $n=\lceil n_\varepsilon\rceil$ where n_ε is such that $M_\beta(R,n_\varepsilon)=\varepsilon$. In the simulations presented in Sect. 4.3, we fixed $\varepsilon=10^{-10}$. The full procedure is summarised in Algorithm 4.

Algorithm 4 β-Ginibre process on $\mathbf{B}_R$ by the eigenvalues method

```
set $n = \lceil n_\varepsilon \rceil$ where $n_\varepsilon$ solves $M_\beta(R, n_\varepsilon) = \varepsilon$; see (10)
sample the matrix $M = [(A_{k,l}+iB_{k,l})\sqrt{\beta/2}]_{1\leq k,l\leq n}$ where $i=\sqrt{-1}$
and $A_{k,l}$ and $B_{k,l}$ are all independent $\mathcal{N}(0,1)$ random variables
compute the eigenvalues $\{Z_1,\ldots,Z_n\}$ of $M$
set $X = \{\emptyset\}$
for $i = 1$ to $n$ do
    generate $U \sim \mathcal{U}([0,1])$
    if $U < \rho\beta\pi$ and $|Z_i| < R$ then
        $X \leftarrow X \cup \{Z_i\}$
    end if
end for
return $X$
```

4.2 Spectral algorithm

Starting from $\sum_{k\geq 0}(x\bar{y}/\beta)^k/k!=\exp(x\bar{y}/\beta)$, it is not difficult to verify that the spectral representation of the β-Ginibre kernel (8) on $\mathbb{C}$ reads

$$K(x,y)=\sum_{k\geq 0}\rho\beta\pi\,\Phi_k(x)\overline{\Phi}_k(y),\quad x,y\in\mathbb{C}, \tag{11}$$

where

$$\Phi_k(x)=\frac{x^k}{\sqrt{\pi k!\beta^{k+1}}}e^{-|x|^2/(2\beta)},\quad x\in\mathbb{C}.$$

As noticed in Decreusefond et al. (2015), the functions Φ_k satisfy the unusual property to remain orthogonal on the ball $\mathbf{B}_R$ for any $R>0$. So the spectral representation of K on $\mathbf{B}_R$ is simply the restriction of (11) to $\mathbf{B}_R$:

$$K(x,y)=\sum_{k\geq 0}\rho\beta\pi\|\Phi_k\|_R^2\frac{\Phi_k(x)}{\|\Phi_k\|_R}\frac{\overline{\Phi}_k(y)}{\|\Phi_k\|_R},\quad x,y\in\mathbf{B}_R,$$

where $\|\Phi_k\|_R^2=\int_{\mathbf{B}_R}|\Phi_k(x)|^2dx$ is introduced to normalise the eigenfunctions on $\mathbf{B}_R$. Some algebra yields $\|\Phi_k\|_R^2=\gamma(k+1,R^2/\beta)/k!$ where $\gamma(a,z)=\int_0^z t^{a-1}e^{-t}dt$ is the incomplete gamma function. Finally the spectral representation of K on $\mathbf{B}_R$ is

$$K(x,y)=\sum_{k\geq 0}\lambda_k\tilde{\Phi}_k(x)\overline{\tilde{\Phi}}_k(y),\quad x,y\in\mathbf{B}_R, \tag{12}$$

where

$$\lambda_k=\rho\beta\pi\gamma(k+1,R^2/\beta)/k! \tag{13}$$

and

$$\tilde{\Phi}_k(x)=\frac{x^k}{\sqrt{\pi\beta^{k+1}\gamma(k+1,R^2/\beta)}}e^{-|x|^2/(2\beta)},\quad x\in\mathbf{B}_R. \tag{14}$$

To apply the spectral algorithm of Hough et al. (2006), we need to first simulate a sequence of Bernoulli random variables B_k, with respective parameters λ_k, $k\geq 0$. We know that $M=\max_{k\geq 0}\{B_k\neq 0\}$ is finite because $\sum\lambda_k<\infty$, but the simulation of M is not straightforward; see (Lavancier et al. 2014, Appendix D). A simple solution consists in setting a maximal number $k_{\max}$ of Bernoulli variables that we generate. This amounts to truncate the representation (12) to $k_{\max}-1$, which gives exactly the same kernel as in (9) but restricted to $\mathbf{B}_R$. A clever choice of $k_{\max}$ can then be carried out as for the choice of n in (9), that is $k_{\max}=\lceil k_\varepsilon\rceil$ where k_ε is the solution of $M_\beta(R,k_\varepsilon)=\varepsilon$. Here ε is a prescribed error ($\varepsilon = 10^{-10}$ in Sect. 4.3) that guarantees through (10) that the choice of $k_{\max}$ has a little impact on the expected number of points. The details of this spectral approach are given Algorithm 5.

Algorithm 5 β-Ginibre process on $\mathbf{B}_R$ by the spectral method

```
set $k_{\max} = \lceil k_\varepsilon \rceil$ where $k_\varepsilon$ solves $M_\beta(R, k_\varepsilon) = \varepsilon$; see (10)
sample independently $B_0 \sim \mathcal{B}(\lambda_0), \dots, B_{k_{\max}-1} \sim \mathcal{B}(\lambda_{k_{\max}-1})$
where the $\lambda_k$'s are given by (13)
if $B_k = 0$ for all $k = 0, \dots, k_{\max} - 1$ then
    return the empty point configuration $\{\emptyset\}$
else
    Given $B_0, \dots, B_{k_{\max}-1}$, generate the projection DPP with kernel

    $\sum_{k=0}^{k_{\max}-1} B_k \tilde{\Phi}_k(x) \overline{\tilde{\Phi}}_k(y), \quad x, y \in \mathbf{B}_R,$

    using Algorithm 2, where the $\tilde{\Phi}_k$'s are given by (14)
end if
```

4.3 Simulation study

In order to compare Algorithms 4 and 5, we have performed several simulations of the β-Ginibre process, for different values of the parameters ρ and β, on the ball $\mathbf{B}_R$ of area 1 where $R = 1/\sqrt{\pi}$, so that the expected number of points coincide with the intensity ρ. An example using each algorithm is shown in Fig. 3, where the gray points on the left correspond to the deleted points in the process of Algorithm 4, while the black points represent the final simulated point pattern.

Table 2 displays the average computation time (over 500 replications) of each algorithm. The picture is quite clear: the spectral algorithm 5 is more efficient to generate a moderate number of points, while Algorithm 4 becomes preferable for a high number of points. On the other hand, the effect of β is mild for the spectral algorithm, whereas it has a strong impact on the efficiency of Algorithm 4 (the smaller β is, the less efficient it is), which is consistent with the fact that when β is small, many points are generated to be afterwards deleted by the thinning step of Algorithm 4, slowing down the process.

5 Exact spectral expansion for other kernels

This section aims at paving the way to (nearly) perfect simulation of the Gaussian-type DPP and the Bessel-type DPP, two widely used DPPs in spatial statistics. In both cases we provide the explicit spectral representation of the kernel on the unit ball, which opens the way to perfect simulation by the spectral algorithm. The numerical evaluation of the detailed spectral representations may pose some separate numerical challenges which are left for future research.

5.1 Gaussian-type kernels

5.1.1 Inhomogeneous kernel

We call the DPP with the following kernel an inhomogeneous Gaussian-type DPP:

$$K_\sigma(x, y) = \rho\, e^{-\|y-x\|^2/\alpha^2} \sqrt{p_\sigma(x) p_\sigma(y)}, \quad x, y \in \mathbb{R}^d, \tag{15}$$

where $\rho > 0$ is the intensity parameter, $\alpha > 0$ is the range parameter, and p_σ denotes the density of a standard normal distribution in $\mathbb{R}^d$ with standard deviation $\sigma > 0$, i.e.

$$p_\sigma(x) = \frac{1}{(\sigma\sqrt{2\pi})^d} e^{-\|x\|^2/(2\sigma^2)}, \quad x \in \mathbb{R}^d.$$

As verified in "Appendix C", this DPP exists if and only if

$$2\rho^{1/d} \le 1 + \sqrt{1 + 8\sigma^2/\alpha^2}, \tag{16}$$

or equivalently $2\sigma^2/\alpha^2 \ge \rho^{2/d} - \rho^{1/d}$.

Since the intensity is $K_\sigma(x, x) = \rho\, p_\sigma(x)$, $x \in \mathbb{R}^d$, this DPP is an inhomogeneous finite point process on $\mathbb{R}^d$ with expected cardinality ρ. Accordingly, realised point patterns are concentrated near the origin. Specifically, let $\chi^2_d(p)$ denote the quantile of the chi-squared distribution with d degrees of freedom, then for any $0 \le \varepsilon \le 1$, the expected number of points in the centered ball with radius $\sigma\sqrt{\chi^2_d(1-\varepsilon)}$ is $(1-\varepsilon)\rho$. For instance in dimension $d = 2$, about 99% of the points are in the centered ball with radius 3σ.

A special property of the inhomogeneous Gaussian-type DPP is that the spectral representation of its kernel is perfectly known on $\mathbb{R}^d$, as explained below [see for instance (Williams and Rasmussen 2006, Section 4.3.1)]. This decomposition specifically reads

$$K_\sigma(x, y) = \sum_{j \in \mathbb{N}^d} \lambda_j \Phi_j(x) \Phi_j(y), \quad x, y \in \mathbb{R}^d, \tag{17}$$

where, setting $a = \sigma^{-2}/4$, $b = \alpha^{-2}$, $c = \sqrt{a^2 + 2ab}$, $A = a + b + c$ and $B = b/A$, for $j = (j_1, \dots, j_d) \in \mathbb{N}^d$ and $x = (x_1, \dots, x_d) \in \mathbb{R}^d$,

$$\lambda_j = \rho \left(\frac{2a}{A}\right)^{d/2} B^{\sum_{k=1}^d j_k} \tag{18}$$

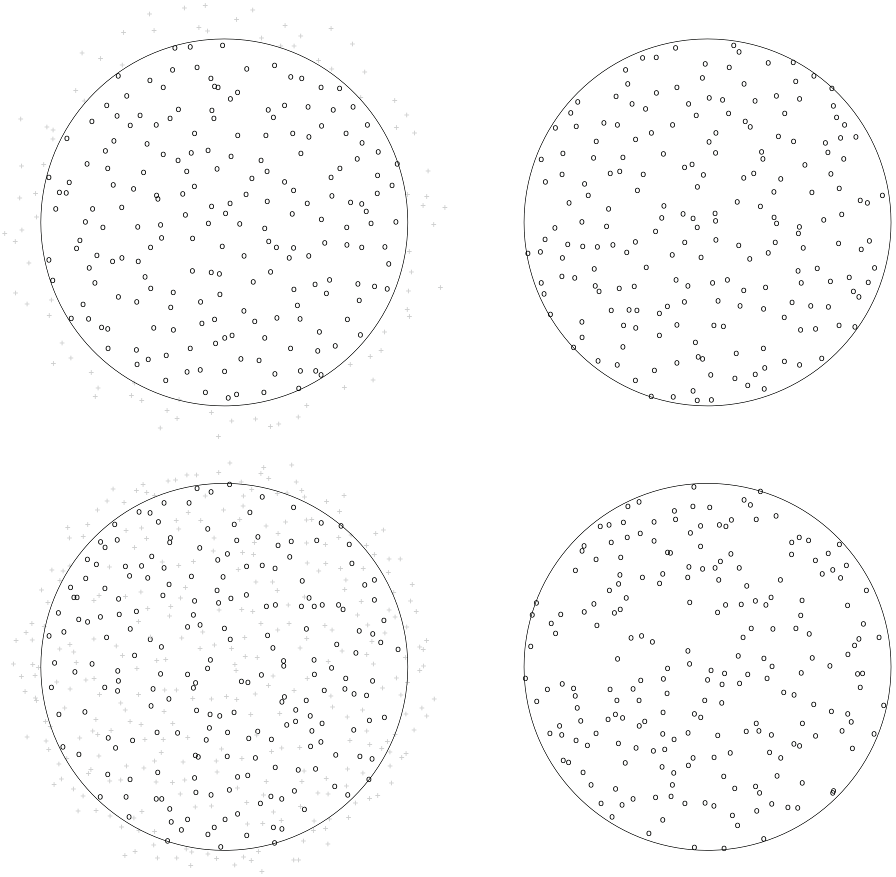

Fig. 3 Realisations of two β-Ginibre processes generated with Algorithms 4 (left) and 5 (right). In all cases $\rho = 200$. In the top row $\beta = \beta_{\max}$ and in the bottom row $\beta = \beta_{\max}/2$. For Algorithm 4 on the left the gray points represent the points deleted by the algorithm

Table 2 Average computation time (in seconds) for a single realization of a β-Ginibre process on $\mathbf{B}_R$, $R = 1/\sqrt{\pi}$, with the given parameters, where $\beta_{\max} = 1/(\rho\pi)$, using Algorithms 4 (left value) and 5 (right value)

Algorithm 4/5	$\rho = 100$	$\rho = 200$	$\rho = 400$	$\rho = 800$
$\beta = \beta_{\max}/3$	0.58/**0.09**	2.99/**0.36**	19.21/**6.63**	130.14/**96.44**
$\beta = \beta_{\max}/2$	0.17/**0.07**	0.59/**0.27**	**3.11**/4.58	**18.43**/68.83
$\beta = \beta_{\max}$	0.09/**0.07**	0.28/**0.23**	**1.32**/3.43	**6.45**/50.09

The fastest method is boldfaced

and $\Phi_j(x) = \prod_{k=1}^d \varphi_{j_k}(x_k)$ with

$$\varphi_k(x) = \left(\frac{\sqrt{2c}}{2^k k!\sqrt{\pi}}\right)^{1/2} e^{-cx^2} H_k(\sqrt{2c}\,x),$$

H_k being the (physicists') Hermite polynomial of order k. The spectral decomposition (17) is a tensorial product and is deduced from the decomposition when $d = 1$. The latter is easily verified since when $d = 1$, the identity $\int_{\mathbb{R}} K_\sigma(x, y)\varphi_k(x)dx = \rho\sqrt{2a/A}\, B^k \varphi_k(y)$ is a consequence of equation 7.374-8 in Gradshteyn and Ryzhik (2014), that is

$$\int_{\mathbb{R}} e^{-(x-y)^2} H_k(tx)dx = \sqrt{\pi}(1-t^2)^{k/2} H_k(ty/\sqrt{1-t^2}),$$
$$|t| < 1,\ y \in \mathbb{R}.$$

The orthonormality of the eigenfunctions is in turn a consequence of the relation $\int_{\mathbb{R}} H_{k_1}(x)H_{k_2}(x)e^{-x^2}dx = \sqrt{\pi}2^k k!$ if $k_1 = k_2$ and 0 otherwise.

Based on (17), the spectral algorithm is in theory feasible to simulate the inhomogeneous Gaussian-type DPP on $\mathbb{R}^d$. Two approximations are however needed in practice. The first one is the truncation of (17) to $|j| \leq \ell$, for some ℓ such that the expected number of generated points $\sum_{|j|\leq\ell} \lambda_j$ is sufficiently close to ρ. In fact, we know in theory the exact distribution of the index ℓ of the maximal non-null eigenvalue, and perfect simulation of ℓ could in principle be considered, see "Appendix D" in Lavancier et al. (2014). But this distribution is in practice impossible to generate and manual truncation is mandatory. This truncation allows us to generate the $(2\ell + 1)^d$ Bernoulli variables B_j that are needed in the first step of the spectral algorithm. Then, given the B_j's, the simulation boils down to generating the projection DPP with kernel $K_{proj}(x, y) = \sum_{|j|\leq \ell} B_j \Phi_j(x)\Phi_j(y)$ using Algorithm 2. But in this algorithm, the simulation with respect to p_i given by (5) cannot easily be performed by Strategy 1 in Sect. 3.1 because the simulation with respect to the unnormalised density $K_{proj}(x, x)$ is not straightforward. Instead we need to apply Strategy 2, where the proposal in the rejection sampling is the uniform distribution. To do so, we need a second approximation which is the choice of a compact set S that contains with high probability the generated point pattern. The natural choice for S is the centered ball with radius $\sigma\sqrt{\chi^2_d(1-\varepsilon)}$ for a prescribed error ε.

5.1.2 Homogeneous Gaussian-type kernel

The homogeneous Gaussian-type kernel is defined for any $x, y \in \mathbb{R}^d$ by

$$K(x, y) = \rho\, e^{-\|y-x\|^2/\alpha^2}, \quad x, y \in \mathbb{R}^d. \tag{19}$$

The DPP with this kernel exists on $\mathbb{R}^d$ if and only if $\rho\alpha^d\pi^{d/2} \leq 1$ Lavancier et al. (2015), and it is stationary with intensity ρ.

Note that the homogeneous Gaussian-type DPP is the limit in distribution, when $\sigma \to \infty$, of the inhomogeneous Gaussian-type DPP with intensity parameter $\rho\sigma^d(2\pi)^{d/2}$. This comes from the fact that the convergence of the associated kernel holds uniformly on all compact sets, which implies that the Laplace transforms of the two DPPs asymptotically coincide; see (Shirai and Takahashi 2003, Proposition 3.10). Despite this property, the spectral representation of (19) is not known, neither on $\mathbb{R}^d$ or on any subset. For this reason, the standard approach to simulate the homogeneous Gaussian-type DPP is to resort to the spectral approximation by a Fourier series.

We explain below an alternative simulation procedure, that is arguably much more accurate than the Fourier series approximation, although more time demanding. The idea is to first generate an inhomogeneous Gaussian-type DPP using the exact spectral representation (17), then apply an independent thinning procedure to get the final point pattern.

Assume that we wish to simulate the homogeneous Gaussian-type DPP with parameter ρ and α on the unit ball $\mathbf{B}$ (the procedure extends straightforwardly to any compact set). Let

$$\tilde{\rho}_\sigma = \rho \sup_{x\in\mathbf{B}} p_\sigma^{-1}(x) = \rho\sigma^d(2\pi)^{d/2}e^{1/(2\sigma^2)}.$$

The main steps of the simulation algorithm are the following:

(i) Choose $\sigma_0 = \operatorname*{argmin}_{\sigma>0} \tilde{\rho}_\sigma$ subject to $\sigma^2 \geq \alpha^2(\rho^{2/d} - \rho^{1/d})/2$.

(ii) Generate the inhomogeneous Gaussian-type DPP with intensity parameter $\tilde{\rho}_{\sigma_0}$, range parameter α and standard deviation σ_0, on $\mathbb{R}^d$, using the spectral algorithm (see Sect. 5.1.1).

(iii) Apply an independent thinning procedure with retention probability

$$q(x) = \frac{p_{\sigma_0}^{-1}(x)}{(\sigma_0\sqrt{2\pi})^d e^{1/(2\sigma_0^2)}} 1_{\mathbf{B}}(x), \quad x \in \mathbb{R}^d.$$

The first step aims at choosing the value of σ that minimises the intensity of the inhomogeneous Gaussian-type DPP, thus optimising the computational cost, while ensuring the existence of the process through (16). It is easily verified that the final point pattern is a DPP on $\mathbb{R}^d$ with kernel

$$\begin{aligned}&\tilde{\rho}_{\sigma_0}\, e^{-\|y-x\|^2/\alpha^2}\sqrt{p_{\sigma_0}(x)p_{\sigma_0}(y)} \times \sqrt{q(x)q(y)}\\ &= \rho\, e^{-\|y-x\|^2/\alpha^2} 1_{\mathbf{B}}(x)1_{\mathbf{B}}(y),\end{aligned}$$

that is a homogeneous Gaussian-type DPP on $\mathbf{B}$. This procedure provides a nearly perfect simulation algorithm of this DPP, up to the two mild approximations discussed in Sect. 5.1.1 used for the second step. It can however be very time consuming, and we think that it is to be considered instead of the Fourier series approximation only if distributional accuracy is of main concern.

5.2 Bessel-type kernel

The Bessel-type kernel is defined for any $x, y \in \mathbb{R}^d$ by

$$K(x, y) = \rho\,\Gamma(1 + d/2)\,\frac{J_{d/2}\,(2\|x-y\|/\alpha)}{(\|x-y\|/\alpha)^{d/2}}, \tag{20}$$

where $\rho > 0$ is the intensity, $\alpha > 0$ is the range parameter and $J_{d/2}$ is the Bessel function of the first kind. This kernel defines a DPP in $\mathbb{R}^d$ whenever $\rho\,\alpha^d\pi^{d/2}\Gamma(1 + d/2) \leq 1$; see Biscio and Lavancier (2016). Given $\rho > 0$, the maximal possible value of α corresponds to the most repulsive stationary DPP in $\mathbb{R}^d$ with intensity ρ, while $\alpha \to 0$ corresponds to the Poisson point process with intensity ρ.

As explained below, the spectral representation of this kernel is explicit on the unit ball $\mathbf{B}$ and involves the so-called generalized prolate spheroidal functions introduced and studied in (Slepian 1964). This representation opens the way to perfect simulation, by the spectral algorithm, of the Bessel-type DPP on any ball, and therefore on any bounded set (by first simulating on an outer ball and then taking the restriction to the desired set).

The generalized prolate spheroidal functions are the orthogonal eigenfunctions of the truncated Fourier transform, which means that given an arbitrary $c \in \mathbb{R}$, each such function ψ satisfies for a certain $\gamma \in \mathbb{C}$

$$\int_{\|y\| \le 1} \psi(y) e^{cix \cdot y} dy = \gamma \psi(x), \quad x \in \mathbb{R}^d. \tag{21}$$

The eigenvalues γ and the eigenfunctions ψ depend of course on c. These functions are proved to be real-valued, so iterating the latter equality and its transpose we obtain that

$$\begin{aligned} |\gamma|^2 \psi(x) &= \int_{\|z\| \le 1} \bar{\gamma} \psi(z) e^{cix \cdot z} dz \\ &= \int_{\|z\| \le 1} \left(\int_{\|y\| \le 1} \psi(y) e^{-ciz \cdot y} dy \right) e^{cix \cdot z} dz. \end{aligned}$$

If we let K_c denote the Fourier transform of the unit ball, i.e. (see Grafakos 2008)

$$\begin{aligned} &K_c(x-y) = \int_{\|z\| \le 1} e^{ciz \cdot (x-y)} dz \\ &= (2\pi)^{d/2} \frac{J_{\frac{d}{2}}(c\|x-y\|)}{(c\|x-y\|)^{d/2}}, \end{aligned}$$

this means that

$$\int_{\|y\| \le 1} K_c(x-y) \psi(y) dy = |\gamma|^2 \psi(x).$$

By choosing $c = 2/\alpha$, we deduce that ψ is an eigenfunction of the Bessel-type kernel (20) associated with the eigenvalue $\rho \Gamma(1 + d/2) |\gamma|^2 / \pi^{d/2}$.

The generalized prolate spheroidal functions ψ and the eigenvalues γ have rather complicated expressions. In the one dimensional case ($d = 1$), these expressions are obtained in Slepian and Pollak (1961) and detailed in Moore and Cada (2004), where the eigenvalues form a sequence γ_n associated to the eigenfunctions ψ_n. To make the connection with our setting, note that $\lambda_n(c)$ in Moore and Cada (2004) corresponds to $c|\gamma_n|^2/(2\pi)$, and the associated eigenfunction $\psi_n(c, x)$ corresponds to $\psi_n(x)$. Note finally that the norm of $\psi_n(c, x)$ on the unit ball is $\sqrt{\lambda_n(c)}$ (see (9) in Moore and Cada (2004)) so that in dimension $d = 1$, the spectral expansion of the Bessel-type kernel on the unit ball is

$$K(x, y) = \rho\alpha \tfrac{\pi}{2} \textstyle\sum_{n \ge 0} \lambda_n^2(2/\alpha) \tfrac{\psi_n(2/\alpha, x)}{\sqrt{\lambda_n(2/\alpha)}} \tfrac{\psi_n(2/\alpha, y)}{\sqrt{\lambda_n(2/\alpha)}}, \\ \forall x, y \in [-1, 1],$$

using the notation of Moore and Cada (2004). Several algorithms are available to evaluate λ_n and ψ_n; see for instance Moore and Cada (2004), Osipov et al. (2013), Bremer (2022) and the references therein.

In dimension $d \ge 2$, following (Slepian 1964), the sequence of eigenvalues can be written $\{\gamma_{N,n}, N \ge 0, n \ge 0\}$, where the eigenvalues $\gamma_{0,n}$ are simple while the eigenvalues $\gamma_{N,n}$ for $N \ge 1$ have multiplicity $h(N, d) = (2N + d - 2)(N + d - 3)!/((d - 2)!N!)$. Accordingly, their associated eigenfunctions form a sequence $\psi_{N,n,l}$ where $l = 1, \dots, h(N, d)$. All formulas are given in Slepian (1964) and we do not reproduce them in the general case. We only provide some details in "Appendix D" for the case $d = 2$; see also Shkolnisky (2007) for this particular case. This results in the following spectral representation, for any $x, y \in \mathbf{B}$,

$$\begin{aligned} &K(x, y) = 2\pi\rho\alpha \sum_{n \ge 0} \lambda_{0,n}^2(2/\alpha) \psi_{0,n}(x) \psi_{0,n}(y) \\ &+ 2\pi\rho\alpha \sum_{N \ge 1} \sum_{n \ge 0} \lambda_{N,n}^2(2/\alpha) (\psi_{N,n,1}(x) \psi_{N,n,1}(y) \\ &+ \psi_{N,n,2}(x) \psi_{N,n,2}(y)), \end{aligned} \tag{22}$$

where the specific form of $\lambda_{0,n}, \psi_{0,n}, \lambda_{N,n}, \psi_{N,n,1}$ and $\psi_{N,n,2}$ are given in "Appendix D". Contrary to $d = 1$, numerical algorithms to compute these eigenvalues and eigenfunctions in dimension $d = 2$ are not mature yet, but they are the subject of an active current research and software is becoming available online (Shkolnisky 2007; Greengard and Serkh 2018; Lederman 2017). This will hopefully make the computation of the above spectral representation straightforward in the near future.

Author Contributions FL and ER have contributed equally to this work.

Funding Open access funding provided by Aalborg University Library

Declarations

Competing interests The authors declare no competing interests.

A Conditional distributions

A.1 Palm distribution and Algorithms 1 and 2

Let X be a DPP on S with kernel K and let $y_1, \dots, y_m \in S$. Remember that the conditional distribution of $X \setminus \{y_1, \dots, y_m\}$ given that $\{y_1, \dots, y_m\} \in X$ is formally called the reduced Palm distribution of X given $y_1, \dots, y_m$. The intensity function of order n of the reduced Palm distribution is

$$\rho^{(n)}_{y_1,\dots,y_m}(x_1, \dots, x_n) = \frac{\rho^{(n+m)}(x_1, \dots, x_n, y_1, \dots, y_m)}{\rho^{(m)}(y_1, \dots, y_m)} \tag{23}$$

if $\rho^{(m)}(y_1, \dots, y_m) > 0$, and 0 otherwise; see Coeurjolly and Lavancier (2019). Since X is a DPP, we may simplify the expression of $\rho^{(n+m)}$ thanks to (1) and the standard formula

$$\det \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \det(A)\det(D - CA^{-1}B), \tag{24}$$

where $\det(A) = \rho^{(m)}(y_1, \dots, y_m)$, whereby we deduce that the reduced Palm distribution of X given $y_1, \dots, y_m$ is still a DPP with kernel

$$K_{y_1,\dots,y_m}(x, y) = K(x, y) - \mathbf{k}^*_m(x)\mathbf{K}_m^{-1}\mathbf{k}_m(y), \quad x, y \in S, \tag{25}$$

where $\mathbf{k}^*_m(x) = (K(x, y_1), \dots, K(x, y_m))$ and $\mathbf{K}_m = (K(y_j, y_l))_{1 \le j,l \le m}$.

This result is not enough to simulate a general DPP, because it says nothing about its cardinality. But if X is a projection DPP on S, then we know its cardinality $n = \int_S K(x, x)dx$ and $\rho^{(n)}$ becomes the unnormalised density of the n-tuple set of points of X, with respect to the Lebesgue measure on S^n. In this case, for any $i = n-1, \dots, 1$, by (23) where (m, n) is $(n-i, i)$, $\rho^{(i)}_{y_{i+1},\dots,y_n}$ is the unnormalised density of $X \setminus \{y_{i+1}, \dots, y_n\}$ given that $\{y_{i+1}, \dots, y_n\} \in X$. We can thus apply a sequential conditional simulation: generate the first point y_n of X, the marginal density of which is $K(x, x)/n$, then the second point y_{n-1} given the first one, according to $K_{y_n}(x, x)/(n-1)$, then the third one given the first two points, according to $K_{y_n,y_{n-1}}(x, x)/(n-2)$, and so on. This procedure leads to Algorithm 1.

To accelerate this algorithm, the expression of $K_{y_{i+1},\dots,y_n}(x, x)$ or equivalently of $ip_i(x)$ in (3) can be rewritten to get (5) in Algorithm 2, provided we know the spectral representation of K given by (4). Then it is possible to calculate the matrix of the projection onto the orthogonal complement of $span\{\boldsymbol{v}(X_n), \dots, \boldsymbol{v}(X_{i+1})\}$ which is $P_i = I_n - \mathbf{V}(\mathbf{V}^*\mathbf{V})^{-1}\mathbf{V}^* = I_n - \mathbf{V}\overline{\mathbf{K}}_i^{-1}\mathbf{V}^*$, where $\mathbf{V}$ is the $(n, n-i)$ matrix with entries $\Phi_i(X_j)$, $i = 1, \dots, n$, $j = n, \dots, i+1$. Then, starting from (3), it is not difficult to verify that $ip_i(x) = \| P_i \boldsymbol{v}(x) \|^2$. This means that ip_i may be obtained by successive orthogonalisation, which is exploited in Algorithm 2 [see Lavancier et al. (2015) for more details].

A.2 Conditional distribution given an outside set

Let A and A^c be a partition of S, i.e. $S = A \cup A^c$ and $A \cap A^c = \emptyset$. If X is a projection DPP on S with cardinality n, then the density of $X \cap A$ given that $X \cap A^c = \{y_1, \dots, y_m\}$, $m < n$, is

$$f_{A|A^c}(X \cap A = \{x_1, \dots, x_{n-m}\} | X \cap A^c = \{y_1, \dots, y_m\}) \\ = \frac{f_S(X = \{x_1, \dots, x_{n-m}, y_1, \dots, y_m\})}{f_{A^c}(X \cap A^c = \{y_1, \dots, y_m\})}$$

if $\{x_1, \dots, x_{n-m}\} \subset A$ and 0 otherwise. We have excluded the trivial case $m = n$ since $X \cap A$ is simply the empty set in this case. Since X is a projection DPP on S, the numerator above is (up to a multiplicative constant) the determinant of the 2×2 block matrix

$$\begin{pmatrix} [K(y_i, y_j)]_{1 \le i,j \le m} & [K(y_i, x_j)]_{1 \le i \le m, 1 \le j \le n-m} \\ [K(x_i, y_j)]_{1 \le i \le n-m, 1 \le j \le m} & [K(x_i, x_j)]_{1 \le i,j \le n-m} \end{pmatrix}.$$

By (24) and using the same notation as in (25), we deduce that

$$f_{A|A^c}(X \cap A = \{x_1, \dots, x_{n-m}\} | X \cap A^c = \{y_1, \dots, y_m\}) \propto \\ \det[K_{y_1,\dots,y_m}(x_i, x_j)]_{1 \le i,j \le n-m},$$

where the constant of proportionality does not depend on $\{x_1, \dots, x_{n-m}\}$. This proves that given $X \cap A^c = \{y_1, \dots, y_m\}$, $X \cap A$ is a projection DPP with cardinality $n - m$ and kernel $K_{y_1,\dots,y_m}(x, y)\mathbf{1}_A(x)\mathbf{1}_A(y)$.

A.3 How to choose a projection DPP for conditional simulation?

We start by a simple setting. Assume that we observe m points of a point process Y that is an independent thinning of a homogeneous point process X on $S = [0, 1]^d$ with constant thinning probability $\tau \in [0, 1]$. Assume moreover that we know the value of τ. Then a way to model X as a projection DPP on S that fits with the observed thinned point process Y is as follows.

Choose for X a parametric stationary model of DPP with kernel $K(x, y) = \rho K_\alpha(x - y)$ with $K_\alpha(0) = 1$, for which we know the Fourier transform φ_α of K_α, for instance the Gaussian-type DPP model (Lavancier et al. 2015). Here ρ is the intensity of X and α stands for the remaining parameters of the model, typically α is a univariate range parameter. Under this model, Y is a DPP with kernel $K_{thin}(x, y) =$

$\tau K(x, y)$ and so its intensity is $\tau\rho$ while its pair correlation function is $1 - K_{\alpha}^{2}(x - y)$. Consequently we can estimate from Y the intensity of X by $\hat{\rho} = m/\tau$ and the remaining parameter α by a contrast estimation method based on the pcf or the Ripley's K function of Y to get $\hat{\alpha}$ (see Lavancier et al. 2015; Biscio and Lavancier 2017). Using the Fourier approximation of Lavancier et al. (2015), we thus have

$$K(x, y) \approx \hat{\rho} \sum_{k \geq 0} \varphi_{\hat{\alpha}}(k) e^{ik\cdot(x-y)}, \quad x, y \in S.$$

Following the simulation procedure to generate a DPP with such kernel, a projection DPP kernel that can be used to approximate the distribution of X is a realisation of

$$K_{proj}(x, y) = \sum_{k \geq 0} B_k e^{ik\cdot(x-y)},$$

conditional on $\hat{\rho}$, $\hat{\alpha}$ and $\sum_{k\geq 0} B_k \geq m$, where the B_k's are independent Bernoulli variables with respective rate $\widehat{\rho\varphi}_{\hat{\alpha}}(k)$.

The above setting can be generalized to the case where the thinning probability is a known function $q(x)$, $x \in S$. For instance we may have $q(x) = \tau a(x)/(\sup_{x\in S} a(x))$ where $\tau \in [0, 1]$ is known and $a(x)$ is an observed auxiliary variable. Note that inpainting conditional simulation in A given a point pattern in A^c corresponds to the particular case where $q(x) = \mathbf{1}_{A^c}(x)$. Then Y becomes a DPP with kernel $K_{thin}(x, y) = \rho\sqrt{q(x)}K_{\alpha}(x - y)\sqrt{q(y)}$ and we can deduce $\hat{\rho} = m/(\int_S q(x)dx)$ and $\hat{\alpha}$ by minimum contrast estimation. The approximation of X by a projection DPP can then be carried out as above.

A further generalization is possible in the case where X is inhomogeneous with intensity $\rho_{\theta}(x)$, where θ is an unknown parameter. As before, we assume to observe m points of Y, which is a thinning of X with thinning probability $q(x) = \tau a(x)/(\sup_{x\in S} a(x))$ where $\tau \in [0, 1]$ is known and $a(x)$ is an observed auxiliary variable. We can then start with a parametric DPP model for X with kernel $K(x, y) = \sqrt{\rho_{\theta}(x)}K_{\alpha}(x - y)\sqrt{\rho_{\theta}(y)}$ from which we deduce that Y is an inhomogeneous DPP with kernel $\sqrt{q(x)\rho_{\theta}(x)}K_{\alpha}(x - y)\sqrt{\rho_{\theta}(y)q(y)}$. Based on the observation of Y, we may deduce the estimations $\hat{\theta}$ and $\hat{\alpha}$ by a two-step estimation method as in Lavancier et al. (2021). Remember that X can be seen as a thinning of a homogeneous DPP X^+ with kernel $K^+(x, y) = \rho^+ K_{\alpha}(x - y)$ and thinning probability $\rho_{\theta}(x)/\rho^+$, where ρ^+ is an upper bound of $\rho_{\theta}(x)$ on S. Following the same scheme as before, we may approximate X^+ by a projection DPP with random kernel

$$K_{proj}(x, y) = \sum_{k \geq 0} B_k e^{ik\cdot(x-y)},$$

conditional on $\hat{\rho}^+$, $\hat{\alpha}$ and $\sum_{k\geq 0} B_k \geq m$, where the B_k's are independent Bernoulli variables with respective rate $\hat{\rho}^+\varphi_{\hat{\alpha}}(k)$. A conditional simulation of X^+ given Y is then possible. To finally get a realisation of X given Y, it suffices to thin the realisation of $X^+ \setminus Y$ with thinning probability $\rho_{\hat{\theta}}(x)/\hat{\rho}^+$.

B A bound on the conditional probability density function for the Fourier basis

From Lavancier et al. (2015), we know that the conditional density p_i, given the $n - i$ first points $X_{i+1}, \ldots, X_n$ satisfies

$$p_i(x) \leq \frac{1}{i} \min_{i+1\leq k\leq n} \left(K(x, x) - \frac{|K(x, X_k)|^2}{K(X_k, X_k)} \right). \tag{26}$$

When using the Fourier basis on the unit square of $\mathbb{R}^d$, the projection kernel reads

$$K(x, y) = \sum_{j\in J} \mathrm{e}^{2\mathrm{i}\pi j\cdot(x-y)}$$

where $J \subset \mathbb{Z}^d$ with cardinality $|J| = n$. The inequality (26) becomes in this case

$$p_i(x) \leq \frac{1}{i} \left(n - \max_{i+1\leq k\leq n} \frac{|K(x, X_k)|^2}{n} \right).$$

But for any x, y we have

$$\begin{aligned}
|K(x, y)|^2 &= \left(\sum_{j\in J} \cos(2\pi j \cdot (x - y)) \right)^2 \\
&\quad + \left(\sum_{j\in J} \sin(2\pi j \cdot (x - y)) \right)^2 \\
&= \sum_{j,k\in J} \cos(2\pi j \cdot (x - y)) \cos(2\pi k \cdot (x - y)) \\
&\quad + \sin(2\pi j \cdot (x - y)) \sin(2\pi k \cdot (x - y)) \\
&= \sum_{j,k\in J} \cos(2\pi (j - k) \cdot (x - y)) \\
&= \sum_{p\geq 0} \frac{(-1)^p (2\pi)^{2p}}{(2p)!} \sum_{j,k\in J} [(j - k) \cdot (x - y)]^{2p} \\
&\geq n^2 - 2\pi^2 \sum_{j,k\in J} [(j - k) \cdot (x - y)]^2,
\end{aligned}$$

where the last inequality uses the fact that $|K(x, y)|^2$ is expanded into an alternating series. Since $\sum_{j,k\in J}[(j - k) \cdot (x - y)]^2 = 2n \sum_{j\in J}[j \cdot (x - y)]^2 - 2 \left(\sum_{j\in J} j \cdot (x - y) \right)^2$, we obtain

$$
\begin{aligned}
ip_i(x) &\le n - \max_{i+1\le k\le n}\left(n - 4\pi^2\sum_{j\in J}[j\cdot(x-X_k)]^2\right.\\
&\qquad \left.+\frac{4\pi^2}{n}\left(\sum_{j\in J} j\cdot(x-X_k)\right)^2\right)_+ .\\
&= \min_{i+1\le k\le n}\left(n, 4\pi^2\sum_{j\in J}[j\cdot(x-X_k)]^2\right.\\
&\qquad \left.-\frac{4\pi^2}{n}\left(\sum_{j\in J} j\cdot(x-X_k)\right)^2\right),
\end{aligned}
$$

which is (6).

B.1 A new proposal in dimension $d = 1$

When $d = 1$, we have $S = [0, 1]$ and the inequality (6) simplifies to give

$$
ip_i(x) \le \min_{i+1\le k\le n}\left(n, a^2(x-X_k)^2\right)
$$

where $a^2 = 4\pi^2(\sum_{j\in J} j^2 - (\sum_{j\in J} j)^2/n)$.

Note that $a^2(x - X_k)^2 \le n$ iff $|x - X_k| \le \sqrt{n}/a$. The minimum above thus simplifies greatly if all intervals $|x - X_k| \le \sqrt{n}/a$, for $k = i + 1, \ldots, n$, are disjoint. This motivates us to consider a thinning of $\{X_{i+1}, \ldots, X_n\}$ such that all points are at a distance greater than $2\sqrt{n}/a$ from each other and also at a distance larger than $\sqrt{n}/a$ from the borders 0 and 1. This can be achieved by first removing the points too close to the borders, followed by a hardcore Matern II thinning procedure (Baddeley et al. 2015). We denote by $\{\tilde{x}_1, \ldots, \tilde{x}_p\}$ the retained points and we assume that they are ordered ($\tilde{x}_1 < \cdots < \tilde{x}_p$). We then have

$$
\begin{aligned}
ip_i(x) &\le \min_{1\le k\le p}\left(n, a^2(x-\tilde{x}_k)^2\right)\\
&= \sum_{k=1}^{p} a^2(x-\tilde{x}_k)^2\mathbf{1}_{\mathbf{B}(\tilde{x}_k,\sqrt{n}/a)}(x) + n\mathbf{1}_{[0,1]\setminus\bigcup\mathbf{B}(\tilde{x}_k,\sqrt{n}/a)}(x)\\
&= \begin{cases} a^2(x-\tilde{x}_k)^2 \text{ if } |x-\tilde{x}_k| < \sqrt{n}/a \text{ for some } k\in\{1,\ldots,p\}\\ n \text{ otherwise.}\end{cases}
\end{aligned}
$$

Let us denote by $f(x)$ this upper-bound and by $F(x)$ its primitive. In order to generate a sample from $p_i(x)$ on $[0, 1]$, we can use rejection sampling where the proposal has density $f(x)/F(1)$ and the rate of acceptance is $i/F(1)$. Simulation from the proposal can be done by the inversion method: if $U \sim \mathcal{U}([0, 1])$, $F^{-1}(F(1)U)$ is distributed from the proposal.

To complete this procedure, it remains to provide the formulas for $F(1)$ and $F^{-1}(x)$. Note that for all k,

$$
\int_{\tilde{x}_k-\sqrt{n}/a}^{\tilde{x}_k+\sqrt{n}/a} a^2(x-\tilde{x}_k)^2dx = \frac{2}{3a}n^{3/2}.
$$

By construction of $\{\tilde{x}_1, \ldots, \tilde{x}_p\}$, we deduce that

$$
F(1) = p\frac{2}{3a}n^{3/2} + n\left(1 - 2p\frac{\sqrt{n}}{a}\right) = n - \frac{4p}{3a}n^{3/2}. \quad (27)
$$

For $k = 1, \ldots, p$, we denote $y_k^- = F(\tilde{x}_k - \sqrt{n}/a)$, $y_k = F(\tilde{x}_k)$ and $y_k^+ = F(\tilde{x}_k + \sqrt{n}/a)$ and we have

$$
\begin{aligned}
y_k^- &= (k-1)\frac{2}{3a}n^{3/2} + n\left(\tilde{x}_k - \frac{\sqrt{n}}{a} - (k-1)2\frac{\sqrt{n}}{a}\right)\\
&= n\tilde{x}_k - \frac{4k-1}{3a}n^{3/2},\\
y_k &= y_k^- + \frac{1}{3a}n^{3/2} = n\tilde{x}_k - \frac{4k-2}{3a}n^{3/2},\\
y_k^+ &= y_k + \frac{1}{3a}n^{3/2} = n\tilde{x}_k - \frac{4k-3}{3a}n^{3/2}.
\end{aligned}
$$

For $\tilde{x}_k - \sqrt{n}/a \le x \le \tilde{x}_k$,

$$
\begin{aligned}
F(x) &= y_k^- + \int_{\tilde{x}_k-\sqrt{n}/a}^{x} a^2(t-\tilde{x}_k)^2dt = n\tilde{x}_k\\
&\quad -\frac{4k-2}{3a}n^{3/2} + \frac{a^2}{3}(x-\tilde{x}_k)^3.
\end{aligned}
$$

The same results holds true when $\tilde{x}_k \le x \le \tilde{x}_k + \sqrt{n}/a$. We deduce that for $y_k^- \le y \le y_k^+$,

$$
F^{-1}(y) = \tilde{x}_k + \left(\frac{3}{a^2}\left(y - n\tilde{x}_k + (4k-2)n^{3/2}/a\right)\right)^{1/3}.
$$

Finally, if $\tilde{x}_k + \sqrt{n}/a \le x \le \tilde{x}_{k+1} - \sqrt{n}/a$, then

$$
F(x) = y_k^+ + n(x - (\tilde{x}_k + \sqrt{n}/a)) = nx - \frac{4k}{3a}n^{3/2},
$$

so that for $y_k^+ \le y \le y_{k+1}^-$,

$$
F^{-1}(y) = \frac{1}{n}\left(y + \frac{4k}{3a}n^{3/2}\right).
$$

With this new proposal, the acceptance rate in the rejection sampling for the last step of Algorithm 2, i.e. for the simulation of the last point given the $n - 1$ first ones, becomes $1/F(1)$ instead of $1/n$ for a uniform proposal. For the most repulsive kernel (i.e. $J = \{-\ell, \ldots, \ell\}$ implying $n = 2\ell+1$), we obtain $a^2 \leq \pi^2 n(n^2-1)/3$, so that this new rate can be 3 times more effective, since by (27)

$$\frac{1/F(1)}{1/n} \geq \left(1 - \frac{4p}{\pi\sqrt{3(n^2-1)}}\right)^{-1},$$

which is greater than 3 if $p = n-1$ and $n > 10$.

C Existence of the inhomogeneous Gaussian-type DPP

The existence is equivalent to $0 \leq \lambda_i \leq 1$ for any $i \in \mathbb{N}^d$, where λ_i is given by (18). By this formula, since $B \leq 1$, we have that for any i, $\lambda_i \leq \lambda_0$ and the condition is verified if only if $\rho(2a/A)^{d/2} \leq 1$. Since $A = a + b + \sqrt{a^2 + 2ab}$, this is equivalent to $a(2\rho^{2/d} - 1) - b \leq \sqrt{a^2 + 2ab}$, which is (16).

D Generalised prolate spheroidal wave functions in dimension $d = 2$

We provide in this appendix some details on the spectral representation (22) of the Bessel type kernel when $d = 2$. As explained in Sect. 5.2, this amounts to describing the eigenvalues and the associated generalized prolate spheroidal eigenfunctions in (21), where $c = 2/\alpha$. The following formulas come from Slepian (1964) and Shkolnisky (2007). When $d = 2$, the sequence of eigenvalues in (21) is $\{\gamma_{N,n},\ N \geq 0,\ n \geq 0\}$ where the eigenvalues $\gamma_{0,n}$, $n \geq 0$, are simple, and the eigenvalues $\gamma_{N,n}$ for $N \geq 1$ and $n \geq 0$ are associated to $h(N,2) = 2$ eigenfunctions. Turning to polar coordinates by setting $x = (r\cos\theta, r\sin\theta)$ with $r \in [0,1]$ and $\theta \in [0, 2\pi]$, the eigenfunctions read for any $n \geq 0$,

$$\psi_{0,n}(x) = \frac{1}{\sqrt{2\pi}} R_{0,n}(r), \quad \forall x = (r\cos\theta, r\sin\theta) \in \mathbf{B},$$

and for any $N \geq 1$ and $n \geq 0$, for all $x = (r\cos\theta, r\sin\theta) \in \mathbf{B}$,

$$\psi_{N,n,1}(x) = \frac{1}{\sqrt{\pi}} R_{N,n}(r)\cos(N\theta),$$
$$\psi_{N,n,2}(x) = \frac{1}{\sqrt{\pi}} R_{N,n}(r)\sin(N\theta).$$

In these expressions $R_{N,n}(r)$ is given in terms of a series of Zernike polynomials:

$$\sqrt{r} R_{N,n}(r) = \sum_{k \geq 0} d_k^{N,n} T_{N,k}(r), \quad \forall r \in [0,1]. \tag{28}$$

In our notation the Zernike polynomials are defined for any $r \in [0,1]$ by

$$T_{N,k}(r) = \sqrt{2(2k+N+1)} r^{N+1/2} P_k^{N,0}(1-2r^2), \tag{29}$$

where $P_k^{N,0}$ are Jacobi polynomials, and they satisfy $< T_{N,k}, T_{N,l} >= \delta_{k,l}$. (Note that the Zernike polynomials in Slepian (1964) are not normalised, unlike in Shkolnisky (2007) and in our case, explaining a slight difference in the definition). Concerning the coefficients $(d_k^{N,n})_{k\geq 0}$ in (28), they satisfy the recurrence relation

$$b_{k,k-1}^N d_{k-1}^{N,n} + b_{k,k}^N d_k^{N,n} + b_{k,k+1}^N d_{k+1}^{N,n} = \chi^{N,n} d_k^{N,n},$$

where $\chi^{N,n} \in \mathbb{R}$ is an unknown quantity that does not depend on k and where the coefficients $b_{i,j}^N$, the expression of which is given below, are symmetric ($b_{i,j}^N = b_{j,i}^N$) and do not depend on n. This means that given N, the (infinite) vector $(d_k^{N,n})_{k\geq 0}$ is the n-th eigenvector associated to the eigenvalue $\chi^{N,n}$ of the symmetric tridiagonal (infinite) matrix B^N with entries $b_{i,j}$:

$$B^N = \begin{pmatrix} b_{0,0}^N & b_{0,1}^N & & & \\ b_{1,0}^N & b_{1,1}^N & b_{1,2}^N & & \\ & b_{2,1}^N & b_{2,2}^N & b_{2,3}^N & \\ & & \ddots & \ddots & \ddots \end{pmatrix}.$$

Here, for any $k \geq 0$,

$$b_{k,k-1}^N = c^2 \frac{k(k+N)}{(2k+N)\sqrt{2k+N+1}\sqrt{2k+N-1}},$$
$$b_{k,k}^N = -\frac{c^2}{2}\left(1 + \frac{N^2}{(2k+N)(2k+N+2)}\right)$$
$$- (2k+N+1/2)(2k+N+3/2),$$
$$b_{k,k+1}^N = c^2 \frac{(k+1)(N+k+1)}{(2k+N+2)\sqrt{2k+N+1}\sqrt{2k+N+3}},$$

and we recall that we have set $c = 2/\alpha$.

For the numerical computation of (28), one needs for each N to truncate the matrix B^N at some size $k_{\max}$, depending on the expected accuracy, which allows to deduce all coefficients $d_k^{N,n}$ for $k \leq k_{\max} - 1$ and $n \leq k_{\max} - 1$. The choice of $k_{\max}$ is discussed in Greengard and Serkh (2018). Note that in view of the orthonormal property of the Zernike polynomials, i.e. $< T_{N,k}, T_{N,l} >= \delta_{k,l}$, the eigenfunctions $\psi_{0,n}$, $\psi_{N,n,1}$ and $\psi_{N,n,2}$ are orthonormal whenever the coefficients $(d_k^{N,n})_{k\geq 0}$ are normalised, i.e. $\sum_{k\geq 0}(d_k^{N,n})^2 = 1$, which is numerically achieved by normalising the eigenvectors of the (truncated) matrix B^N.

Concerning the eigenvalues in (22), they are equal to $2\pi\rho\alpha\lambda_{N,n}^2(2/\alpha)$ where $\lambda_{N,n}(c)$, $c = 2/\alpha$, satisfies (see (12) in Shkolnisky (2007)):

$$\lambda_{N,n}(c)\sqrt{r}R_{N,n}(r) = \int_0^1 J_N(crr')\sqrt{crr'}\sqrt{r'}R_{N,n}(r')dr'. \tag{30}$$

Using (28) and (29), we have for the left hand side

$$\lambda_{N,n}(c)\frac{\sqrt{r}R_{N,n}(r)}{r^{N+1/2}} = \lambda_{N,n}(c)$$
$$\sum_{k\geq 0} d_k^{N,n}\sqrt{2(2k+N+1)}P_k^{N,0}(1-2r^2)$$
$$\underset{r\to 0}{\sim} \lambda_{N,n}(c)\sum_{k\geq 0} d_k^{N,n}\sqrt{2(2k+N+1)}\binom{N+k}{k}, \tag{31}$$

where the equivalence comes from the fact that for a Jacobi polynomial, $P_k^{N,0}(1) = \binom{N+k}{k}$. For the right hand side in (30), we have at $r = 0$

$$\frac{J_N(crr')}{r^{N+1/2}}\sqrt{crr'}\sqrt{r'}R_{N,n}(r') \sim \frac{(cr')^{N+1/2}}{2^N N!}\sqrt{r'}R_{N,n}(r'),$$

so that using (28) again

$$\int_0^1 J_N(crr')\sqrt{crr'}\sqrt{r'}R_{N,n}(r')dr' \sim \frac{c^{N+1/2}}{2^N N!}$$
$$\sum_{k\geq 0} d_k^{N,n}\int_0^1 r'^{N+1/2}T_{N,k}(r')dr'.$$

Since $T_{N,0}(r) = \sqrt{2(N+1)}r^{N+1/2}$ and using the fact that $< T_{N,0}, T_{N,k} >= \delta_{0,k}$, we obtain

$$\int_0^1 J_N(crr')\sqrt{crr'}\sqrt{r'}R_{N,n}(r')dr' \sim \frac{c^{N+1/2}d_0^{N,n}}{2^N N!\sqrt{2(N+1)}}. \tag{32}$$

In view of (30), the two right hand side expressions in (31) and (32) are equal, which gives

$$\lambda_{N,n}(c) = \frac{c^{N+1/2}d_0^{N,n}}{2^{N+1}N!\sqrt{N+1}\sum_{k\geq 0} d_k^{N,n}\sqrt{N+2k+1}\binom{N+k}{k}}. \tag{33}$$

For the numerical aspects, formula (33) may be implemented directly, or only for $\lambda_{N,0}$ and then a recursive relation can be used to get $\lambda_{N,n}$ for $n \geq 1$, as advised in dimension $d = 1$ in Osipov et al. (2013) through their relation (7.13), an approach generalised in $d \geq 2$ in Lederman (2017); Greengard and Serkh (2018).

References

Affandi, R.H., Fox, E., Taskar, B.: Approximate inference in continuous determinantal processes. In: Advances in Neural Information Processing Systems (2013), pp. 1430–1438

Affandi, R.H., Kulesza, A., Fox, E., Taskar, B.: Nystrom approximation for large-scale determinantal processes. In: Carvalho, C.M., Ravikumar, P. (eds.) Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics (Scottsdale, Arizona), Proceedings of Machine Learning Research, PMLR, vol. 31, pp. 85–98 (2013)

Baddeley, A.J., Rubak, E., Turner, R.: Spatial Point Patterns: Methodology and Applications with R. Interdisciplinary Statistics. Chapman & Hall/CRC, Boca Raton (2015)

Bardenet, R., Hardy, A.: Monte Carlo with determinantal point processes. Ann. Appl. Probab. **30**(1), 368–417 (2020)

Belhadji, A., Bardenet, R., Chainais, P.: Kernel quadrature with DPPs. In: Advances in Neural Information Processing Systems, pp. 12927–12937 (2019)

Biscio, C., Lavancier, F.: Quantifying repulsiveness of determinantal point processes. Bernoulli **22**, 2001–2028 (2016)

Biscio, C., Lavancier, F.: Contrast estimation for parametric stationary determinantal point processes. Scand. J. Stat. **44**, 204–229 (2017)

Bremer, J.: On the numerical evaluation of the prolate spheroidal wave functions of order zero. Appl. Comput. Harmon. Anal. **60**, 53–76 (2022)

Coeurjolly, J.-F., Lavancier, F.: Understanding spatial point patterns through intensity and conditional intensities. In: Coupier, D. (ed.) Highlights in Stochastic Geometry, Lecture Notes in Mathematics, vol. 2237, pp. 45–85. Springer (2019)

Coeurjolly, J.-F., Mazoyer, A., Amblard, P.-O.: Monte Carlo integration of non-differentiable functions on $[0, 1]^\iota$, $\iota = 1, \ldots, d$, using a single determinantal point pattern defined on $[0, 1]^d$. Electron. J. Stat. **15**(2), 6228–6280 (2021)

Decreusefond, L., Flint, I., Privault, N., Torrisi, G.L.: Determinantal point processes. In: Peccati, G., Reitzner, M. (eds.) Stochastic Analysis for Poisson Point Processes: Malliavin Calculus, Wiener-Itô Chaos Expansions and Stochastic Geometry, pp. 311–342. Springer, Cham (2016)

Decreusefond, L., Flint, I., Vergne, A.: A note on the simulation of the Ginibre point process. J. Appl. Probab. **52**(4), 1003–1012 (2015)

Deng, N., Zhou, W., Haenggi, M.: The Ginibre point process as a model for wireless networks with repulsion. IEEE Trans. Wirel. Commun. **14**(1), 107–121 (2014)

Derezinski, M., Calandriello, D., Valko, M.: Exact sampling of determinantal point processes with sublinear time preprocessing. In: Advances in Neural Information Processing Systems. Curran Associates, Inc., vol. 32, pp. 11546–11558 (2019)

Gillenwater, J., Kulesza, A., Mariet, Z., Vassilvtiskii, S.: A tree-based method for fast repeated sampling of determinantal point processes. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning (Long Beach, California), Proceedings of Machine Learning Research, vol. 97, pp. 2260–2268 (2019)

Ginibre, J.: Statistical ensembles of complex, quaternion, and real matrices. J. Math. Phys. **6**(3), 440–449 (1965)

Goldman, A.: The palm measure and the Voronoi tessellation for the Ginibre process. Ann. Appl. Probab. **20**(1), 90–128 (2010)

Gradshteyn, I.S., Ryzhik, I.M.: Table of Integrals, Series, and Products. Academic Press (2014)

Grafakos, L.: Classical Fourier Analysis. Graduate Texts in Mathematics. Springer (2008)

Greengard, P., Serkh, K.: On generalized prolate spheroidal functions. arXiv:1811.02733 (2018)

Hough, J.B., Krishnapur, M., Peres, Y., Virág, B.: Determinantal processes and independence. Probab. Surv. **3**, 206–229 (2006)

Kulesza, A., Taskar, B.: Determinantal point processes for machine learning. Found. Trends Mach. Learn. **5**, 123–286 (2012)

Launay, C., Galerne, B., Desolneux, A.: Exact sampling of determinantal point processes without eigendecomposition. J. Appl. Probab. **57**(4), 1198–1221 (2020)

Lavancier, F., Møller, J., Rubak, E.: Determinantal point process models and statistical inference: extended version. arXiv:1205.4818v5 (2014)

Lavancier, F., Møller, J., Rubak, E.: Determinantal point process models and statistical inference. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **77**, 853–877 (2015)

Lavancier, F., Poinas, A., Waagepetersen, R.: Adaptive estimating function inference for nonstationary determinantal point processes. Scand. J. Stat. **48**(1), 87–107 (2021)

Lederman, R.R.: Numerical algorithms for the computation of generalized prolate spheroidal functions. arXiv:1710.02874 (2017)

Li, C., Sra, S., Jegelka, S.: Fast mixing Markov chains for strongly Rayleigh measures, DPPs, and constrained sampling. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 29. Curran Associates, Inc., pp. 4188–4196 (2016)

Lyons, R.: Determinantal probability: basic properties and conjectures. In: Proceedings of the International Congress of Mathematicians, Seoul, Korea IV, pp. 137–161 (2014)

Macchi, O.: The coincidence approach to stochastic point processes. Adv. Appl. Probab. **7**, 83–122 (1975)

Miyoshi, N., Shirai, T.: A cellular network model with Ginibre configured base stations. Adv. Appl. Probab. **46**(3), 832–845 (2014)

Møller, J., O'Reilly, E.: Couplings for determinantal point processes and their reduced Palm distributions with a view to quantifying repulsiveness. J. Appl. Probab. **58**(2), 469–483 (2021). https://doi.org/10.1017/jpr.2020.101

Møller, J., Waagepetersen, R.: Statistical Inference and Simulation for Spatial Point Processes. Chapman and Hall/CRC, Boca Raton (2004)

Moore, I.C., Cada, M.: Prolate spheroidal wave functions, an introduction to the Slepian series and its properties. Appl. Comput. Harmon. Anal. **16**(3), 208–230 (2004)

Osipov, A., Rokhlin, V., Xiao, H.: Prolate spheroidal wave functions of order zero. In: Springer Series in Applied Mathematics Science, vol. 187 (2013)

Poulson, J.: High-performance sampling of generic determinantal point processes. Philos. Trans. R. Soc. A **378**(2166), 20190059 (2020)

R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna (2022)

Shirai, T., Takahashi, Y.: Random point fields associated with certain Fredholm determinants I: fermion, Poisson and boson point processes. J. Funct. Anal. **205**, 414–463 (2003)

Shkolnisky, Y.: Prolate spheroidal wave functions on a disc-integration and approximation of two-dimensional bandlimited functions. Appl. Comput. Harmon. Anal. **22**(2), 235–256 (2007)

Slepian, D.: Prolate spheroidal wave functions, Fourier analysis and uncertainty-IV: extensions to many dimensions; generalized prolate spheroidal functions. Bell Syst. Tech. J. **43**(6), 3009–3057 (1964)

Slepian, D., Pollak, H.O.: Prolate spheroidal wave functions, Fourier analysis and uncertainty-I. Bell Syst. Tech. J. **40**(1), 43–63 (1961)

Tremblay, N., Barthelmé, S., Usevich, K., Amblard, P.-O.: Extended l-ensembles: a new representation for determinantal point processes. Ann. Appl. Probab. **33**(1), 613–640 (2023)

Williams, C.K., Rasmussen, C.E.: Gaussian Processes for Machine Learning, vol. 2. MIT Press, Cambridge (2006)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.