FAST HIGH-DIMENSIONAL NODE GENERATION WITH VARIABLE DENSITY

O. VLASIUK, T. MICHAELS, N. FLYER, AND B. FORNBERG

Abstract. We present an algorithm for producing discrete distributions with a prescribed nearest-neighbor distance function. Our approach is a combination of quasi-Monte Carlo (Q-MC) methods and weighted Riesz energy minimization: the initial distribution is a stratified Q-MC sequence with some modifications; a suitable energy functional on the configuration space is then minimized to ensure local regularity. The resulting node sets are good candidates for building meshless solvers and interpolants, as well as for other purposes where a point cloud with a controlled separation-covering ratio is required. Applications of a three-dimensional implementation of the algorithm, in particular to atmospheric modeling, are also given.

1. Introduction

1.1. **RBFs and meshless methods.** In a number of important applications, usefulness of meshless methods in general, and of radial basis functions (RBFs) in particular, is well-known. They have found their way into high-dimensional interpolation, machine learning, spectral methods, vector-valued approximation and interpolation, just to name a few [42, 13, 19, 8, 38]. RBFs have multiple advantages, most importantly extreme flexibility in forming stencils (in the case of RBF-FD) and high local adaptivity; allowing spectral accuracy on irregular domains; the fact that the corresponding interpolation matrix (denoted by $\boldsymbol{A}$ below) is positive-definite for several types of radial functions and does not suffer from instability phenomena characteristic of some of the alternative interpolation methods.

Applying RBF-FD stencils to building solvers requires an efficient way of distributing the *nodes* of basis elements in the domain, which can be either a solid or a surface. The tasks of modeling and simulation often call for massive numbers of nodes, so it is important to ensure that the distribution process is easily scalable. One further has to be able to place the RBFs according to a certain density, as a method of local refinement, for example, at the boundary, or in regions of special interest. Yet another challenge arises when it is necessary to deal with complex or non-smooth domains and/or surfaces.

Recall [21] that an RBF is a linear combination of the form

$$s(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \phi(\|\boldsymbol{x} - \boldsymbol{x}_k\|), \tag{1}$$

where $\phi(\cdot)$ is a radial function, and $\boldsymbol{x}_k,\ k = 1, \dots, K$, is a collection of pairwise distinct points in $\mathbb{R}^d$. A common choice of ϕ is the Gaussian $\phi(r) = e^{-(\epsilon r)^2}$, although one may also use $1/(1 + (\epsilon r)^2)$, $r^{2p}\log(r),\ p \in \mathbb{N}$, etc. In this discussion, we are not concerned with the distinctions between the

Date: May 9, 2018.

Key words and phrases. Quasi-uniformity, irrational lattice, quasi-Monte Carlo methods, Riesz energy, gradient descent, separation distance, nearest neighbors.

different radial kernels, so the reader can assume that $\phi(r) = e^{-(\epsilon r)^2}$. In contrast to pseudospectral methods [19], RBF-FD approach means that to obtain a useful approximation of a function, or a differential operator, the nodes in expressions like (1) must be in the vicinity of the point $\boldsymbol{x}$, and therefore a large number of stencils are constructed throughout the underlying set. It is well-known that the matrix

$$\boldsymbol{A} = \begin{bmatrix} \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|) & \dots & \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_K\|) \\ \phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_2\|) & \dots & \phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_K\|) \\ \vdots & \vdots & & \vdots \\ \phi(\|\boldsymbol{x}_K - \boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_K - \boldsymbol{x}_2\|) & \dots & \phi(\|\boldsymbol{x}_K - \boldsymbol{x}_K\|) \end{bmatrix}$$

is positive-definite if the nodes $\boldsymbol{x}_1 \dots \boldsymbol{x}_K$ are all distinct [37], and so under this assumption there exist K-point RBF interpolants for any function data. A different question, however, is whether the matrix $\boldsymbol{A}$ will be well-conditioned: it is not the case, for example, when the nodes are placed on a lattice and $\epsilon \to 0$, [23]. The other extreme, having low regularity, also does not provide a reliable source of nodes, as can be seen on the example of the Halton sequence [23]. Furthermore, node clumping can lead to instability of PDE solvers, [19]. To avoid this, one must guarantee that the nodes are separated. In effect, generally the quasi-uniform node sets generated by the present algorithm, or, for example, the one constructed by the third and fourth authors [22], perform better than either lattice or the Halton sequence.

In many applications, one has to ensure that the distance from a node $\boldsymbol{x}$ to its nearest neighbor behaves approximately as a function of the position of the node [22]. Prescribing this function, $\rho(\boldsymbol{x})$, which we call the *radial density*, is a natural way to treat the cases when a local refinement is required in order to capture special features of the domain. In the present paper we will describe a method of node placement for which the actual distance to the nearest neighbor, denoted by $\Delta(\boldsymbol{x}) = \min_{\boldsymbol{x}' \neq \boldsymbol{x}} \|\boldsymbol{x}' - \boldsymbol{x}\|$, satisfies the above description. To summarize, we are interested in a procedure for obtaining discrete configurations inside a compact set that will:

- guarantee that $\Delta(\boldsymbol{x}) \asymp \rho(\boldsymbol{x})$ (that is, differ only up to a constant factor) for a given function $\rho(\boldsymbol{x})$ with a reasonably wide choice of ρ;
- be suitable for mesh-free PDE discretizations using RBFs, i.e., produce well-separated configurations without significant node alignment;
- result in quasi-uniform node distributions also on the surface boundaries of the domain;
- be computationally efficient, easily scalable, and suitable for parallelization.

1.2. **Notation and layout.** The bold typeface is reserved for vectors in $\mathbb{R}^d$. With few exceptions, letters of the Greek alphabet denote functions, calligraphic letters and Ω denote sets, and the regular Roman typeface is used for scalar variables. The symbolic notation employed throughout the paper is summarized in Table 1.

The paper is structured as follows: Section 2.1 gives a brief sketch of common practices involving RBFs; Sections 2.2 and 2.3 introduce the two essential components of our approach, Riesz energy functionals and quasi-Monte Carlo methods. The main algorithm and its discussion are the subjects of Sections 3.1 and 3.2, respectively. Sections 4.1 and 4.3 offer applications of the algorithm, and Section 5 contains the final remarks. The Appendix is dedicated to numerical experiments with the mean and minimal separation distance of Riesz minimizers and irrational lattices.

2. Choice of method

2.1. **Computations with radial basis functions.** In this section we shall outline the common practices involving RBFs, in order to motivate the requirements that have to be imposed on the

Symbol	Description
$\bar{x}$	mean value of x
$\alpha_1, \alpha_2, \ldots, \alpha_{d-1}$	fixed linearly independent irrational numbers
$\mathcal{C}$	unit cube $[0,1]^d$
C_1, C_2	positive constants in (6) and (8)
$\boldsymbol{c}_m$	the closest to the origin corner of $\mathcal{V}_m$
$\chi(\cdot\,;\Omega)$	characteristic function of the set Ω
$\mathcal{D}$	$m \in \mathcal{D}$ if $\mathcal{V}_m$ has nonempty neighbors on `Step 1`
$\Delta(\boldsymbol{x})$	distance from the node $\boldsymbol{x}$ to the nearest neighbor
$\Delta(\{\boldsymbol{x}_1, \ldots \boldsymbol{x}_N\})$	separation of the configuration
$\Delta^k(\boldsymbol{x}),\ k = 1, \ldots, K$	distance from $\boldsymbol{x}$ to the k-th nearest neighbor
Δ_n	minimal separation of periodized $\mathcal{L}_n$, $\mathcal{M}_n$, p. 8
$\bar{\Delta}_n$	mean separation of the periodized $\mathcal{L}_n$, $\mathcal{M}_n$, p. 8
$\mathcal{E}$	$m \in \mathcal{E}$ if $\mathcal{V}_m$ is empty after `Step 2`
$E_s,\ E_s^\kappa$	Riesz s-energy and weighted s-energy, (3)-(4)
$\mathrm{frac}(x)$	fractional part of the nonnegative number x
$\mathcal{H}_d$	d-dimensional Hausdorff measure
κ	kernel of the weighted Riesz energy, (4)
λ	interpolated inverse of $\bar{\Delta}_n$, p. 8
$\mathcal{L}_n,\ n \geq 1$	n-point irrational lattice, (6)
$\mathcal{M}_n,\ n \geq 1$	n-point periodic Riesz minimizer, p. 6
$\mathcal{L}_n',\ \mathcal{M}_n'$	translated and rescaled $\mathcal{L}_n$ and $\mathcal{M}_n$, (9)-(10)
n_m	number of nodes in $\mathcal{V}_m$
Ω	target distribution support
$\Omega_{\mathrm{etopo}},\ \Omega_{\mathrm{shell}}$	the underlying sets in Sections 4.1 and 4.3
$(r,\, a,\, p)$	spherical coordinates, p. 11
$\rho(\boldsymbol{x})$	objective radial density, pp. 2, 7
$\mathcal{V}_m,\ m = 1, \ldots, M^d$	cube-shaped voxel in $\mathbb{R}^d$, p. 6
$\boldsymbol{x};\ \boldsymbol{x}_i,\ i = 1, \ldots, N$	points in $\mathbb{R}^d$; nodes of the configuration
$\boldsymbol{x}_{j(i,k)},\ k = 1, \ldots, K$	the k-th nearest neighbor of $\boldsymbol{x}_i$, (4)
$\boldsymbol{z}_m$	center of $\mathcal{V}_m$

TABLE 1. Symbolic notation employed throughout the paper.

node distribution used in the respective computations. For a more in-depth discussion see one of [21, 42, 10, 18]. A significant portion of the RBF approach hinges on the theory of positive-definite functions; in fact, the latter have experienced a revival over the recent decades precisely due to its computational ramifications; we refer the reader to [14, 9, 33].

Let $\mathcal{F}$ be a space of continuous functions $\psi : \mathbb{R}^d \mapsto \mathbb{R}$, possessing perhaps additional smoothness properties, and assume that $\mathfrak{L}$ is a linear operator acting on $\mathcal{F}$ and bounded on it. A generalization of the standard [20] finite-difference approach consists in approximating the action of $\mathfrak{L}$ on $\mathcal{F}$ pointwise, that is, for every pair of point sets $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$ and $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L\}$ one constructs an L-by-K matrix $\boldsymbol{M}$ such that the vector of values of $\psi(\boldsymbol{x}_k)$ and the vector of values of $\mathfrak{L}\psi(\boldsymbol{y}_l) = \mathfrak{L}(\psi)(\boldsymbol{y}_l)$ are related

by the linear mapping,

$$\mathfrak{L}\psi(\boldsymbol{y}_1,\dots,\boldsymbol{y}_L)^{\mathrm{tr}} = \boldsymbol{M}\,\psi(\boldsymbol{x}_1,\dots,\boldsymbol{x}_K)^{\mathrm{tr}}, \tag{2}$$

whenever $\psi \in \mathcal{G}$ for some subspace $\mathcal{G} \subset \mathcal{F}$. Here $\psi(\boldsymbol{x}_1,\dots,\boldsymbol{x}_K)^{\mathrm{tr}} = (\psi(\boldsymbol{x}_1),\dots,\psi(\boldsymbol{x}_K))^{\mathrm{tr}}$ denotes a K-dimensional vector; similarly, $\mathfrak{L}\psi(\boldsymbol{y}_1,\dots,\boldsymbol{y}_L)^{\mathrm{tr}} = (\mathfrak{L}\psi(\boldsymbol{y}_1),\dots,\mathfrak{L}\psi(\boldsymbol{y}_L))^{\mathrm{tr}}$ is L-dimensional. The equation (2) is especially useful when $\mathfrak{L}$ is a differential or an interpolation operator, so that for example the function ψ is known only at the points $\{\boldsymbol{x}_k\}$, but not at $\{\boldsymbol{y}_l\}$. In order for this approach to be practical, the subspace $\mathcal{G}$ has to be finite-dimensional; since on the other hand all the important spaces $\mathcal{F}$ are infinite-dimensional, $\mathcal{G}$ must depend on both $\{\boldsymbol{x}_k\}$ and $\{\boldsymbol{y}_l\}$; the same dependence then applies to the matrix $\boldsymbol{M}$.

To see what the computation of $\boldsymbol{M}$ involves, let us fix the sets $\{\boldsymbol{x}_1,\dots\boldsymbol{x}_K\}$ and $\{\boldsymbol{y}_1,\dots\boldsymbol{y}_L\}$, and thus the space $\mathcal{G}$. It is reasonable to assume $\mathcal{G}$ has the rank K, so that given values of ψ at the distinct vectors $\boldsymbol{x}_1,\dots,\boldsymbol{x}_K$, there exists a unique $s \in \mathcal{G}$ such that $s(\boldsymbol{x}_k) = \psi(\boldsymbol{x}_k)$, $1 \le k \le K$. Indeed, expanding s in a fixed basis of $\mathcal{G}$, denoted by $\phi_1,\dots,\phi_K$, we have

$$s(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \phi_k(\boldsymbol{x}).$$

If $\mathcal{G}$ had higher rank, we would need to introduce additional regularization constraints. Clearly, (1) is a special case of the previous equation. Observe that the vector of coefficients $\boldsymbol{w} = (w_1,\dots,w_K)^{\mathrm{tr}}$ of interpolant s is the solution of

$$\boldsymbol{A}_{\mathcal{G}}\boldsymbol{w} = \psi(\boldsymbol{x}_1,\dots,\boldsymbol{x}_K)^{\mathrm{tr}} = s(\boldsymbol{x}_1,\dots,\boldsymbol{x}_K)^{\mathrm{tr}},$$

with $\boldsymbol{A}_{\mathcal{G}} = [\phi_k(\boldsymbol{x}_j)]_{j,k=1}^{K}$; we used the same notation $s(\boldsymbol{x}_1,\dots,\boldsymbol{x}_K)^{\mathrm{tr}}$ for the vector of values $s(\boldsymbol{x}_k)$, as before for the function ψ. This yields

$$\mathfrak{L}s(\boldsymbol{y}_1,\dots,\boldsymbol{y}_L)^{\mathrm{tr}} = \sum_{k=1}^{K} w_k\, \mathfrak{L}\phi_k(\boldsymbol{y}_1,\dots,\boldsymbol{y}_L)^{\mathrm{tr}} = \boldsymbol{D}_{\mathcal{G}}\boldsymbol{w},$$

where we used the linearity of $\mathfrak{L}$ and the notation $\boldsymbol{D}_{\mathcal{G}} = [\mathfrak{L}\phi_k(\boldsymbol{y}_l)]_{l,k=1}^{L,K}$. Since $\psi \in \mathcal{G}$ would imply that s coincides with ψ, the above equation for $\boldsymbol{w}$ gives for (2)

$$\boldsymbol{M} = \boldsymbol{D}_{\mathcal{G}}\boldsymbol{A}_{\mathcal{G}}^{-1}\psi(\boldsymbol{x}_1,\dots,\boldsymbol{x}_K)^{\mathrm{tr}},$$

where $\boldsymbol{A}_{\mathcal{G}} = [\phi_k(\boldsymbol{x}_j)]_{j,k=1}^{K}$ and $\boldsymbol{D}_{\mathcal{G}} = [\mathfrak{L}\phi_k(\boldsymbol{y}_l)]_{l,k=1}^{L,K}$. This brings us to the question of how well-conditioned is the matrix $\boldsymbol{A}_{\mathcal{G}}$; observe that when the basis of the space $\mathcal{G}$ is given by $\{f(\|\boldsymbol{x}-\boldsymbol{x}_k\|^2)\}$ for a non-constant completely monotone f, the matrix $\boldsymbol{A}_{\mathcal{G}}$ is (strictly) positive-definite [37, 14, Chapter 15], thus justifying the use of the function $\phi(r) = e^{-(\epsilon r)^2}$ as the RBF kernel. For the other commonly used kernels, existence of $\boldsymbol{A}_{\mathcal{G}}^{-1}$ follows from Micchelli's theorem [33, 14, Chapter 16].

The above explains the requirement that the nodes used for RBF computations be separated: a close pair $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ leads to a pair of almost equal rows in $\boldsymbol{A}_{\mathcal{G}}$, which then becomes ill-conditioned. Furthermore, taking the limit $\epsilon \to 0$ can cause the interpolant s to diverge for certain RBF kernels [25, 11]; this phenomenon however does not occur for $\phi(r) = e^{-(\epsilon r)^2}$. For other kernels, such divergence is connected to the notion of unisolvency, so that it only occurs if the interpolant in a space of polynomials of fixed degree is not uniquely defined by its values at the nodes $\{\boldsymbol{x}_k\}$, as is the case when the $\{\boldsymbol{x}_k\}$ lie on a lattice [11]. We conclude the discussion by observing that the motivation for considering the "increasingly flat" limit $\epsilon \to 0$ is that the resulting RBFs can be used to obtain highly accurate solutions of elliptic problems and approximants of smooth data [25].

2.2. **Riesz energy.** To generate nodes both devoid of lattice alignment and having near-optimal local separation, we apply the energy minimization technique. Let us first introduce the *Riesz s-energy* of a collection $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ as

$$E_s(\{\boldsymbol{x}_1, \dots \boldsymbol{x}_N\}) = \sum_{i \neq j} \frac{1}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^s}. \tag{3}$$

There exists extensive literature dedicated to this and derived functionals for $s \geq d$, their asymptotics and limiting measures, see for example [28, 4, 6]. The paper [4] introduces multiplicative weights in (3) which we will now discuss. The *weighted Riesz s-energy* with kernel κ is defined as

$$E_s^\kappa(\{\boldsymbol{x}_1, \dots \boldsymbol{x}_N\}) = \sum_{i \neq j} \frac{\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^s}.$$

It was proved [4], that the counting measures of the minimizers of the weighted energy converge to $\kappa(\boldsymbol{x}, \boldsymbol{x})^{-d/s} d\mathcal{H}_d(\boldsymbol{x})$, with $\mathcal{H}_d$ denoting the d-dimensional Hausdorff measure. It will be useful to assume that the kernel κ is chosen so that all the terms, except when $\boldsymbol{x}_j$ is among the K nearest nodes to $\boldsymbol{x}_i$, are zero, a condition equivalent to *truncated* weights in [5]. Under this assumption, the expression for E_s^κ can be rewritten as

$$E_s^\kappa(\{\boldsymbol{x}_1, \dots \boldsymbol{x}_N\}) = \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{\kappa\big(\boldsymbol{x}_i, \boldsymbol{x}_{j(i,k)}\big)}{\|\boldsymbol{x}_i - \boldsymbol{x}_{j(i,k)}\|^s}, \tag{4}$$

where nodes $\boldsymbol{x}_{j(i,k)}$, $k = 1, \dots, K$, are the K nearest neighbors of $\boldsymbol{x}$. The form (4) will make E_s^κ the most useful for our purposes. As described in detail in [5], the truncated kernel has the same asymptotics and limiting distribution as the full weighted Riesz s-energy, yet can be computed in the amount of time of order $O(NK)$, unlike the $O(N^2)$ order computation time for the complete kernel. The configurations which minimize this functional over n-point subsets of a compact set are well-separated, that is, the quantity $\Delta(\{\boldsymbol{x}_1, \dots \boldsymbol{x}_N\}) = \min_i \Delta(\boldsymbol{x}_i)$ satisfies

$$\Delta(\{\boldsymbol{x}_1, \dots \boldsymbol{x}_N\}) \geq C N^{-1/d}. \tag{5}$$

for all N. The value of the exponent s is chosen so that $s \geq d$ to ensure that the energy functional is sufficiently repulsive; it is known from the classical potential theory that for $s < d$ the minimal energy configurations are not necessarily uniform, and their local structure depends on the shape of the domain [30]. Property (5) holds for any $s > d$, when minimizing the energy over any fixed compact set $\Omega \subset \mathbb{R}^d$. While any symmetric kernel that grows fast enough towards the diagonal of $\Omega \times \Omega$ and smooth away from it would produce similar results, we chose the Riesz kernel because the properties of its minimizers are well understood.

Note that simply looking for minimizers of the weighted Riesz s-energy doesn't provide node sets satisfying $\Delta(\boldsymbol{x}) \asymp \rho(\boldsymbol{x})$; in fact, boundary nodes of such minimizers will often have smaller separations than desired. Since in such cases the boundary has a lower Hausdorff dimension, it does not influence the volumetric density, which agrees with the results above. With this motivation in mind, we are ready to introduce the second component of our method.

2.3. **Quasi-Monte Carlo methods.** To facilitate convergence of whichever optimization algorithm is used to find minimizers of (4), we can initialize it with a configuration that approximates the limiting measure. One has to rule out Monte Carlo methods due to the separation requirement: random points exhibit clustering [7], which makes deterministic post-processing, in particular by energy minimization, costly; on the other hand, purely stochastic approaches do not generally yield satisfying results (see for example *thinning* discussed in [32]). Instead, we turn to the quasi-Monte

Carlo (Q-MC) approach. As has been pointed out at the end of Section 2.1, in order to ensure convergence of RBF-FD interpolants, the underlying node set must be (locally) unisolvent; for our purposes this just means that the nodes are in a generic position with respect to each other. The latter is clearly not the case for lattice-like Q-MC configurations, which explains why we resort to energy minimization. On the other hand, we choose not to use other popular Q-MC sequences, such as Halton nodes, since they are not necessarily better-conditioned than lattice nodes [24, Figure 5.1], and are harder to handle when the distribution support Ω has complex geometry.

The key element of our construction lies in distributing the node set in a deterministic way so that to guarantee low discrepancy between the desired and the obtained radial densities. This is achieved by a Q-MC analog of the *stratification* of the Monte Carlo method [12]: nodes are distributed with piecewise constant (radial) density that approximates the desired one. We consider two different Q-MC sequences to draw from with (near-)constant radial density: irrational lattices and periodic Riesz minimizers. After dividing the set Ω into cube-shaped *voxels*, each voxel is filled with nodes obtained in one of the two ways, appropriately scaled, then the weighted s-energy (4) of the whole node set is minimized. Although we discuss the radial density in the present paper, an argument for the volumetric density can be produced along the same lines.

Yet another reason to make use of a Q-MC sequence is to avoid recursive data structures, which in some cases can be detrimental to the overall performance. Even though such structures have seen significant developments over the years, both dynamic update and parallelization for them remain challenging, [36, 43]. The approach of the present paper should therefore be understood as almost opposite to the well-known "quadtree" algorithm [27], that indeed has been used for meshless node generation [40]. Namely, as outlined above, our algorithm places nodes en masse inside the voxels to produce a rough approximation of the target distribution, and subsequently adjusts them by a gradient flow, which is straightforward to parallelize. Although this does involve the computation of the nearest neighbors in (4), which in practice will be done by constructing a k-d tree, by initializing the node configuration with a stratified Q-MC sequence we ensure the indices $j(i,k)$ in (4) will not undergo significant changes during the energy minimization stage, so the k-d tree will not require intensive updates.

An *irrational lattice* (IL) is defined as a discrete subset of the d-dimensional unit cube $[0,1]^d$

$$\mathcal{L}_n = \left\{ \left(\operatorname{frac}(C_1 + i/n), \operatorname{frac}(i\alpha_1), \{i\alpha_2\}, \dots, \operatorname{frac}(i\alpha_{d-1}) \right) \ : \ i = 1, \dots, n \right\}, \tag{6}$$

where $\operatorname{frac}(x) = \operatorname{mod}(x,1) = x - \lfloor x \rfloor$ denotes the fractional part of x, $C_1 > 0$ is fixed, and $\alpha_1, \alpha_2, \dots, \alpha_{d-1}$ are irrational numbers, linearly independent over the rationals. This terminology seems to be accepted in the low-discrepancy community [2], while closely related objects, when used for Q-MC purposes, are known as *Korobov/lattice point sets* [31].

The motivation for using an IL in this context is due to the existing results on the discrepancy of ILs. It is known for example, that the two-dimensional ILs have the optimal order of L^2 discrepancy, [3, 2]. Furthermore, in all dimensions ILs are uniformly distributed [29, Chapter 1.6], that is, the fraction of lattice points inside any rectangular box with faces parallel to the coordinate planes converges to its volume. The simple linear structure of ILs makes them especially attractive for SIMD-parallelization.

Another Q-MC sequence that has proven to suit our purposes consists of *periodic Riesz minimizers* on the unit flat torus, that is, n-point collections $\mathcal{M}_n = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ that minimize (3) on $([0,1]^d)^n$ with the Euclidean distance $\|\cdot\|$ replaced by the periodic metric

$$\|\boldsymbol{x} - \boldsymbol{y}\|_{\sim}^2 = \Pi(x_1 - y_1) + \Pi(x_2 - y_2) + \Pi(x_3 - y_3), \tag{7}$$

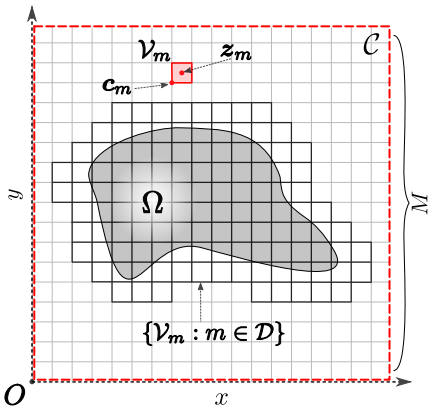

FIGURE 1. An illustration of some of the symbolic notation used in the algorithm below, in the case $d = 2$.

where $\Pi(x) = \min(x^2, (1-x)^2)$, $0 \le x \le 1$. It follows from [28] that such configurations have optimal separation and asymptotically uniform volumetric density. It follows from the numerics also, that in this case the nearest neighbor distances vary very little from node to node; this and that minimizing configurations do not suffer from the lattice-like alignment, makes their rescaled copies good candidates for the stratification.

The number of nodes in individual voxels is defined by the function ρ, so the resulting collection has piecewise constant density; refining the voxel partition leads to an improved piecewise approximation of the desired (e.g., smooth) density. In practice, the dependence between the number of nodes contained in the unit cube, and average/minimal nearest neighbor distance is tabulated in advance, and then inverted during the construction of the node set.

3. The algorithm

The interested reader will find a Matlab codebase implementing the algorithm described here, as well as the sources for all the figures contained in the present paper, at [41].

3.1. **Formulation.** If the nodes must be restricted to a certain compact set Ω, for example, support of a given indicator, we will refer to the set as *density support*, and to the indicator function as *point inclusion function*. We may further assume that Ω is contained in the d-dimensional unit cube $\mathcal{C} = [0,1]^d$; the case of an arbitrary compact set then follows by choosing a suitable enclosing cube and applying scaling and translation. Suppose the radial density is prescribed by a Lipschitz-1 function; i.e., $|\rho(\boldsymbol{x}) - \rho(\boldsymbol{y})| \le \|\boldsymbol{x} - \boldsymbol{y}\|$. The reason for this assumption is the respective property of $\Delta(\boldsymbol{x})$, and is explained in further detail in the following section. Recall that we use an exponent $s > d$. We summarize the discussion in Section 2 into the following algorithm for generating nodes with radial density ρ:

Initial node layout.

`Step 0` Choose one of the two Q-MC sequences described in Section 2, $\{\mathcal{L}_n : n \ge 1\}$ or $\{\mathcal{M}_n : n \ge 1\}$, draw configurations with up to $n_{\max}$ nodes from it, and determine the average nearest

neighbor distance for its periodization by the integer lattice, denoted by $\bar{\Delta}_n$ for n nodes. Let $\lambda : (0, \infty) \to \{0, 1, 2, \ldots, n_{\max}\}$ be the interpolated inverse to $\bar{\Delta}_n : \{1, 2, \ldots, n_{\max}\} \to (0, 1]$.[1]

`Step 1` Partition $\mathcal{C}$ into M^d equal cube-shaped voxels $\mathcal{V}_m$, $m = 1, \ldots, M^d$ of side length $1/M$, with faces parallel to the coordinate planes. Let $\{\mathcal{V}_m : m \in \mathcal{D}\}$ be the subset for which at least one of the adjacent (i.e., sharing a face) voxels has a vertex inside Ω.

`Step 2` Let $\bar{\rho}_m$ be the average value of ρ at the 2^d vertices of a voxel $\mathcal{V}_m$, $m \in \mathcal{D}$. Place inside $\mathcal{V}_m$ a scaled and translated version of the n_m-point IL (6), or of the n_m-point periodic Riesz minimizer, using n_m defined by

$$n_m = \lambda(\bar{\rho}_m M).$$

Repeat for each $m \in \mathcal{D}$.

Saturation and cleanup.

`Step 3` Let $\mathcal{E} \subset \mathcal{D}$ be the set of m for which voxels $\{\mathcal{V}_m : m \in \mathcal{E}\}$ contain no nodes and the centers $\{\boldsymbol{z}_m : m \in \mathcal{E}\}$ satisfy $\Delta(\boldsymbol{z}_m) > \rho(\boldsymbol{z}_m)$. Sort $\mathcal{E}$ by the increasing values of $\rho(\boldsymbol{z}_m)$. Repeat until $\mathcal{E}$ is empty: for every $m \in \mathcal{E}$ place a node in $\boldsymbol{z}_m$; recompute $\mathcal{E}$.

`Step 4` For all nonempty voxels, remove nodes outside Ω.

Repel-type iterations, boundary detection.

`Step 5` Perform T iterations of the partial gradient descent on the weighted s-energy functional (4) with $\kappa(\boldsymbol{x}, \boldsymbol{x}) = c\rho(\boldsymbol{x})^s$, using the K nearest neighbors of each node: Let the initial configuration be the 0-th iteration, $\boldsymbol{x}_i^{(0)} = \boldsymbol{x}_i$, $i = 1, \ldots, N$, with N denoting the total number of nodes distributed. On the t^{th} iteration, $1 \le t \le T$, given a node $\boldsymbol{x}_i^{(t)}$ with K nearest neighbors $\boldsymbol{x}_{j(i,k)}^{(t)}$, $k = 1, \ldots, K$, form the weighted vector sum

$$\boldsymbol{g}_i^{(t)} = s\,\rho\left(\boldsymbol{x}_i^{(t)}\right)^s \sum_{k=1}^{K} \frac{\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_{j(i,k)}^{(t)}}{\|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_{j(i,k)}^{(t)}\|^{s+2}}, \quad 1 \le i \le N,$$

and the new node position can now be expressed as

$$\boldsymbol{x}_i^{(t+1)} = \begin{cases} \boldsymbol{x}_i^{(t)} + \dfrac{\Delta(\boldsymbol{x}_i^{(t)})}{t + C_2} \dfrac{\boldsymbol{g}_i^{(t)}}{\|\boldsymbol{g}_i^{(t)}\|} & \text{if this sum is inside } \Omega; \\ \boldsymbol{x}_i^{(t)}, & \text{otherwise,} \end{cases} \qquad 1 \le i \le N. \tag{8}$$

where C_2 is a fixed offset chosen to control the step size between $\boldsymbol{x}_i^{(t)}$ and $\boldsymbol{x}_i^{(t+1)}$. If a "pullback" function is provided from a neighborhood of Ω to its boundary, the condition of $\boldsymbol{x}_i^{(t+1)}$ being inside Ω is replaced with applying the pullback; furthermore, if the radial density has an easily computable gradient, or is changing rapidly, an additional term must be included in (8) (see discussion below).

Update the neighbor indices $j(i, k)$ after every few iterations.

`Step 6` If no boundary node set/pullback function is prescribed, define the boundary nodes as follows. Evaluate the point inclusion function for $\boldsymbol{x}_i \pm \Delta(\boldsymbol{x}_i)\boldsymbol{e}_l$, $l = 1, \ldots, d$, $i = 1, \ldots, N$, where $\boldsymbol{e}_l$ is the l-th basis vector. If at least one such point lies outside Ω, the $\boldsymbol{x}_i$ is considered to be a boundary node.

[1] Note that both ILs and the minimizers can have the nearest neighbor distance of at most 1, due to periodicity. We therefore take $\lambda(x) = 0$ whenever $x > 1$.

3.2. **Discussion.** Our assumption of ρ being Lipschitz-1 is natural, since $\Delta(\cdot)$ is always Lipschitz-1, if viewed as a function of position. To see this, consider any two nodes $\boldsymbol{x}, \boldsymbol{y}$, and let $\boldsymbol{x'}, \boldsymbol{y'}$ be their nearest neighbors, respectively, so that $\|\boldsymbol{x} - \boldsymbol{x'}\| = \Delta(\boldsymbol{x})$ and $\|\boldsymbol{y} - \boldsymbol{y'}\| = \Delta(\boldsymbol{y})$. It follows, $\Delta(\boldsymbol{x}) \leq \|\boldsymbol{x} - \boldsymbol{y'}\| \leq \|\boldsymbol{x} - \boldsymbol{y}\| + \Delta(\boldsymbol{y})$, which by symmetry implies Δ is Lipschitz-1.

Initial node layout. In the parts of the density support with constant ρ, the nodes will locally look like a periodization of the initial Q-MC sequence; hence the average neighbor distance in `Step 0` is tabulated for the periodized version. Observe that there is some freedom in the notion of voxel adjacency used to define $\{\mathcal{V}_m : m \in \mathcal{D}\}$ in `Step 1`; for example, in the case of a highly non-convex domain Ω, it might be reasonable to denote the $3^d - 1$ voxels sharing a vertex with a given $\mathcal{V}_m$ as adjacent to it, rather than only the $2d$ voxels that it has a common face with. This would then ensure that no part of Ω will be omitted in the node allocation; imagine a long and thin peninsula in Ω containing no corners of $\mathcal{V}_m$, $m = 1, \ldots, M^d$. We have found however, that the subsequent repel iterations will guarantee that such a peninsula is adequately filled with nodes even when using only the face-adjacent voxels.

If the IL sequence is chosen in `Step 0`, the n_m-node set placed in voxel $\mathcal{V}_m$ at `Step 2` is an adjusted version of (6) as follows. Let for every $\mathcal{V}_m$ the corner with the smallest absolute value be $\boldsymbol{c}_m$; the points $\boldsymbol{c}_m$ are then vertices of a lattice. Before scaling and translating $\mathcal{L}_{n_m}$, apply a random permutation to the coordinates of each node in it, so that to remove long-range lattice structure from the distribution; we will denote such an operation by σ. Then the IL in voxel $\mathcal{V}_m$ becomes

$$\mathcal{L}'_{n_m} = \boldsymbol{c}_m + \frac{f}{M}\sigma(\mathcal{L}_{n_m}) + \frac{\boldsymbol{h}}{M}, \tag{9}$$

where

$$f = 1 - c_d\,(n_{\max})^{-1/d}, \qquad \boldsymbol{h} = \frac{1-f}{2} \cdot (1, 1, \ldots, 1)^{\mathrm{tr}},$$

with c_d depending only on the dimension. The quantities f and $\boldsymbol{h}$ ensure that the lattice points in $\mathcal{L}_m$ are inset into the voxel by about half the separation distance, avoiding poorly separated points along the voxel interfaces.

When the periodic minimizer sequence is selected in `Step 0`, the inset is defined in a similar way, but the permutation is just an identity, $\sigma \equiv \mathrm{id}$, as the minimizers don't have the lattice structure. Likewise, the scaling factor and translation are

$$f_m = 1 - c_d\,(n_m)^{-1/d}, \qquad \boldsymbol{h}_m = \frac{1-f_m}{2} \cdot (1, 1, \ldots, 1)^{\mathrm{tr}}.$$

The analog of (9) thus takes the form

$$\mathcal{M}'_{n_m} = \boldsymbol{c}_m + \frac{f_m}{M}\mathcal{M}_{n_m} + \frac{\boldsymbol{h}_m}{M}. \tag{10}$$

As one would expect, the average separation for the sequence $\{\mathcal{M}_n\}$ is larger than that of $\{\mathcal{L}_n\}$ for the respective values of n. While the inset for the latter is necessary to account for the node proximity after periodization, for the former it serves to mitigate the effects of interfacing voxels containing different number of nodes. This is further discussed in the Appendix.

Saturation and cleanup. Observe that after `Step 2`, voxels in $\{\mathcal{V}_m : m \in \mathcal{D}\}$ satisfying $\bar{\rho}_m M > 1$ do not contain any nodes. The goal of `Step 3` is therefore to remove any redundant sparsity that may be present whenever the radial density ρ is larger than $1/M$, as in this case the function λ in `Step 2` is set to zero. More careful geometric considerations would lead one to set $\lambda(x) > 0$ when $0 < x < \sqrt{d}/M$, the length of a voxel diagonal, and thus make λ dependent on the dimension; on

the other hand, using the interval $0 < x < 1/M$ as we did appears to suffice due to correction of density in `Step 5`.

Note that in practice, when recomputing $\mathcal{E}$ in `Step 3`, to verify $\Delta(\boldsymbol{z}_{m_0}) > \rho(\boldsymbol{z}_{m_0})$ for a fixed $m_0 \in \mathcal{E}$ it is enough to check $\|\boldsymbol{z}_{m_0} - \boldsymbol{z}_m\| > \rho(\boldsymbol{z}_{m_0})$ for the previously selected $\boldsymbol{z}_m$ with $\rho(\boldsymbol{z}_m) < \rho(\boldsymbol{z}_{m_0})$. Indeed, let $\boldsymbol{z}_{m_0}$ be the center of $\mathcal{V}_{m_0}$. Then, by the definition of λ in `Step 2`, the radial density $\rho(\boldsymbol{z}_{m_0}) = (1 + D)/M$ for some $D > 0$, so the Lipschitz-1 property implies, for any $\boldsymbol{x}$ such that $\rho(\boldsymbol{x}) \le 1$ there holds $\|\boldsymbol{z}_{m_0} - \boldsymbol{x}\| \ge |\rho(\boldsymbol{z}_{m_0}) - \rho(\boldsymbol{x})| \ge D/M$. This ensures that distances from $\boldsymbol{z}_{m_0}$ to the nodes produced on `Step 2` satisfy

$$\|\boldsymbol{z}_{m_0} - \boldsymbol{x}\| \ge \frac{D}{1+D}\rho(\boldsymbol{z}_{m_0}).$$

This shows, when $D \gg 1$, not checking the inequality $\|\boldsymbol{z}_{m_0} - \boldsymbol{x}\| > \rho(\boldsymbol{z}_{m_0})$ leads to at most a bounded factor error. On the other hand, for $D \ll 1$ distances from $\boldsymbol{z}_{m_0}$ to the nodes from `Step 2` are also controlled: it follows from (9)–(10) that for n_m small, nodes in the voxel $\mathcal{V}_m$ have larger inset (depending on c_d). This analysis is certainly not rigorous; however, applying the partial gradient descent in `Step 5`, we are able to ensure that the ratio ρ/Δ is close to 1, as desired.

Observe that in `Step 2` the nodes are only placed in $\mathcal{V}_m$'s for which either of the adjacent voxels has corners inside the density support Ω, so removing nodes outside Ω in `Step 4` does not lead to much overhead. Furthermore, since the density is evaluated at the corners only and not at individual nodes, the total number of evaluations may be significantly reduced, which is especially useful when ρ is computationally expensive. It is essential here that due to the Lipschitz-1 property, ρ is well estimated by its values at the corners $\boldsymbol{c}_m$; specifically, $|\rho(\boldsymbol{x}) - \rho(\boldsymbol{c}_m)| \le \sqrt{d}/2M$ with $\boldsymbol{c}_m$ the nearest voxel corner to $\boldsymbol{x}$.

Repel-type iterations, boundary detection. The equality $\kappa(\boldsymbol{x}, \boldsymbol{x}) = c\rho(\boldsymbol{x})^s$ can be justified by observing that each node $\boldsymbol{x}$ of the target distribution must be contained in a ball of radius $\rho(\boldsymbol{x})$, not containing any of the other nodes, hence, the volumetric density must be inverse proportional to $\rho(\boldsymbol{x})^d$. On the other hand, minimizers of (4) converge to the distribution with volumetric density $\kappa(\boldsymbol{x}, \boldsymbol{x})^{-d/s}$; hence $\kappa(\boldsymbol{x}, \boldsymbol{x})^{d/s} = c\rho(\boldsymbol{x})^d$.

The vector sum in `Step 5` is the partial $\boldsymbol{x}$-gradient of the weighted Riesz s-energy (4) in the sense that a single summand of (4) is $e(\boldsymbol{x}, \boldsymbol{y}) = \kappa(\boldsymbol{x}, \boldsymbol{y})\|\boldsymbol{x} - \boldsymbol{y}\|^{-s}$, and thus its complete $\boldsymbol{x}$-gradient is equal to

$$\nabla_{\boldsymbol{x}}\, e(\boldsymbol{x}, \boldsymbol{y}) = -s\,\kappa(\boldsymbol{x}, \boldsymbol{y})(\boldsymbol{x} - \boldsymbol{y})\|\boldsymbol{x} - \boldsymbol{y}\|^{-s-2} + \nabla_{\boldsymbol{x}}\kappa(\boldsymbol{x}, \boldsymbol{y})\|\boldsymbol{x} - \boldsymbol{y}\|^{-s}.$$

For our purposes, the $\boldsymbol{y}$ here is one of the K nearest nodes to $\boldsymbol{x}$, and, since due to the Q-MC initialization there will be few isolated nodes, and since off-diagonal values of $\kappa(\boldsymbol{x}, \boldsymbol{y})$ do not influence the limiting distribution (for details see [5]), we assume $\kappa(\boldsymbol{x}, \boldsymbol{y}) \approx \kappa(\boldsymbol{x}, \boldsymbol{x})$ to rewrite the previous equation as

$$\nabla_{\boldsymbol{x}}\, e(\boldsymbol{x}, \boldsymbol{y}) = -s\,\rho\,(\boldsymbol{x})^s\,(\boldsymbol{x} - \boldsymbol{y})\|\boldsymbol{x} - \boldsymbol{y}\|^{-s-2} + s\nabla_{\boldsymbol{x}}\rho(\boldsymbol{x})\,\rho(\boldsymbol{x})^{s-1}\|\boldsymbol{x} - \boldsymbol{y}\|^{-s}. \tag{11}$$

As has been pointed out at the beginning of Section 3.2, in order to be meaningful as a radial density, the function ρ must be Lipschitz-1. Then by the Rademacher's theorem, $\nabla_{\boldsymbol{x}}\rho$ exists almost everywhere; this validates the use of it in (11) as well as the approximation $\kappa(\boldsymbol{x}, \boldsymbol{y}) \approx \kappa(\boldsymbol{x}, \boldsymbol{x})$. The ratio of the second term to the first one in (11) is bounded by $\nabla_{\boldsymbol{x}}\rho(\boldsymbol{x})\|\boldsymbol{x} - \boldsymbol{y}\|/\rho(\boldsymbol{x})$ and, provided that the distances from $\boldsymbol{x}$ to its nearest neighbors are close to the value of $\rho(\boldsymbol{x})$, is at most $c\nabla_{\boldsymbol{x}}\rho(\boldsymbol{x})$ for a constant c. This condition is satisfied because the chosen Q-MC sequences have very regular local structure. In practice, the node distance is small on the scale of the support and varies slowly, so the second term will have negligible impact on the direction of the gradient after normalization; besides, precise gradient computation may prove costly. For these reasons we omit the second term

in equation (8). If it is necessary to deal with a fast-changing radial density, a trade-off between the computational costs and the resulting distribution properties must be sought.

It doesn't matter which minimization method is applied to the weighted s-energy, rather the gradient descent is chosen due to its simplicity. Note, the second case in (8), leading to shrinking of the line stepping distance, can be thought of as a simplistic backtracking line search; it turns out to be sufficient for our purposes. Furthermore, applying a more involved line search may significantly degrade performance for complicated or nonsmooth domains.

The number of nearest neighbors K in (4), and the number of iterations T in `Step 5` can be adjusted to achieve a trade-off between execution speed/memory consumption/local separation. In our experiments,[2] even relatively small values of K and T produce good results: we used $K \approx T \approx 30$ for 1.36 million nodes with constant density in Section 4.1, and $K = 30$, $T = 200$ for 0.58 million and 0.36 million nodes with variable densities in Sections 4.2 and 4.3, respectively.

4. Sample applications

4.1. **Atmospheric node distribution using surface data.** We use the geodata [1] from the collection of global relief datasets produced by NOAA (National Oceanic and Atmospheric Administration), which contains a 1 arc-minute resolution model. We generate a sample configuration consisting of 1,356,566 nodes distributed uniformly inside an atmospheric-type shell Ω_{etopo}: the outer boundary of Ω_{etopo} is spherical, the inner one is an interpolation of the relief from ETOPO1 data, exaggerated by a factor of 100. The scale is chosen so that the average Earth radius, assumed to be 6,371,220 meters, has unit length; the radius of the outer boundary is set to 1.1, which corresponds to the height of 6,371 meters above the average radius, given the exaggeration factor.

The ETOPO1 dataset stores relief as a 21 600-by-10 800 array of elevations above the sea level; equivalently, of radial coordinates that correspond to the spherical angles defined by the array's indices. The data points are equispaced on lines of constant azimuth/inclination with angular distance $B = \pi/10\,800$ between them. To determine whether a given node $\boldsymbol{x} = (r_{\boldsymbol{x}}, a_{\boldsymbol{x}}, p_{\boldsymbol{x}})$ belongs to Ω_{etopo}, its radial coordinate $r_{\boldsymbol{x}}$ was compared with a linear interpolation of the values of radii of three ETOPO1 points with the nearest spherical coordinates. For example, assume that such three points have the spherical coordinates (r_j, a_j, p_j), $j = 1, 2, 3$, where $0 \le a \le 2\pi$ and $0 \le p \le \pi$ are the azimuth and polar angle, respectively, and

$$\begin{array}{lll} a_1 = lB, & a_2 = (l+1)B, & a_3 = lB, \qquad 0 \le l \le 21\,599; \\ p_1 = mB, & p_2 = mB, & p_3 = (m+1)B, \quad 0 \le m \le 10\,799. \end{array}$$

Without loss of generality, the inequalities

$$lB \le a_{\boldsymbol{x}} < (l+1)B; \qquad mB \le p_{\boldsymbol{x}} < (m+1)B$$

hold true. The point inclusion function is defined in this case as

$$\chi(\boldsymbol{x}; \Omega_{\text{etopo}}) = \begin{cases} 1, & r_1 + \frac{a_{\boldsymbol{x}} - a_1}{B}(r_2 - r_1) + \frac{p_{\boldsymbol{x}} - p_1}{B}(r_3 - r_1) \;<\; r_{\boldsymbol{x}} \;<\; 1.1; \\ 0, & \text{otherwise,} \end{cases}$$

with 1.1 being the radius of the outer sphere in the chosen scale. In effect, the algorithm for evaluating the $\chi(\cdot\,; \Omega_{\text{etopo}})$ described here coincides with the star-shaped point location algorithm from [35, Section 2.2], applied to the interpolated Earth surface and the outer spherical boundary.

Our node set consists of 1,356,566 nodes with the nearest-neighbor separation close to the constant $\rho(\boldsymbol{x}) = 0.01124$, and our top priority was to ensure the low variance of the radial separation across

[2]The Matlab code we provide performs naive autotuning of K and T, using the total number of nodes to be placed. Although sufficient for demonstration purposes, there is room for improvement.

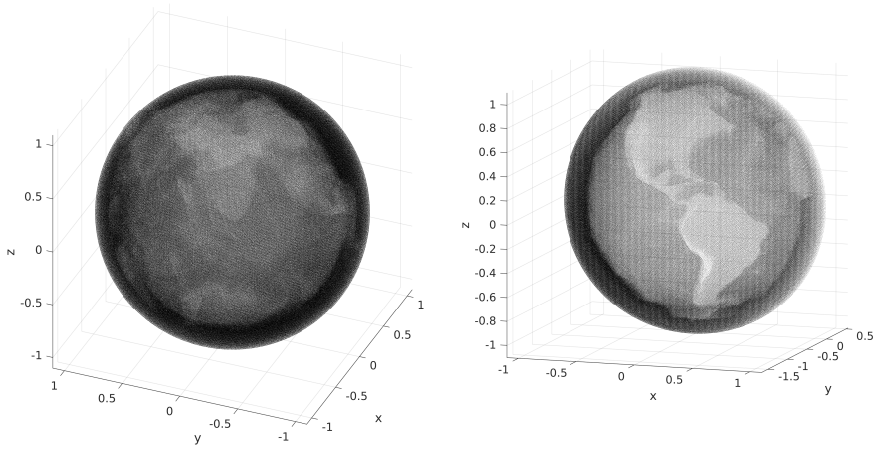

FIGURE 2. Left: a general view of a uniform node distribution in an atmospheric-like shell. Right: a separate view of the Western hemisphere.

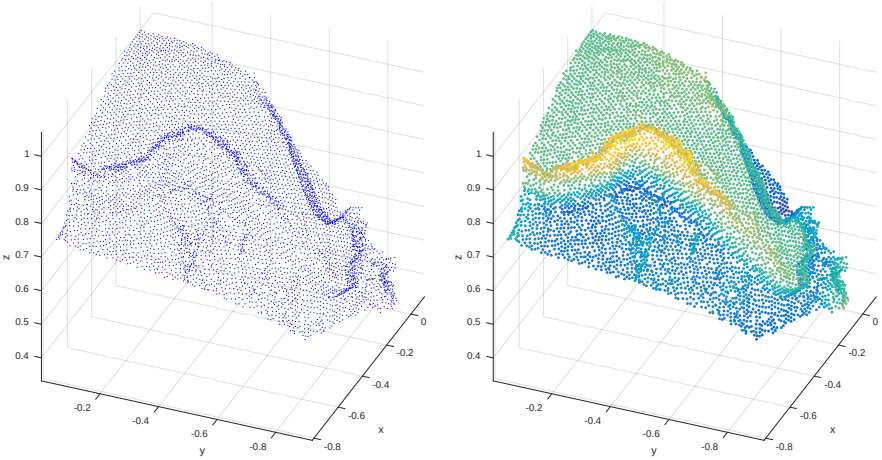

FIGURE 3. Surface subset: a fragment of the Western coast of South America. The nodes on the right are color-coded using heights.

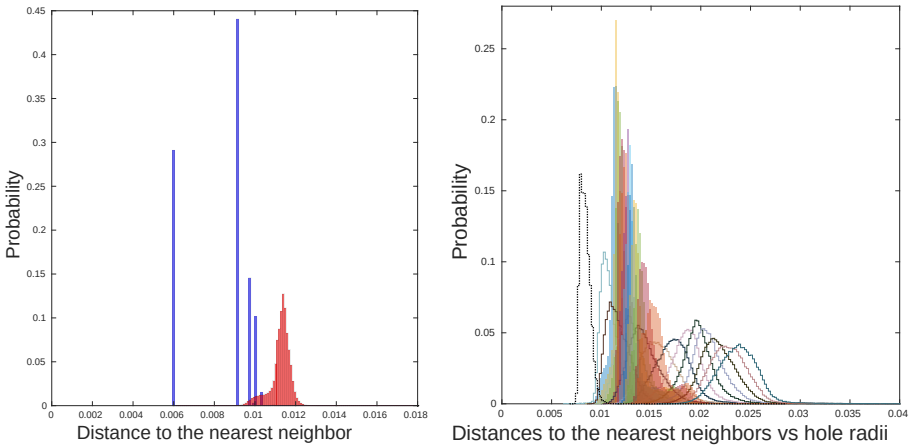

FIGURE 4. The effects of the repel procedure and hole radii. Left: probability distribution of the nearest-neighbor distances in the atmospheric node set, before (blue) and after (red) executing the repel subroutine. Right: distribution of distances to the 12 nearest neighbors for the whole configuration (color only), for the surface subset (contours), the hole radii (black dashed contour on the left).

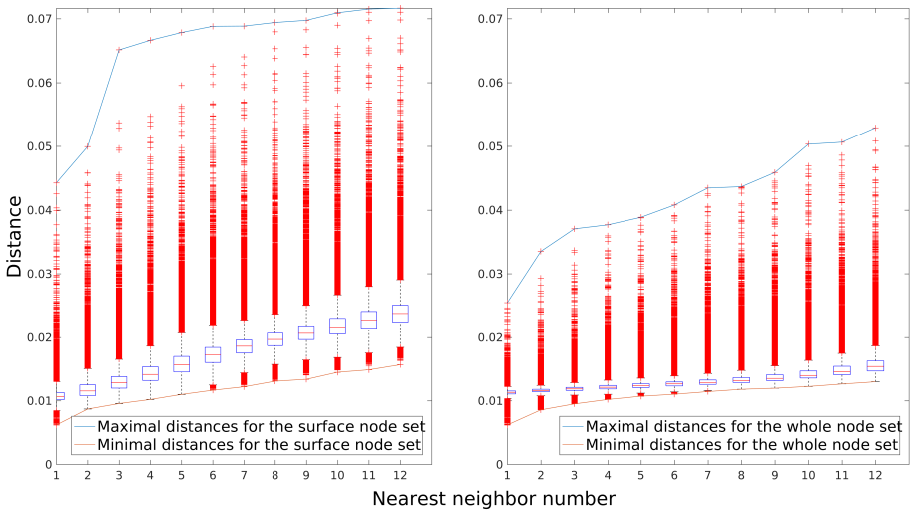

FIGURE 5. Distribution of distances to the 12 nearest neighbors for the atmospheric node configuration; medians and the 25th and 75th percentiles are shown. Left: the surface subset. Right: the whole set. Scales are the same in both subplots.

the configuration, especially on the surface. We used the piecewise IL with golden-ratio derived parameters $\alpha_1 = \sqrt{2}$, $\alpha_2 = (\sqrt{5}-1)/\sqrt{2}$; regarding these α_1, α_2 see also the discussion in Appendix. Several statistics of the resulting set are presented in the following table; here again we used the common notation $\bar{x}$ for the averaged value of a quantity x. Notation Δ^k stands for the distance to the k-th nearest neighbor.

	Whole node set	Surface nodes
$\overline{\Delta^{12}(\boldsymbol{x})/\Delta^2(\boldsymbol{x})}$	1.3674	2.0353
$\overline{\Delta^4(\boldsymbol{x})/\Delta^1(\boldsymbol{x})}$	1.0859	1.34019
99th percentile of $\{\Delta(\boldsymbol{x}_i)\}$	0.012143	0.014444
$\overline{\Delta(\boldsymbol{x})}$	0.011243	0.010879
1st percentile of $\{\Delta(\boldsymbol{x}_i)\}$	0.009652	0.009340

Figure 3 illustrates the distribution of nodes close to the surface of Ω_{etopo}. No pullback function has been used, just the inclusion check performed as in (8). Observe that the near-surface nodes display no artifacts, and the spacing does not significantly depend on the local surface shape. The left subplot in the Figure 4 illustrates the effect of `Step 5` on the distribution of distances to the nearest neighbor. In the right subplot, we have collected distances to the 12 nearest neighbors for the whole configuration, and separately for the surface subset. The histogram also contains the distribution of *hole radii*, that is, distances from the Voronoi centers of the entire node configuration to their respective nearest nodes. It is a well-known fact that the Voronoi centers are local maxima of the distance from the node set [16], considered as a function on the whole space $\mathbb{R}^3$. Note that all the histograms on the right are normalized by probability, not by the node count.

The pair of plots in Figure 5 shows in detail the distribution of distances to the nearest neighbors in the sample node set. It has been produced using the standard Matlab routine *boxplot*. For each of

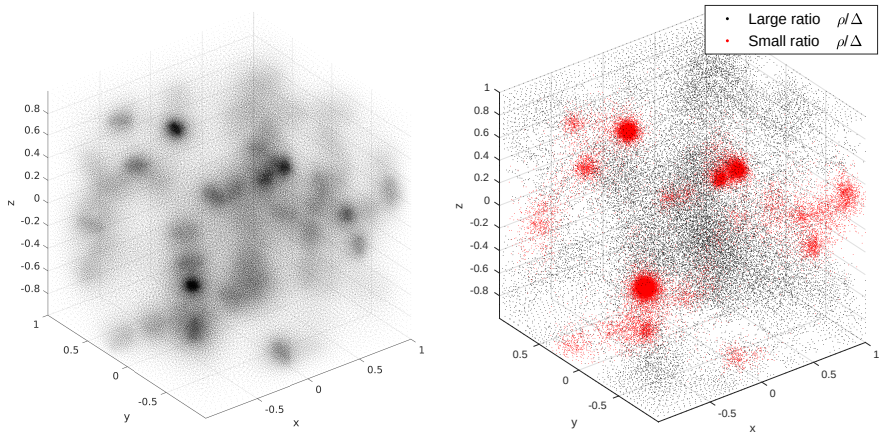

FIGURE 6. Left: the node set from Section 4.2. Right: node locations that contribute to the distribution of the ratio $\rho(\boldsymbol{x})/\Delta(\boldsymbol{x})$ beyond the 5- and 95-percentiles.

the blue boxes corresponding to a specific nearest neighbor, the central mark is the median, the edges of the box denote the 25th and 75th percentiles. The red crosses mark outliers.

4.2. **Point cloud.** To demonstrate a nonuniform node distribution using our algorithm, we fix a collection of 100 points, $\mathcal{P}_{100}$, inside the cube $[-1,1]^3$, and consider the following radial density function:

$$\rho(\boldsymbol{x}) = \left(\Delta(\boldsymbol{x};\mathcal{P}_{100}) + \Delta^2(\boldsymbol{x};\mathcal{P}_{100})\right)/20,$$

where, as above, Δ^k is the distance to the k-th nearest neighbor. A possible interpretation of this density is a distribution that concentrates about a set of points $\mathcal{P}_{100}$, which are of particular interest for a certain model.

We proceed as in the `Step 5` of the algorithm, not using the full gradient expression described in (11). In fact, it is instructive to note that computing the second term in (11) would be quite cumbersome here in view of ρ being a piecewise-defined function. One could thus consider the density recovery for this distribution, Figure 7, as a validation of the gradient truncation approach in `Step 5`; cf. Section 3.2. The Q-MC voxels were drawn from the sequence $\{\mathcal{L}_n\}$ with the same lattice parameters as in Section 4.1, $\alpha_1 = \sqrt{2}$, $\alpha_2 = (\sqrt{5}-1)/\sqrt{2}$.

Figure 7 contains the distribution of the ratio $\rho(\boldsymbol{x})/\Delta(\boldsymbol{x})$. The minimal and maximal values of the ratio are about 0.8099 and 1.8231 respectively; its mean value is 0.9797, and the variance is 0.0019. The 5- and 95-percentiles are 0.9208 and 1.0441, respectively.

4.3. **Spherical shell.** The motivation for this example comes from atmospheric modeling. Representing the Earth surface by a sphere, we consider first a thin 3-dimensional shell Ω_{shell} of inner radius R_{inner} and outer radius $R_{\text{inner}} + H_{\text{atm}}$ with constant target separation h between points in the radial (vertical) direction, and the tangential (horizontal) separation to be $\tau(r) = C \cdot r$ at radius r, for some constant C. With typical choices of parameters, τ will be much larger than h, reflecting the much higher resolution needed in the vertical direction due to $H_{\text{atm}} \ll R_{\text{inner}}$. We make a radial change of variables, which can written in spherical coordinates as $(r\,, a\,, p) \to (\hat{r}(r),\, a,\, p)$, so that any configuration in Ω_{shell} having the 2-directional resolutions $\tau(r)$ and h will have isotropic resolution after the transformation. It is much easier to construct RBF bases in the isotropic case, hence our deliberation.

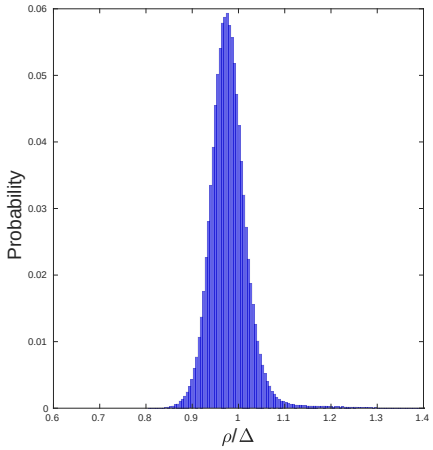

FIGURE 7. Distribution of the ratios $\rho(\boldsymbol{x})/\Delta(\boldsymbol{x})$ for the node set in Section 4.2.

Following this change of variables, the radial/tangential node separations become, respectively,

$$\begin{aligned} \hat{\nu}(r) &= h \cdot \hat{r}'(r) \\ \hat{\tau}(r) &= C \cdot \hat{r}(r). \end{aligned} \tag{12}$$

Setting these two quantities to be equal, we obtain the ODE

$$\hat{r}'(r) = \frac{C}{h} \cdot \hat{r}(r)$$

with initial condition $\hat{r}(R_{\text{inner}}) = 1$, and its solution becomes

$$\hat{r}(r) = \exp\left(C \cdot \frac{r - R_{\text{inner}}}{h} \right).$$

From the second equation in (12) follows that our goal is to generate a node set in the $(\hat{r},\, a,\, p)$-space, whose separation is proportional to $\hat{r}$ and is equal in all directions: $\rho(\boldsymbol{x}) = C \cdot \|\boldsymbol{x}\|$. The outer radius of the image of Ω_{shell} in the $(\hat{r},\, a,\, p)$-space is a function of R_{inner} and H_{atm}; our model implies $R_{\text{inner}} = 6{,}371{,}220$, the mean radius of the Earth in meters, and $H_{\text{atm}} = 12{,}000$, the thickness of the atmospheric layer we are interested in. The constant C is determined by the desired tangential separation at the $r = R_{\text{inner}}$ level.

Say, we intend to generate nodes corresponding to the 2 degree resolution on the spherical "Earth surface" and $h = 400$ meter vertical resolution. Due to the peculiarities of atmospheric modeling, we would like to fix two much denser sets of nodes on the inner and outer boundary of Ω_{shell}; specifically, we are using 12,100 approximate Riesz energy minimizers on a sphere, appropriately rescaled. The interior nodes are generated using our algorithm, and then `Step 5` is modified so that to leave the boundary subset intact. This, however, causes a difficulty: the separation distances between the interior and the surface nodes must remain large; on the other hand, our generic formulation of `Step 5` does not account for the much higher concentration of nodes on the surface, which causes excessive repelling force, seen in the oscillations of the radial distribution, Figure 9. Mitigating this effect requires artificially weakening the repulsive force caused by the boundary nodes, a straightforward task using our codebase. Instead, we show in Figure 9 the performance of the generic algorithm, to illustrate complications that may arise when applying it to specialized problems.

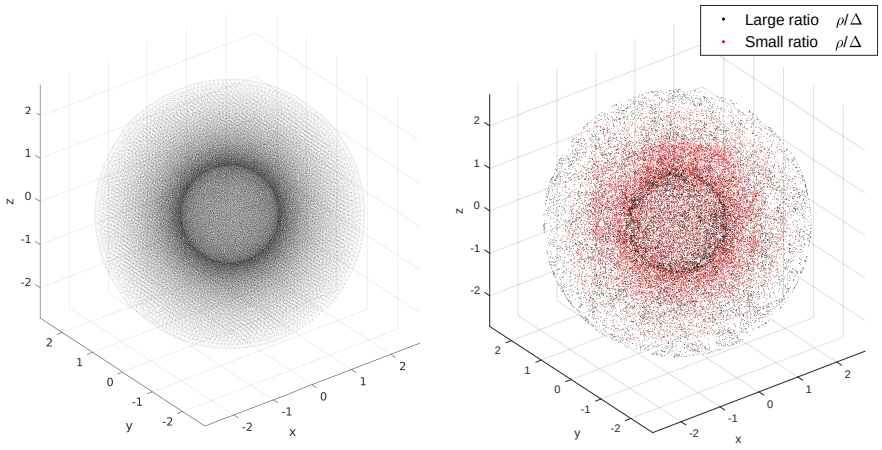

FIGURE 8. Left: the node set from Section 4.3. Right: node locations that contribute to the distribution of the ratio $\rho(\boldsymbol{x})/\Delta(\boldsymbol{x})$ beyond the 5- and 95-percentiles.

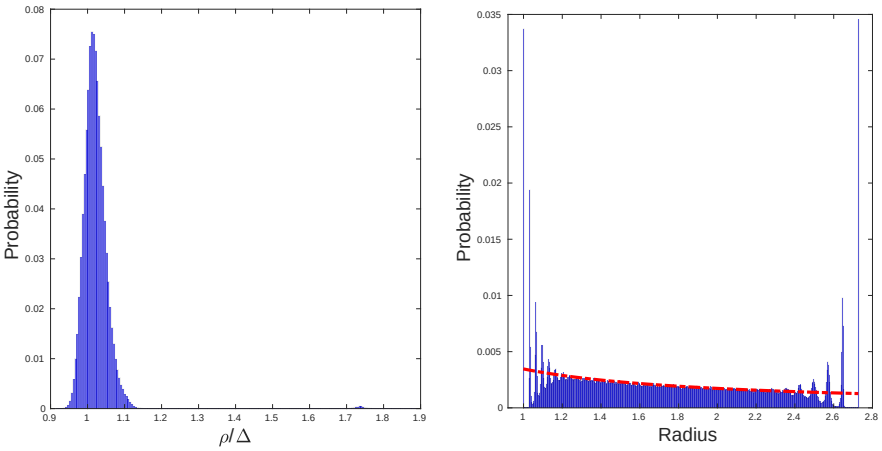

FIGURE 9. Left: Distribution of the ratios $\rho(\boldsymbol{x})/\Delta(\boldsymbol{x})$ for the node set in Section 4.3. Right: Radial node distribution, actual (blue) and the theoretic (red) continuous component; i.e., without the δ-function spikes at the endpoints.

The set Ω_{shell} can be challenging for the basic form of our algorithm, as described in Section 3.1: obtaining satisfying convergence requires using the full version of gradient descent (11). The reasons for it being more difficult to tackle than, say, Ω_{etopo} in Section 4.1, are that due to convexity of the outer boundary, the weighted s-energy minimizers on it are denser than on the sphere with radius $\hat{r}(R_{\text{inner}} + H_{\text{atm}}) - 10^{-3}$, for example; see also discussion at the end of Section 2.2. Getting rid of the artifacts at the endpoints of the radial distribution is done by using the full gradient, weakening the repulsion of the fixed boundary nodes, and *not* striving for the full convergence of a minimization method applied to the Riesz energy.

In this example, we used the $\{\mathcal{M}_n\}$ sequence to fill individual voxels. The left subplot in Figure 9 contains the distribution of the ratio $\rho(\boldsymbol{x})/\Delta(\boldsymbol{x})$. The minimal and maximal values of the ratio are

about 0.9165 and 1.8989 respectively; its mean value is 1.0226, and the variance is 0.0024. The 5- and 95-percentiles are 0.9782 and 1.0717, respectively.

Example	K	T	N	Q-MC distribution times, s	Repel times, s
Atmospheric nodes	33	29	$1,356,566$	5	89
Point cloud	30	200	$577,321$	4	840
Spherical shell	30	200	$358,915$	1	144

TABLE 2. Timings of the examples in Sections 4.1–4.3.

4.4. **Run times.** The execution times (in seconds) for the above examples are summarized in Table 2, where, as before, K and T stand for the number of nearest neighbors and the number of iterations used in the repel procedure in `Step 5`, respectively. The fifth column contains times required to fill the voxels selected at `Step 1` with configurations from either $\{\mathcal{L}_n\}$ or $\{\mathcal{M}_n\}$ and to remove any redundant nodes as in `Step 4`. All the computations were performed on a dedicated machine with 40 GB RAM and an 8-core *Intel Xeon* CPU. Note that the basic Q-MC node sets for both sequences were precomputed, and the pre-computation times are not included in the table. Computation of configurations in $\{\mathcal{L}_n\}$ for $1 \le n \le 200$ took less than 1 second. An implementation of the $\{\mathcal{M}_n\}$ sequence for $1 \le n \le 200$ took 4311 seconds to generate; coordinates of the resulting minimizers as well as the corresponding average separation distances are distributed with [41].

5. Remarks and comparisons to other methods

Our algorithm has proven very efficient for slowly varying radial densities that are small (recall that small radial density means a large number of nodes per unit volume) compared to the entire node set scale, and is capable of handling very complex underlying sets.

The range of dimensions where our method is applicable is determined by the applicability of Q-MC initialization and the nearest neighbor searches: the repelling iterations for Riesz energy in `Step 5` are largely (with a proper value of s) dimension-agnostic. A shortcoming that is common to all quasi-Monte Carlo methods (but of little practical relevance) is a much worse performance (measured by L^2 discrepancy), compared to Monte-Carlo distribution, in dimensions starting at about 15 [12]. Furthermore, using the uniform grid to detect the support Ω, as is done in `Step 1`–`Step 2`, becomes unfeasible already for $d = 10$; instead, one needs an efficient way to determine which corners of the grid are in some sense close to Ω. This is certainly not a feature of our approach, but a manifestation of the curse of dimensionality: treatment of a complicated high-dimensional set is a computationally intensive task. Regarding finding nearest neighbors it should be noted that common implementations of k-d trees are efficient only up to about $d = 20$; additionally, the k-d tree approach is faster than the full brute force search only if $N \gg 2^d$ [26]. On the other hand, as has already been noted, our repelling procedure does not require frequent updates of the search tree, as the local adjacency largely remains intact.

The suggested algorithm is very local, and it therefore must be straightforward to add multi-resolution and adaptive refinement, as is widely done for grids [15, 17], yet as of this writing, our proof-of-concept implementation does not include these features. Still, we would like to observe that refining the voxel structure is indeed easier than refining a mesh, since no geometry is taken into account. This partially addresses the previous remark on detection of Ω in high dimensions.

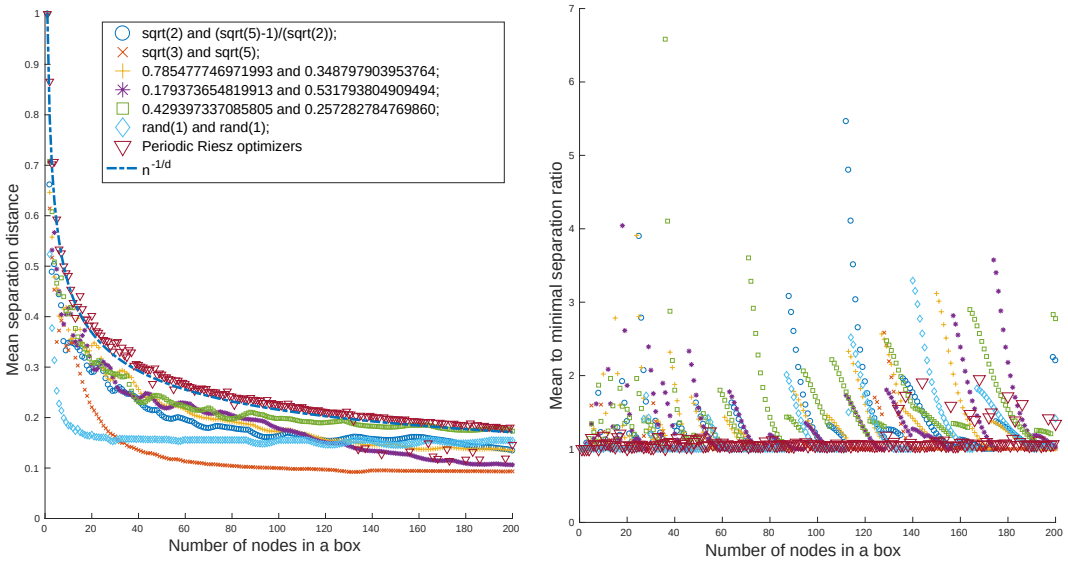

FIGURE 10. Left: dependence of the mean separation distances on the number of nodes in the unit cube for different values of parameters $\alpha_1,\ \alpha_2$; the $n^{-1/d}$ decay rate shown as a dash-dot line. Right: ratios of the mean separation distances to the minimal ones for the same configurations.

The closest set of goals to what we have presented here, that we're aware of, is posed in the pioneering paper [39]; our method is crafted for full-dimensional domains, and apparently performs faster in this case. The bubble packing algorithm in [39] is conceptually similar to the greedy filling of centers in `Step 3`, while physical relaxation is an alternative to the energy minimization we employ; of course, the idea of relaxation can also be found in a number of related references, and is a well-known approach in this context, see for example [34]. Our method requires computing the gradient of the desired radial density in the cases when the outer boundary of the underlying set is uniformly convex, and/or when the radial density changes quickly. Alternatively, fine partition of the set is necessary. Either solution, however, may be computationally expensive.

Of the two Q-MC sequences we considered, the periodic Riesz minimizers appear more promising, being devoid of the lattice structure and having high space utilization. On the other hand, we have successfully applied ILs as an elementary uniform configuration. One could use different sets of irrational parameters $\alpha_1, \dots, \alpha_{d-1}$ for different numbers of nodes in a voxel. Although this might be useful in mitigating the non-isotropic behavior of ILs, it makes hard to control node separation at voxel interfaces.

6. Appendix: separation properties of sequences $\{\mathcal{L}_n\}$ and $\{\mathcal{M}_n\}$

This Appendix deals with the results of our numerical experiments, set in the 3-dimensional space. The function $\lambda(r)$ used in `Step 2` is the number of nodes in the unit cube $[0,1]^3$, placed according to (6), or obtained by minimizing the Riesz s-energy (3) with periodic metric, such that the mean separation distance of these nodes is the closest to r. To compute $\lambda(r)$ for the periodization of $\{\mathcal{L}_n\}$, we tabulate mean separations $\bar{\Delta}_n$ in a sample configuration comprising $\mathcal{L}_n$ and its $26 = 3^3 - 1$ copies, obtained translating $\mathcal{L}_n$ by the vectors $\big\{(i,j,k)^{\mathrm{tr}} : i,j,k \in \{0,\pm 1\}$ and $|i| + |j| + |k| > 0\big\}$. The tabulated dependence of separation on n is then inverted and interpolated using a piecewise cubic

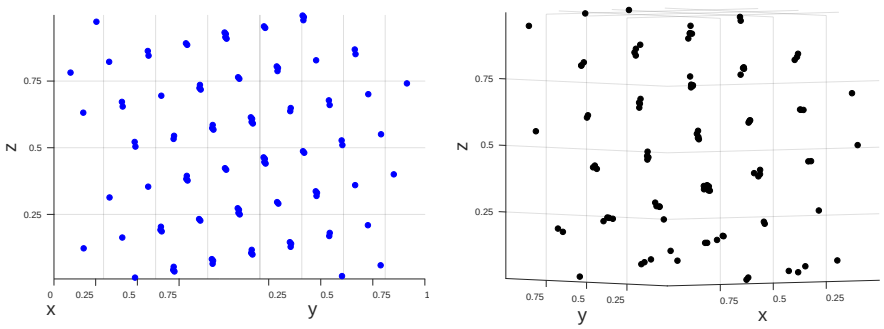

FIGURE 11. Left: A cross-section of the IL $\mathcal{L}_{100}$ with parameters 0.179373654819913 and 0.531793804909494. Right: A (different) cross-section of $\mathcal{M}_{100}$.

Hermite interpolating polynomial. The reason to consider separation distance between configurations in 3^d cubes in dimension d (and not a single cube with a single instance of $\mathcal{L}_n$) is to account for the boundary effects. Likewise, to compute $\lambda(r)$ for the Riesz minimizers, the mean separation of $\mathcal{M}_n$ is tabulated for $1 \le n \le n_{\max}$, then the inverse dependence is interpolated. No copies of $\mathcal{M}_n$ are considered alongside the original configuration, since periodicity condition is already included in the metric (7).

In general, putting too many nodes in individual voxels is justified only if the radial density function ρ varies slowly. For our applications, $n_{\max} \le 100$ was sufficient. The left plot in Figure 10 illustrates the delicate dependence of the separation distances of ILs on the lattice parameters. While any set of irrational quantities $\alpha_1, \dots, \alpha_{d-1}$ in (6) that are linearly independent over rationals will give a uniformly distributed IL as n grows, certain values may perform better than the others. In particular, adjustments can be made to improve the distribution for small values of n. For example, it is known from [2] that a 2-dimensional IL generated by the golden ratio has optimal L^2 discrepancy. Numerical experiments have shown that its 3-dimensional analog with parameters $\alpha_1 = \sqrt{2}$, $\alpha_2 = (\sqrt{5}-1)/\sqrt{2}$ does perform well for large numbers of nodes; yet by carrying out a Monte Carlo search for the parameters maximizing separation distance in (6), we found several (necessarily rational) pairs that performed at least just as well for up to $n = 200$, see Figure 10.

Curiously enough, a pair of random numbers drawn uniformly from $[0, 1]$ (shown in the legend as rand(1)), consistently performed better than the pair $\sqrt{3}$ and $\sqrt{5}$, starting at $n \approx 40$. We were able to reproduce this behavior in a number of runs; in fact, we haven't seen a random pair that wouldn't always outperform $\sqrt{3}$ and $\sqrt{5}$ after a fairly small n.

The second graph in Figure 10 shows the ratios of the mean to minimal separation distances $\bar{\Delta}_n/\Delta_n$ for the same range of n. In both subfigures, Riesz periodic minimizers clearly stand out, by having the largest mean separation (left), and by smallest ratios (right). This means, the nearest neighbor distances $\Delta(\boldsymbol{x})$ vary little from node to node in the $\{\mathcal{M}_n\}$ sequence. We conclude this section by presenting a pair of cross-sections of the IL $\mathcal{L}_{100}$ and the configuration $\mathcal{M}_{100}$ that look remarkably similar. In fact, we found the vague resemblance between the low-energy periodic configurations and lattice structures, similar to ILs, quite interesting, given the connection between packing and Riesz energy minimization [28], and that the highest packing density in the 3-dimensional space is achieved, in particular, by the hcp lattice [16].

Acknowledgements

We thank Edward Saff and Douglas Hardin for their interest and useful comments. O.V. and T.M. were supported, in part, by the NSF grant DMS-1516400; they would also like to express their gratitude to the University of Colorado, Boulder, where a part of this work was completed.

References

[1] Amante, C., and Eakins, B. W. ETOPO1 1 Arc-Minute Global Relief Model: Procedures, Data Sources and Analysis. `https://doi.org/10.7289/V5C8276M`, 2009.

[2] Bilyk, D. The L^2 discrepancy of irrational lattices. In *Monte Carlo and Quasi-Monte Carlo Methods 2012*. Springer, 2013, pp. 289–296.

[3] Bilyk, D., Temlyakov, V. N., and Yu, R. The L^2 Discrepancy of Two-Dimensional Lattices. In *Springer Proc. Math. Stat.*, vol. 25. 2012, pp. 63–77.

[4] Borodachov, S. V., Hardin, D. P., and Saff, E. B. Asymptotics for discrete weighted minimal Riesz energy problems on rectifiable sets. *Trans. Amer. Math. Soc. 360*, 3 (2008), 1559–1580.

[5] Borodachov, S. V., Hardin, D. P., and Saff, E. B. Low Complexity Methods For Discretizing Manifolds Via Riesz Energy Minimization. *Found. Comput. Math. 14*, 6 (2014), 1173–1208.

[6] Brauchart, J. S., Hardin, D. P., and Saff, E. B. The next-order term for optimal Riesz and logarithmic energy asymptotics on the sphere. *Contemp. Math 578* (2012), 1–31.

[7] Brauchart, J. S., Reznikov, A. B., Saff, E. B., Sloan, I. H., Wang, Y. G., and Womersley, R. S. Random point sets on the sphere—hole radii, covering, and separation. *Experimental Mathematics* (2016), 1–20.

[8] Broomhead, D. S. and Lowe, D. Multivariable Functional Interpolation and Adaptive Networks. *Complex Syst. 2* (1988), 321– 355.

[9] Buhmann, M. D. New Developments in the Theory of Radial Basis Function Interpolation. In *Multivar. Approx. From CAGD to Wavelets* (nov 1993), vol. 2, WORLD SCIENTIFIC, pp. 35–75.

[10] Buhmann, M. D. *Radial Basis Functions*. Cambridge University Press, Cambridge, 2003.

[11] Buhmann, M. D., Dinew, S., and Larsson, E. A note on radial basis function interpolant limits. *IMA J. Numer. Anal. 30*, 2 (2010), 543–554.

[12] Caflisch, R. E. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica 7* (1998), 1.

[13] Chang, C., and Lin, C. LIBSVM. *ACM Trans. Intell. Syst. Technol. 2*, 3 (2011), 1–27.

[14] Cheney, W., and Light, W. *A course in approximation theory*, vol. 101 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2009. Reprint of the 2000 original.

[15] Clawpack Development Team. Clawpack software, 2017. Version 5.4.0.

[16] Conway, J. H., and Sloane, N. J. A. *Sphere packings, lattices and groups*, third ed., vol. 290 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, New York, 1999. With additional contributions by E. Bannai, R. E. Borcherds, J. Leech, S. P. Norton, A. M. Odlyzko, R. A. Parker, L. Queen and B. B. Venkov.

[17] Debreu, L., Vouland, C., and Blayo, E. AGRIF: Adaptive grid refinement in fortran. *Computers & Geosciences 34*, 1 (2008), 8–13.

[18] Fasshauer, G. E. *Meshfree approximation methods with MATLAB*, vol. 6 of *Interdisciplinary Mathematical Sciences*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007. With 1 CD-ROM (Windows, Macintosh and UNIX).

[19] Fornberg, B. *A practical guide to pseudospectral methods*. Cambridge University Press, 1996.

[20] Fornberg, B. Calculation of Weights in Finite Difference Formulas. *SIAM Rev. 40*, 3 (1998), 685–691.

[21] Fornberg, B., and Flyer, N. *A Primer on Radial Basis Functions with Applications to the Geosciences*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015.

[22] Fornberg, B., and Flyer, N. Fast generation of 2-D node distributions for mesh-free PDE discretizations. *Comput. Math. with Appl. 69*, 7 (2015), 531–544.

[23] Fornberg, B., and Flyer, N. Solving PDEs with radial basis functions. *Acta Numerica 24* (2015), 215–258.

[24] Fornberg, B., Lehto, E., and Powell, C. Stable calculation of Gaussian-based RBF-FD stencils. *Comput. Math. with Appl. 65*, 4 (feb 2013), 627–637.

[25] Fornberg, B., Wright, G., and Larsson, E. Some Observations Regarding Interpolants in the Limit of Flat Radial Basis Functions. *Comput. Math. with Appl. 47*, 1 (2004), 37–55.

[26] Freidman, J. H., Bentley, J. L., and Finkel, R. A. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Trans. Math. Softw. 3*, 3 (1977), 209–226.

[27] Frey, P. J., and George, P.-L. *Quadtree-octree Based Methods*. ISTE, 2010, pp. 163–199.
[28] Hardin, D. P., and Saff, E. B. Minimal Riesz Energy Point Configurations for Rectifiable d-Dimensional Manifolds. *Adv. Math. 193*, 1 (2003), 174–204.
[29] Kuipers, L., and Niederreiter, H. *Uniform distribution of sequences*. Dover Publications, Mineola, N.Y, 2006.
[30] Landkof, N. S. *Foundations of modern potential theory*, vol. 180. Springer, 1972.
[31] Lemieux, C. *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer Series in Statistics. Springer New York, New York, NY, 2009.
[32] Link, W. A., and Eaton, M. J. On thinning of chains in MCMC. *Methods Ecol. Evol. 3*, 1 (feb 2012), 112–115.
[33] Micchelli, C. A. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constr. Approx. 2*, 1 (1986), 11–22.
[34] Persson, P.-O., and Strang, G. A Simple Mesh Generator in MATLAB. *SIAM Rev. 46*, 2 (jan 2004), 329–345.
[35] Preparata, F. P., and Shamos, M. I. *Computational Geometry*. Springer New York, 1985.
[36] Procopiuc, O., Agarwal, P. K., Arge, L., and Vitter, J. S. Bkd-tree: A dynamic scalable kd-tree. In *Advances in Spatial and Temporal Databases*. Springer Berlin Heidelberg, 2003, pp. 46–65.
[37] Schoenberg, I. J. Metric Spaces and Completely Monotone Functions. *Ann. Math. 39*, 4 (1938), 811.
[38] Shankar, V., Wright, G. B., Kirby, R. M., and Fogelson, A. L. A radial basis function (RBF)-finite difference (FD) method for diffusion and reaction–diffusion equations on surfaces. *Journal of Scientific Computing 63*, 3 (2014), 745–768.
[39] Shimada, K., and Gossard, D. C. Bubble mesh. In *Proc. third ACM Symp. Solid Model. Appl. - SMA '95* (New York, New York, USA, 1995), ACM Press, pp. 409–419.
[40] Varma, U. M., Rao, S. V. R., and Deshpande, S. M. Point distribution generation using hierarchical data structures. In *Proc. ECCOMAS 2004* (Jyväskylä, 2004).
[41] Vlasiuk, O., and Michaels, T. Boxed lattices and Riesz minimizers for RBF computations. `https://github.com/OVlasiuk/3dRBFnodes.git`, 2017. [Online; accessed 12-October-2017].
[42] Wendland, H. *Scattered Data Approximation*. Cambridge University Press, 2004.
[43] Zhou, K., Hou, Q., Wang, R., and Guo, B. Real-time KD-tree construction on graphics hardware. *ACM Trans. Graph. 27*, 5 (dec 2008), 1.

Center for Constructive Approximation, Department of Mathematics, Vanderbilt University, Nashville, TN 37240, USA
Email address: `oleksandr.vlasiuk@vanderbilt.edu`
Email address: `timothy.j.michaels@vanderbilt.edu`

Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research, Boulder, CO 80305, USA
Email address: `flyer@ucar.edu`

Department of Applied Mathematics, University of Colorado, Boulder, CO 80309, USA
Email address: `bengt.fornberg@colorado.edu`