ELSEVIER

Comput. Methods Appl. Mech. Engrg. 191 (2002) 1349–1366

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Meshfree, probabilistic determination of point sets and support regions for meshless computing

Qiang Du, Max Gunzburger *, Lili Ju

Department of Mathematics, Iowa State University, 400 Carver Hall, Ames, IA 50011-2064, USA

Received 28 November 2000; received in revised form 5 July 2001

Abstract

New algorithms are presented for the determination of point sets and associated support regions that can then be used in meshless computing methods. The algorithms are probabilistic in nature so that they are totally meshfree, i.e., they do not require, at any stage, the use of any coarse or fine boundary conforming or superimposed meshes. Computational examples are provided that show, for both uniform and non-uniform point distributions, that the algorithms result in high-quality point sets and high-quality support regions. Furthermore, the algorithms lend themselves well to parallelization. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Centroidal Voronoi tessellations; Meshless computing; Point placement; Sphere coverings; Monte Carlo methods

1. Introduction

It is well known that mesh generation is a very difficult endeavor for problems with complicated geometries. As a result, there is growing interest in seeking alternative approaches for numerical simulations which involve no mesh at all. These approaches are called *meshless* or *meshfree* or *gridless* methods. Of course, gridless methods are not a new notion; for example, Monte Carlo and particle methods and even polynomial approximation methods for numerical integration, function approximation, or for approximating the solutions of differential equations are meshless-type methods that have been around for decades. The newer meshless methods that are of interest here take a different approach. The main ideas are to generate a set of points in a given domain; attach a patch to each point such that the the union of the patches covers the domain; construct local shape functions having support over the patches; and then, using some data fitting method, set up an algebraic system of equations. Several realizations of these approaches using different data fitting methods have been proposed, including partition of unity methods (PUM) [2], element-free Galerkin methods [5], hp-clouds [10], and reproducing kernel methods [12].

The generation of a point set and the associated patches is always the first step in efficient meshless methods. Many previous algorithms for these steps are complicated since they involve the point set of a mesh, i.e., they are not truly meshless. There exist several algorithms for point set generation which avoid this flaw. In [6], an approach is proposed that uses Voronoi diagrams and weighted bubble packing. Another good simplicial algorithm for the generation of point sets and the associated patches is the bitting method introduced in [13] which uses an advancing front strategy for generating the spherical patches. In [11], a meshless discretization technique is suggested for non-stationary convection–diffusion problems where the point set is formed via a Halton sequence instead of by sampling uniformly distributed points.

* Corresponding author. Tel.: +1-515-294-1752; fax: +1-515-295-5454.
E-mail addresses: qdu@iastate.edu (Q. Du), gunzburg@iastate.edu (M. Gunzburger), jllwyb@iastate.edu (L. Ju).

PII: S0045-7825(01)00327-9

However, all the above methods do not lend themselves well to adaptive techniques so that the implementation of local refinement or coarsening strategies of point sets based on these methods is not a simple matter. For example, in the method of [11], coarser approximations in specific regions are effected just by reducing the number of points used; since no natural multilevel Halton sequence can be given, this leads to a multilevel approach in which non-nested spaces arise. Another example is provided by moving least square methods (MLSM) [4] which are used to build shape functions on the domain of interest and result in the need for multiple coverings of the domain so that the bitting method cannot be used. Still another example is provided by Monte Carlo methods associated with some density function; these are natural approaches for adaptive techniques, but the results are not good in general because the points placement is not sufficiently regular and it also requires a very good random number generator.

In this paper, we present probabilistic (and totally messfree) algorithms for point placement and support sphere determination in any domain; the algorithms are very well suited for adaptation and parallelization and produce very high-quality point sets. In Sections 2 and 3, we respectively, discuss point placements using centroidal Voronoi tessellations (CVT) and the open covering of the domain using spheres. (Spherical coverings are considered only for ease of exposition; our methodologies apply just as well to other support region geometries.) In Section 4, some numerical experiments are performed to compare our methodologies with the Monte Carlo method for uniform and non-uniform point distributions; in addition, for uniform point distributions, we compare with point sets generated via Halton sequences.

2. Point placement using centroidal Voronoi tessellations

Let $\|\cdot\|$ denote the Euclidean norm in $\mathbb{R}^N$. Given a open domain $\Omega \subset \mathbb{R}^N$ and a set of regions $\{V_i\}_{i=1}^n$, if

$$V_i \cap V_j = \emptyset \quad \text{for } i \neq j \quad \text{and} \quad \bigcup_{i=1}^{n} \overline{V}_i = \overline{\Omega},$$

then we call $\{V_i\}_{i=1}^n$ a *tessellation* of Ω. Furthermore, given a set of points $\{\mathbf{x}_i\}_{i=1}^n$ in Ω, if $\{V_i\}_{i=1}^n$ is defined by

$$V_i = \{\mathbf{y} \in \Omega \,|\, \|\mathbf{y} - \mathbf{x}_i\| < \|\mathbf{y} - \mathbf{x}_j\| \text{ for } j = 1, \ldots, n,\ j \neq i\}, \tag{1}$$

then the set $\{V_i\}_{i=1}^n$ is called a *Voronoi tessellation* or *Voronoi diagram* of Ω, the points $\{\mathbf{x}_i\}_{i=1}^n$ are called *generating points* or *generators*, and each V_i is referred to the *Voronoi region* associated with $\mathbf{x}_i$. It is well known that the Voronoi regions are polyhedra.

Suppose now a density function $\rho(\mathbf{x}) \geqslant 0$ defined on $\overline{\Omega}$ is given; then, for each Voronoi region V_i, the *mass centroid* $\widetilde{\mathbf{x}}_i$ of V_i is defined by

$$\widetilde{\mathbf{x}}_i = \frac{\int_{V_i} \mathbf{y}\rho(\mathbf{y})\,\mathrm{d}V}{\int_{V_i} \rho(\mathbf{y})\,\mathrm{d}V} \quad \text{for } i = 1, \ldots, n. \tag{2}$$

We refer to a Voronoi tessellation $\{V_i\}_{i=1}^n$ as a centroidal Voronoi tessellation (CVT) of Ω if and only if

$$\mathbf{x}_i = \widetilde{\mathbf{x}}_i \quad \text{for } i = 1, \ldots, n, \tag{3}$$

i.e., the points $\{\mathbf{x}_i\}_{i=1}^n$ which serve as the generators associated with the Voronoi regions $\{V_i\}_{i=1}^n$ are the mass centroids of those regions. CVTs are very useful in a variety of applications; see [7,16] for detailed discussions.

Note that the situation (3) is special in the sense that for an arbitrary set of points $\{\mathbf{x}_i\}_{i=1}^n$ in some domain Ω with some density function $\rho(\mathbf{x})$, these points generally are not the mass centroids of the corresponding Voronoi regions [7]. Thus, we are faced with the construction problem: given a positive integer n, a domain Ω, and a density function $\rho(\mathbf{x})$ defined on $\overline{\Omega}$, find a CVT of Ω. It is worth noting that the solution of the this problem is in general not unique [7]. Several good algorithms for determining CVTs for both discrete and continuous sets Ω have been proposed, including MacQueen's method [15] which is a random sampling algorithm and Lloyd's method [14] which is the obvious (deterministic) iteration between finding centroids and Voronoi tessellations. However, MacQueen's method converges very slowly and even fails to give the

optimal centroidal Voronoi regions in some situations. Lloyd's method can generally obtain good results but requires accurate calculations of Voronoi diagrams and volumes of Voronoi regions; these are difficult operations for complicated geometries in three dimensions and, in any case, are relatively expensive.

Another probabilistic algorithm, suggested in [9], stems from a combination of the MacQueen and Lloyd methods.

Algorithm 1. Given a region Ω, a density function $\rho(\mathbf{x})$ defined for all $\mathbf{x} \in \overline{\Omega}$, and a positive integer n;

0. choose a positive integer q and positive constants $\{\alpha_i, \beta_i\}_{i=1}^2$ such that $\alpha_1 + \alpha_2 = 1$ and $\beta_1 + \beta_2 = 1$; choose an initial set of n points $\{\mathbf{x}_i\}_{i=1}^n$, e.g., by using a Monte Carlo method; set $j_i = 1$ for $i = 1, \ldots, n$;
1. choose q points $\{\mathbf{y}_r\}_{r=1}^q$ in Ω at random, e.g., by a Monte Carlo method, according to the probability density function $\rho(\mathbf{x})$;
2. for $r = 1, \ldots, q$, determine a $\mathbf{x}_{i_r^*}$ among $\{\mathbf{x}_i\}_{i=1}^n$ that is closest to $\mathbf{y}_r$;
3. for $i = 1, \ldots, n$, gather together in the set W_i all sampling points $\mathbf{y}_r$ closest to $\mathbf{x}_i$, i.e., in the Voronoi region of $\mathbf{x}_i$; if the set W_i is empty, do nothing; otherwise, compute the average $\mathbf{x}_i^*$ of the set W_i and set

$$\mathbf{x}_i^* \leftarrow \frac{(\alpha_1 j_i + \beta_1)\mathbf{x}_i + (\alpha_2 j_i + \beta_2)\mathbf{x}_i^*}{j_i + 1} \quad \text{and} \quad j_i \leftarrow j_i + 1;$$

   the new set of $\{\mathbf{x}_i^*\}$, along with the unchanged $\{\mathbf{x}_j\}$, $j \neq i$, form the new set of points $\{\mathbf{x}_i\}_{i=1}^n$;
4. if the new points meet some convergence criterion, terminate; otherwise, return to step 1.

The Monte Carlo method, i.e., random sampling according to a given density function, plays a key role in this algorithm; we use the rejection method [1] for random point generation; it is much faster than the classical Monte Carlo procedure using numerical integrations and is also much better for dealing with complicated regions [9]. There also are a number of algorithms for efficiently determining closest neighbors which use some data structures for Algorithm 1; see [1,3,17]. The complexity of Algorithm 1 is $O(q \log(n))$ per iteration. The results of numerical experiments [9] also showed that Algorithm 1 can produce almost the same good results as does the deterministic (and certainly not meshless) Lloyd's method but without requiring the calculation of Voronoi regions. Note that in the context of probabilistic algorithms such as Monte Carlo methods and Algorithm 1, uniform point distributions mean that the density function is constant.

We will take the generators $\{\mathbf{x}_i\}_{i=1}^n$ of CVTs as point sets for meshless discretizations. In fact, they can be regarded as improvements on point sets generated via Monte Carlo, e.g., random sampling, methods.

In practical applications, many meshless methods require the imposition of certain constraints on the positions of the points; for example, it may be required that some of the points lie on the geometrical boundary (or more generally, on surfaces or lines in three dimensions) so that boundary conditions can be handled. To satisfy such constraints on a point set, a constrained CVT algorithm is required. There are different approaches for constructing constrained CVTs; see [8]. One approach is to predetermine a subset of points along the boundary via a lower-dimensional CVT construction based on another preselected density function defined on the boundary. A second approach is to construct the CVT using a standard algorithm with the modification that at each iteration, the set of generators whose corresponding Voronoi regions contain a portion of the boundary of the domain are projected onto the geometrical boundary. A drawback of the first approach is that, in practice, it may not be a simple matter to determine a priori the number of generating points on the boundary and the corresponding boundary density function. The second approach, on the other hand, is totally automatic and is free from the need to predetermine

[1] The rejection method for generating random points in a rectangle is the following algorithm. Given Ω defined by $a < x < b$ and $c < y < d$, and given the density function $\rho(x, y)$ defined on $\overline{\Omega}$, set $\widehat{\rho} = \max_{(x,y)\in\overline{\Omega}} \rho(x, y)$. Then, a random point $(x, y) \in \overline{\Omega}$ is determined as follows. First, sample a random point X' with constant density in [0,1] and set $X = a + (b - a)X'$. Then, sample a random point Y' with constant density in [0,1] and set $Y = c + (d - c)Y'$. Next, sample a random point U with constant density in [0,1]. If $U < \rho(X, Y)/\widehat{\rho}$, set $(x, y) = (X, Y)$; otherwise, start again. The rejection of points in non-rectangular regions is discussed below.

the number of boundary points or boundary densities. In [8], a variational formulation is also proposed which covers both approaches mentioned above.

For the numerical experiments of this paper, the second approach is taken since it is a simple matter to probabilistically determine the set of generators whose corresponding Voronoi regions contain a portion of the boundary of the domain. For example, one could a priori sample a set of boundary points; then, for each point on the boundary point set, one searches for the closest generators; then one knows that the corresponding Voronoi region will contain a portion of the boundary of the domain.

3. Sphere supports and overlaps

Suppose that a collection of n points $\{\mathbf{x}_i\}_{i=1}^n$ has been chosen in the domain $\overline{\Omega}$ according to a given density function $\rho(\mathbf{x})$ using the generators of the CVT of $\overline{\Omega}$. Then, to each point $\mathbf{x}_i$, a spherical patch

$$S_i = \{\mathbf{y} \in \mathbb{R}^N \,|\, \|\mathbf{y} - \mathbf{x}_i\| \leqslant h_i\} \tag{4}$$

is attached, where $h_i \in \mathbb{R}^+$ is the radius of the sphere S_i. In our methodology, the patches can also be easily chosen to have other shapes. e.g., ellipsoidal, rectangular, or brick type objects; for simplicity of exposition, here we restrict ourselves to spherical patches. The construction of the patches S_i, e.g., the covering of the domain $\overline{\Omega}$ by spherical regions, is a very important step in the discretization process because, e.g., the S_is define the supports of the trial and test functions of a Galerkin method. The S_is have to satisfy the covering condition $\overline{\Omega} \subset \bigcup_{i=1}^k S_i$. Furthermore, the intersection of $S_i \bigcap S_j$ of two patches should be sufficiently large if it is not empty, e.g., if the intersection is too small, it may be mistakenly identified to be zero when using finite precision arithmetic. An algorithm for determining the size of these patches was proposed in [11]. Although the algorithm there applies to square patches, it can easily be changed for use for spherical patches. That algorithm involves a coarse mesh which often has to be refined locally; this is not so simple when the points $\{\mathbf{x}_i\}_{i=1}^n$ are non-uniformly distributed. Thus, we define a new algorithm to avoid these flaws and which take advantage of properties of CVTs such as the fact each point has an associated Voronoi region which is generally a polyhedron.

We use the algorithm given in [11] as a basis for an algorithm which is totally meshfree.

Algorithm 2. Given a region Ω, a density function $\rho(\mathbf{x})$ defined for all $\mathbf{x} \in \overline{\Omega}$, and a set of points $\{\mathbf{x}_i\}_{i=1}^n$ in Ω;

0. choose positive integers m_1 and m_2 and a constant $\gamma > 1$ and set $h_i = 0$ for all $i = 1, 2, \ldots, n$;
1. select a set of m_1 points $\{\xi_i\}_{i=1}^{m_1}$ uniformly distributed over Ω by, e.g., a Monte Carlo method; select another set of m_2 points $\{\eta_j\}_{j=1}^{m_2}$ distributed over Ω according to the density function $\rho(x)$, e.g., again by a Monte Carlo method; let $\mathscr{P} = \{\xi_i\}_{i=1}^{m_1} \cup \{\eta_j\}_{j=1}^{m_2}$;
2. for all $\mathbf{y} \in \mathscr{P}$:
   - find a $\mathbf{x}_{i^*}$ in $\{\mathbf{x}_i\}_{i=1}^n$ which is closest to $\mathbf{y}$;
   - compute the distance $d_{i^*} = \|\mathbf{y} - \mathbf{x}_{i^*}\|$;
   - if $h_{i^*} < d_{i^*}$, then set $h_{i^*} = d_{i^*}$;
3. set $h_i = \gamma h_i$ for all $i = 1, 2, \ldots, n$.

The points $\{\xi_i\}_{i=1}^{m_1}$ represent the domain Ω and the points $\{\eta_j\}_{j=1}^{m_2}$ provide a further refinement according to the density function $\rho(\mathbf{x})$. Jointly, the points $\{\xi_i\}_{i=1}^{m_1} \cup \{\eta_j\}_{j=1}^{m_2}$ are used as pseudo-points to guarantee that the patches, i.e., the spheres, $\{S_i\}_{i=1}^n$ cover the entire domain Ω. The parameter γ is motivated by the requirement that the overlaps of intersecting patches be sufficiently large and is also used to control the smoothness of the shape functions and the density of the stiffness matrix. Similarly, the overall complexity of Algorithm 2 is $\mathrm{O}(\mathrm{card}(\mathscr{P}) \log(n))$.

The ideas behind Algorithm 2 are as follows. First, we find a discrete approximation of the Voronoi regions corresponding to $\{\mathbf{x}_i\}_{i=1}^n$. Then, for each $i = 1, 2, \ldots, n$, we determine the distance between the generator $\{\mathbf{x}_i\}$ and the farthest point in its region. If the number of the pseudo-points is large, then the sphere with $\{\mathbf{x}_i\}$ as its center and the furthest point distance as its radius should almost cover the Voronoi

region associated with $\mathbf{x}_i$. Then, the union of all these spheres should come close to covering the domain Ω. Since the Voronoi regions are in polyhedra in general, the covering is optimal in some sense. The parameter γ is then used to guarantee the complete covering of the entire domain Ω.

Algorithm 2 is generally not suitable if MLSMs are used to construct the shape functions and if one wishes to obtain more than one order of consistency; this is the case for hp-clouds [10], the element-free Galerkin method [5], and reproducing kernel methods [12]. The reason for this is that, in order to make the method work, one needs the points in Ω to be covered multiple times. For example, for a given integer $p > 1$, which depends on the basis space used, it is required that for any $\mathbf{y} \in \overline{\Omega}$, there exit indices $i_1, i_2, \ldots, i_k$, $k \geqslant p$, such that $\mathbf{y} \in \bigcap_{j=1}^{k} S_{i_j}$, i.e., each point in $\overline{\Omega}$ belongs to at least p of the support sets S_i. An algorithm for determining a covering of square support sets satisfying this requirement is given in [10]; it may be easily ammended to apply to the case of spherical support sets. Here, we suggest a meshless variant of that algorithm.

Algorithm 3. Given a region Ω, a density function $\rho(\mathbf{x})$ defined for all $\mathbf{x} \in \overline{\Omega}$, an integer $p > 1$, and a set of points $\{\mathbf{x}_i\}_{i=1}^{n}$ in Ω;

0. choose positive integers m_1 and m_2 and a constant $\gamma > 1$ and set $h_i = 0$ for all $i = 1, 2, \ldots, n$;
1. select a set of m_1 points $\{\xi_i\}_{i=1}^{m_1}$ uniformly distributed over Ω by, e.g., a Monte Carlo method; select another set of m_2 points $\{\eta_j\}_{j=1}^{m_2}$ distributed over Ω according to the density function $\rho(x)$, e.g., again by a Monte Carlo method; let $\mathscr{Q} = \{\xi_i\}_{i=1}^{m_1} \cup \{\eta_j\}_{j=1}^{m_2} \cup \{\mathbf{x}_i\}_{i=1}^{n}$;
2. For all $\mathbf{y} \in \mathscr{Q}$:
   - determine the set $S_{y,R}$ of all points $\mathbf{x}_i$ that fall within a searching sphere B_R which is centered at $\mathbf{y}$ and whose radius is equal to R; if $\mathrm{card}(S_{y,R}) < p$, then increase R and try again; denote the points in the set $S_{y,R}$ by $\mathbf{x}_{k(j)}$, $j = 1, 2, \ldots, \mathrm{card}(S_{y,R})$;
   - determine the distance $d_{k(j)} = \|\mathbf{y} - \mathbf{x}_{k(j)}\|$ for all $\mathbf{x}_{k(j)} \in S_{y,R}$, i.e., for $j = 1, 2, \ldots, \mathrm{card}(S_{y,R})$;
   - sort the set $\{d_{k(j)}\}_{j=1}^{\mathrm{card}(S_{y,R})}$ in increasing order;
   - for $j = 1, 2, \ldots, p$, if $h_{k(j)} < d_{k(j)}$, then set $h_{k(j)} = d_{k(j)}$;
3. set $h_i = \gamma h_i$ for all $i = 1, 2, \ldots, n$.

An efficient implementation of the evaluation of the set $S_{y,R}$, which is a crucial step in this algorithm, is given in [19]. The overall complexity of Algorithm 3 is $O(\mathrm{card}(\mathscr{Q}) \log(n))$. Note that in Algorithm 2, the set $\mathscr{P} = \{\xi_i\}_{i=1}^{m_1} \cup \{\eta_j\}_{j=1}^{m_2}$ was used because $\mathbf{x}_i$ is the center of S_i, i.e., we already have $\mathbf{x}_i \in S_i$ for each i, but $\{\mathbf{x}_i\}_{i=1}^{n}$ is added to $\mathscr{Q}$ in Algorithm 3 since now $p > 1$ and each $\mathbf{x}_i$ also has to be covered, in addition to S_i, by at least $p - 1$ other support regions.

4. Numerical experiments

We now provide the results of some numerical experiments for square, circular, and non-convex regions with both uniform and non-uniform density functions. We will compare centroidal Voronoi point placements (obtained using Algorithm 1) with those obtained using the Monte Carlo method, i.e., random sampling. In addition, for a constant density function, i.e., for a uniform distribution of points, we also make comparisons with the point sets generated via a (2, 3) Halton sequence. [2] In each case, we will use sphere coverings resulting from the application of both Algorithms 2 and 3, the latter with $p = 3$; this is a typical requirement for the cover when using MLSMs to construct shape functions in two dimensions. We choose $\gamma = 1.3$ throughout.

For convenience of description, the characteristic matrix $M = (M_{ij})$ of a sphere covering $\{S_i\}_{i=1}^{n}$ of the domain Ω is defined by

[2] Halton sequences are pseudo-Monte Carlo sequences which can be described as follows. Given a prime number p, for $k \in \mathbb{N}$, k can be represented as $k = \sum_i k_i p^i$. We then define a mapping H_p from $\mathbb{N}$ to [0,1] by $H_p(k) = \sum_i k_i / p^{i+1}$. Then, the (p, q) Halton sequence of n points in two dimensions is defined as $\{H_p(k), H_q(k)\}_{k=1}^{n}$. To generate points in a unit circle through Halton sequences, the following mapping, which is a variant of the mapping in [18], can be used: $(H_p(k), H_q(k)) \mapsto (\sqrt{1 - H_p(k)^2}\cos(2\pi H_q(k)), \sqrt{1 - H_p(k)^2}\sin(2\pi H_q(k)))$.

$$M_{ij} = \begin{cases} 1 & \text{if } S_i \cap S_j \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, the density of M, i.e., the percentage of non-zero entries, is in general proportional to the density of the stiffness matrix which, in practice, one hopes is a sparse matrix.

Except where noted, for all the figures we chose the number of points $n = 128$. However, we also examined the cases $n = 256$ and $n = 512$ in order to gather statistics about the densities of the corresponding characteristic matrices.

4.1. Uniform point distributions

We give a visual comparison of three different point placement schemes for uniform point distributions. For the Monte Carlo (random point placement) and centroidal Voronoi point placements, uniform point distributions means that the density function is constant.

First, we choose the square region $\Omega = [-1, 1] \times [-1, 1]$. In Fig. 1, the top row corresponds to a Monte Carlo point set, the middle row corresponds to a $(2, 3)$ Halton sequence point set, and the bottom row to a

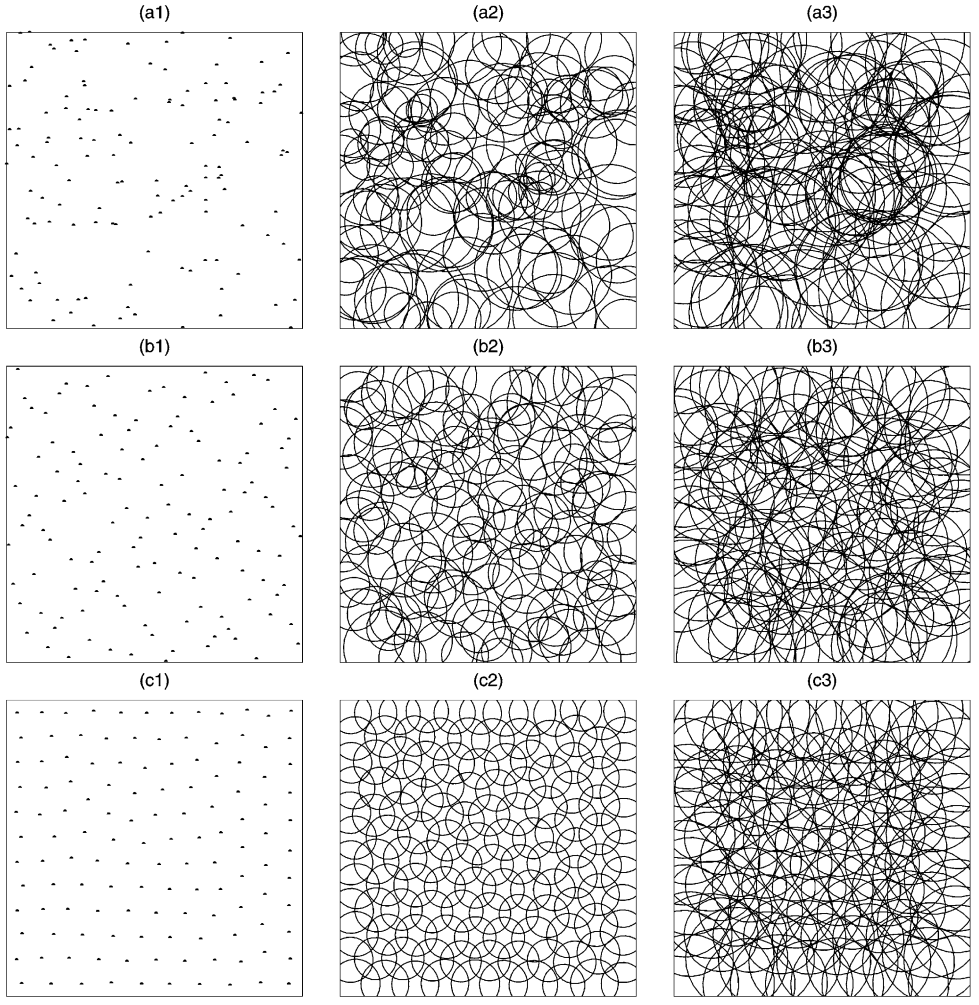

Fig. 1. The sets of 128 points in a square (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top), $(2, 3)$ Halton sequence (middle), and CVT (bottom) point selection methods for a uniform density function.

centroidal Voronoi point set, the first and third of these generated from the density function $\rho(x) = 1$. The first column shows the placement of the 128 points and the second and third columns the coverings of Ω by circular support regions determined by Algorithms 2 and 3, respectively, with $p = 3$ for the latter. We then choose the circular region $\Omega = \{(x_1, x_2) | x_1^2 + x_2^2 \leqslant 1\}$, i.e., the unit circle, for which Fig. 2 provides the same information as did Fig. 1 for the square domain. Then, in Table 1, we provide some information about the sphere coverings. Specifically, we provide, for 128, 256, and 512 point sets, for both the square and circular domains, for the three point placement methods (Monte Carlo, Halton sequence, and centroidal Voronoi), and for the two sphere radii selection methods (Algorithms 2 and 3 with $p = 3$), the densities, i.e., the percentage of non-zero entries, of the characteristic matrices of the corresponding coverings.

From Figs. 1 and 2 and Table 1, one easily concludes that point placement using CVTs results in better point distributions and sphere coverings than do the other methods. Specifically, the point distributions and associated circular patches are much more uniform and the densities of the corresponding characteristic matrices are lower. Note that the densities for Algorithm 3 are higher than for Algorithm 2; this is to be expected since with $p = 3$ Algorithm 3 guarantees (in a probabilistic sense) that each point in Ω is covered by at least three of the circular patches. Note also the inverse linear dependence of the densities on the number of points. We further observe (this is easiest to see for the square domain) that the characteristic matrix densities obtained for the centroidal Voronoi point selection method along with Algorithm 2 for

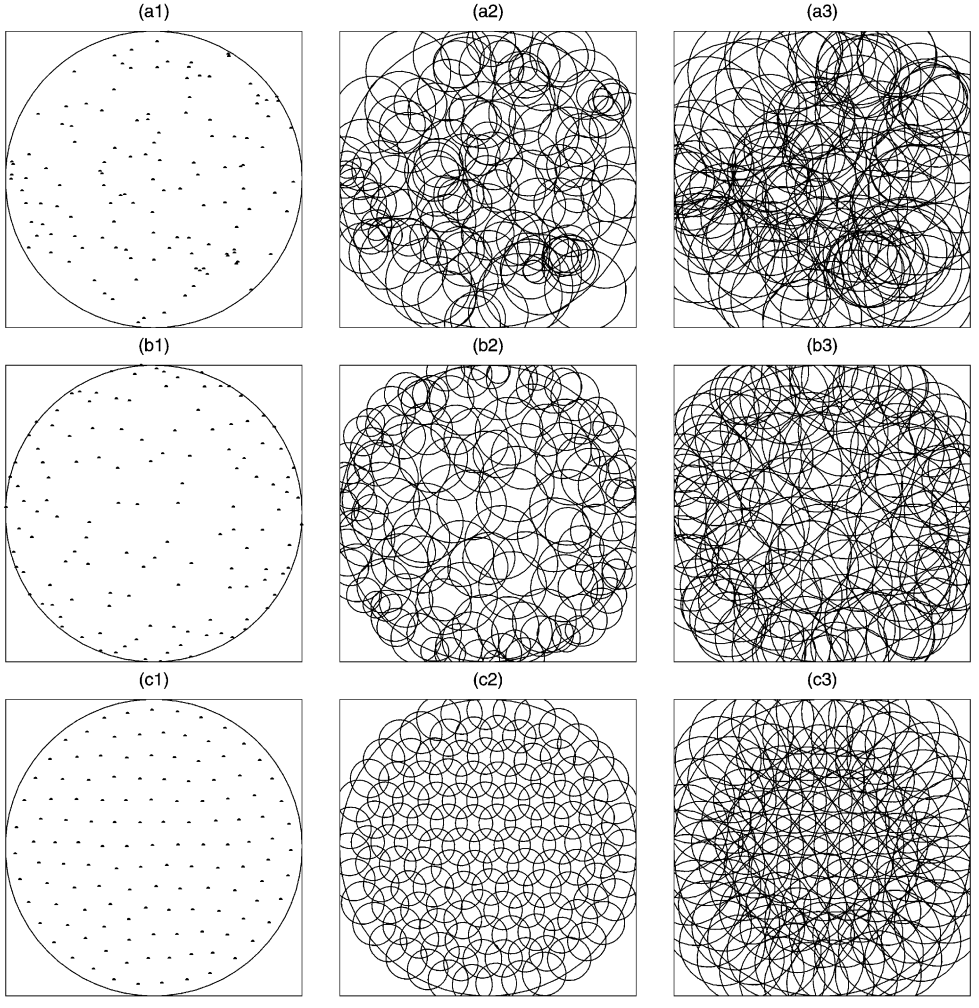

Fig. 2. The sets of 128 points in a circle (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top), $(2, 3)$ Halton sequence (middle), and CVT (bottom) point selection methods for a uniform density function.

Table 1
Densities of non-zero entries of the characteristic matrices of coverings of square and circular domains with uniform point distributions

No. of points	Algorithm	Point selection method		
		Monte Carlo (%)	Halton sequence (%)	CVT (%)
$\Omega = [-1,1] \times [-1,1]$				
128	2	12.02	10.04	5.05
	3 (with $p = 3$)	25.41	21.63	18.68
256	2	6.25	4.95	2.61
	3 (with $p = 3$)	14.23	10.97	9.53
512	2	3.29	2.72	1.33
	3 (with $p = 3$)	7.30	5.91	4.72
$\Omega = \{(x_1, x_2) \mid x_1^2 + x_2^2 \leqslant 1\}$				
128	2	11.90	9.03	5.22
	3 (with $p = 3$)	24.87	19.12	20.94
256	2	6.85	4.20	2.61
	3 (with $p = 3$)	14.99	11.11	9.58
512	2	3.47	2.34	1.33
	3 (with $p = 3$)	7.64	5.50	4.62

determining the radii for the circular patches are very much the same as that which one would obtain for a piecewise linear finite element grid. If Algorithm 3 with $p = 3$ is used instead, then the characteristic matrix densities are very much the same as for a piecewise quadratic finite element grid. If either of the other point selection methods are used, the corresponding matrix densities are substantially higher.

We describe how centroidal Voronoi point placements were obtained for circular domains since it helps to point out a very favorable feature of our completely probabilistic method of determining point placements and support radii. We embed the circular domain into a square domain and sample points in the square; if a sampled point lies outside the circle, we simply reject it. This is equivalent to defining the density function to vanish outside the circle. This procedure can be generalized to arbitrary convex domains; one simply embeds the domain into a rectangular domain, then one samples on the rectangular domain and rejects points lying outside the given domain (or, equivalently, one extends the density function by zero outside of the given domain and then samples on the rectangular domain.) Ultimately, we only need the ability to determine if a point is inside or outside of the given domain. Non-convex domains need an additional rejection step; see Section 4.3.

4.2. Non-uniform point distributions

In Figs. 3–7, we present, for Monte Carlo and centroidal Voronoi point placements, the same information as is found in Figs. 1 and 2, except that now we consider non-uniform point distributions. Specifically, for Figs. 3–5, which are for the square domain $[-1,1] \times [-1,1]$, the points are, respectively, distributed according to the density functions $e^{-3(x^2+y^2)}$ (which has a peak at the center of the square), $e^{-1.5(2+x+y)}$ (which has a peak at the bottom-left corner of the square), and $e^{-(1+x)^2}$ (which peaks at the left side of the square.) For Figs. 6 and 7, which are for the circular domain $\{(x_1, x_2) | x_1^2 + x_2^2 \leqslant 1\}$, the points are, respectively, distributed according to the density functions $e^{-4(x^2+y^2)}$ (which has a peak at the center of the circle) and $e^{-3(1-x^2-y^2)}$ (which peaks at the circumference of the circle).

Table 2 provides the densities, i.e., the percentage of non-zero entries, of the characteristic matrices for circular coverings associated with 128, 256, and 512 points generated for the two domains (square and circle), the two point placement methods (Monte Carlo and centroidal Voronoi), the two radii selection methods (Algorithms 2 and 3 with $p = 3$), and for the non-uniform density functions given above. From Figs. 3–7 and Table 2, we again see the clear superiority of the centroidal Voronoi point placement method.

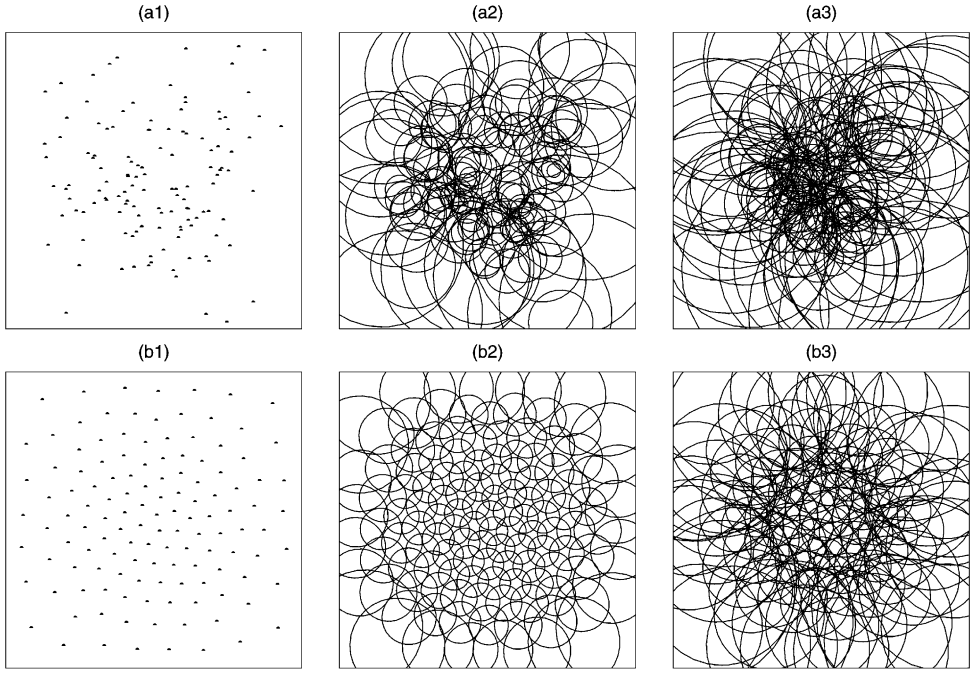

Fig. 3. The sets of 128 points in a square (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-3(x^2+y^2)}$.

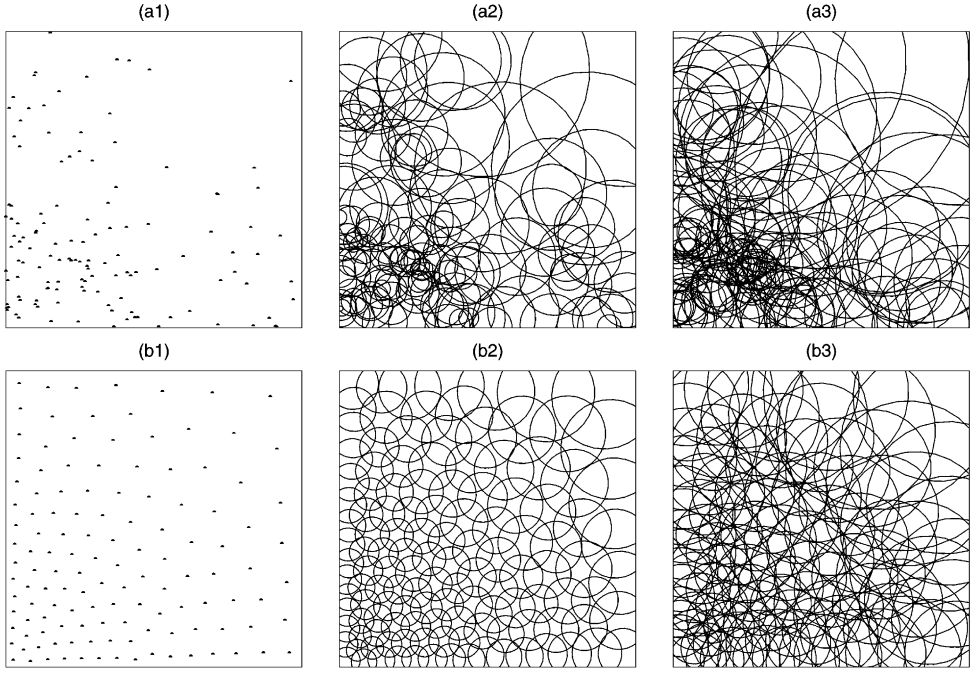

Fig. 4. The sets of 128 points in a square (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-1.5(2+x+y)}$.

4.3. Non-convex domains

Our point placement and support region selection methods are easily adapted to non-convex domains with relatively well-behaved boundaries. Only an additional rejection step is necessary in Algorithm 1 and a

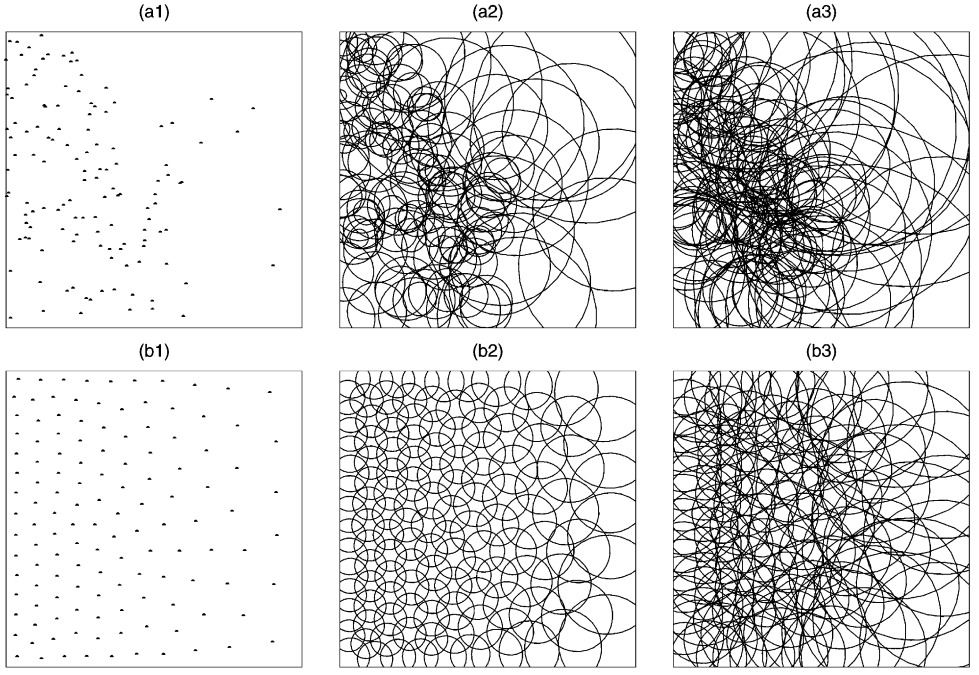

Fig. 5. The sets of 128 points in a square (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-(1+x)^2}$.

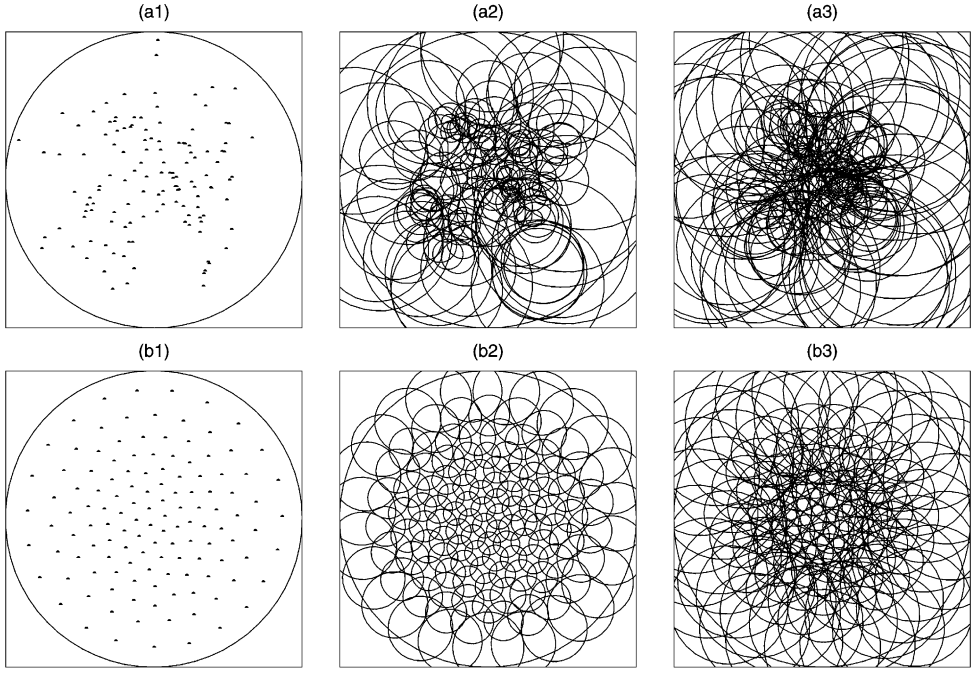

Fig. 6. The sets of 128 points in a circle (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-4(x^2+y^2)}$.

change in the closest point determination may be necessary; otherwise, all the algorithms we have described remain unchanged. The change to Algorithm 1 is simply that whenever the point location $\mathbf{x}_i^*$ computed according to step 3 of that algorithm lies outside the domain, then it is rejected. (This cannot occur for

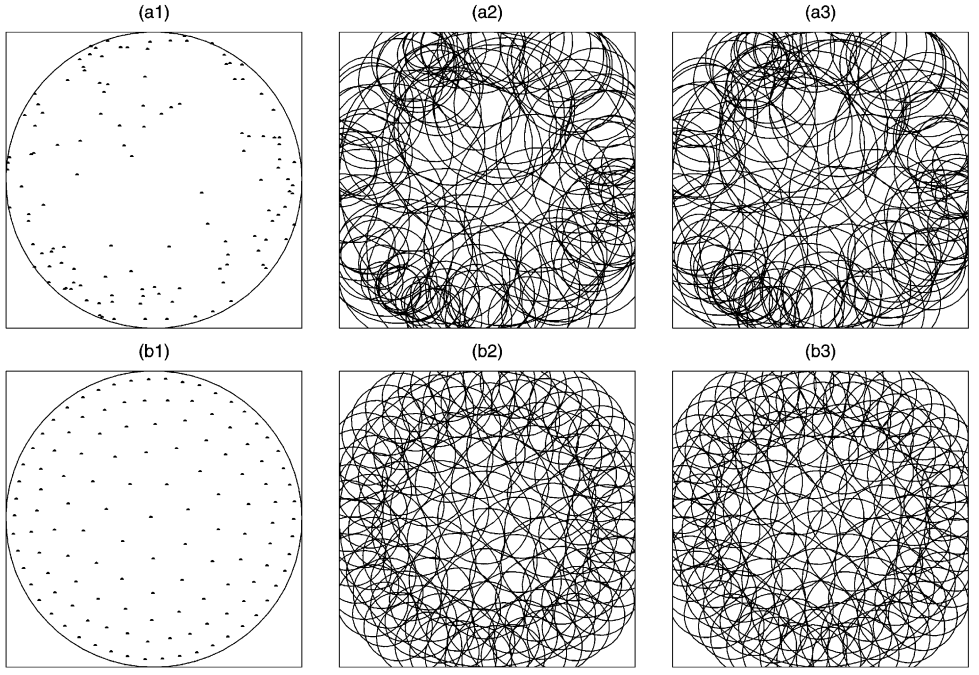

Fig. 7. The sets of 128 points in a circle (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-3(1-x^2-y^2)}$.

convex domains but is possible for non-convex domains.) Equivalently, if the density function at that point vanishes, i.e., if $\rho(\mathbf{x}_i^*) = 0$, then the point $\mathbf{x}_i^*$ is rejected.

For non-convex domains having ill-behaved boundaries such as slits or cracks, it is possible that the line segment connecting a sampled point to the nearest (in terms of Eudlicean distance) generator crosses the boundary of the domain. This situation should be rejected, i.e., Voronoi regions should not have part of the domain boundary traversing through them. The rejection of such situations can be effected automatically through a simple redefinition of the notion of distance. If $\partial\Omega$ denotes the boundary of Ω, we redefine the distance used for searching for closest points in all algorithms as follows. Given two points $\mathbf{y}_1$ and $\mathbf{y}_2$ in Ω, let $\overline{\mathbf{y}_1\mathbf{y}_2}$ denote the line segment between $\mathbf{y}_1$ and $\mathbf{y}_2$. Then, the distance between $\mathbf{y}_1$ and $\mathbf{y}_2$ is given by

$$\mathrm{d}(\mathbf{y}_1, \mathbf{y}_2) = \begin{cases} \infty & \text{if } \partial\Omega \cap \overline{\mathbf{y}_1\mathbf{y}_2} \neq \emptyset, \\ \|\mathbf{y}_1 - \mathbf{y}_2\| & \text{otherwise.} \end{cases}$$

(Note again that for convex domains, we always have that $\partial\Omega \cap \overline{\mathbf{y}_1\mathbf{y}_2} = \emptyset$.)

We first illustrate the superior performance of our algorithms for point placement and support radii determination for non-convex domains in Figs. 8–10 where for three different non-convex domains, we provide (for the Monte Carlo and centroidal Voronoi point placement methods) the same information as found in Figs. 1 and 2 for a square and circle. Figs. 8–10 are for a uniform density function; there is no difficulty applying our methods to non-uniform density functions over these non-convex domains.

As a further illustration, we next consider the square region $\Omega = [-1, 1] \times [-1, 1]$ with a slit $L = \{(0, y) | y \in [-0.5, 0.5]\}$. As noted above, the Voronoi region of any point cannot cross over the slit, i.e., any two points should not influence each other if the line segment connecting them intersects the slit. Numerical experiments were performed for both a uniform and a non-uniform density function; see Figs. 11 and 12 for a depiction of the placement of points and support radii; here 256 points are used. The density function for Fig. 12 is given by

$$\rho(x, y) = \begin{cases} e^{-10r} & \text{if } r < 0.6, \\ e^{-6} & \text{otherwise,} \end{cases} \quad \text{where } r = \begin{cases} \sqrt{x^2 + (y - 0.5)^2} & \text{if } y > 0, \\ \sqrt{x^2 + (y + 0.5)^2} & \text{otherwise.} \end{cases}$$

Table 2
Densities of non-zero entries of the characteristic matrices of coverings of square and circular domains with non-uniform point distributions

Density function	No. of points	Algorithm	Point placement method	
			Monte Carlo (%)	CVT (%)
$\Omega = [-1,1] \times [-1,1]$				
$e^{-3(x^2+y^2)}$	128	2	18.58	6.50
		3 (with $p = 3$)	48.70	28.36
	256	2	7.87	2.91
		3 (with $p = 3$)	21.04	12.46
	512	2	3.97	1.44
		3 (with $p = 3$)	9.96	5.58
$e^{-1.5(2+x+y)}$	128	2	12.57	5.27
		3 (with $p = 3$)	32.80	21.74
	256	2	7.18	2.67
		3 (with $p = 3$)	16.01	10.48
	512	2	3.25	1.36
		3 (with $p = 3$)	7.38	5.12
$e^{-(1+x)^2}$	128	2	15.85	5.26
		3 (with $p = 3$)	39.53	22.05
	256	2	7.84	2.72
		3 (with $p = 3$)	18.17	10.47
	512	2	3.36	1.35
		3 (with $p = 3$)	7.60	5.05
$\Omega = \{(x_1, x_2) \mid x_1^2 + x_2^2 \leqslant 1\}$				
$e^{-4(x^2+y^2)}$	128	2	20.75	5.34
		3 (with $p = 3$)	48.06	26.41
	256	2	9.13	2.94
		3 (with $p = 3$)	23.49	12.11
	512	2	3.90	1.42
		3 (with $p = 3$)	10.01	5.61
$e^{-3(1-x^2-y^2)}$	128	2	12.23	5.34
		3 (with $p = 3$)	25.26	18.59
	256	2	7.22	2.67
		3 (with $p = 3$)	15.78	10.12
	512	2	3.27	1.35
		3 (with $p = 3$)	7.48	5.01

The choice of the density function reflects the desire to pack more points near the two ends of the slit to resolve any possible singularities.

4.4. Point placement under constraints

It is possible to incorporate the techniques for constrained CVT discussed in Section 2 in order to fullfil the requirement of placing some points on the boundary as is required in many current meshless methods. Fig. 13 provides a test example for a square domain. Here, we construct the CVT using Algorithm 1 as usual, but whenever a new set of generators were computed (after step 3) at each iteration, the points whose Voronoi regions include part of a side of the square were projected onto the side along its normal direction and points whose Voronoi regions involve one of the four corners of the square were replaced by the corresponding corner point. Fig. 14 illustrates the same situation for the unit circle where the projection procedure is even simpler in the sense that the points whose Voronoi regions include part of the boundary are automatically projected onto the boundary along the radial direction. Although only uniform density functions are used in Figs. 13 and 14, the same projection procedure can be applied to any non-uniform density function. For example, in Fig. 15, we provide an example of a constrained CVT for a circular domain for the density function $\rho(x, y) = e^{-2.5(1+y)}$.

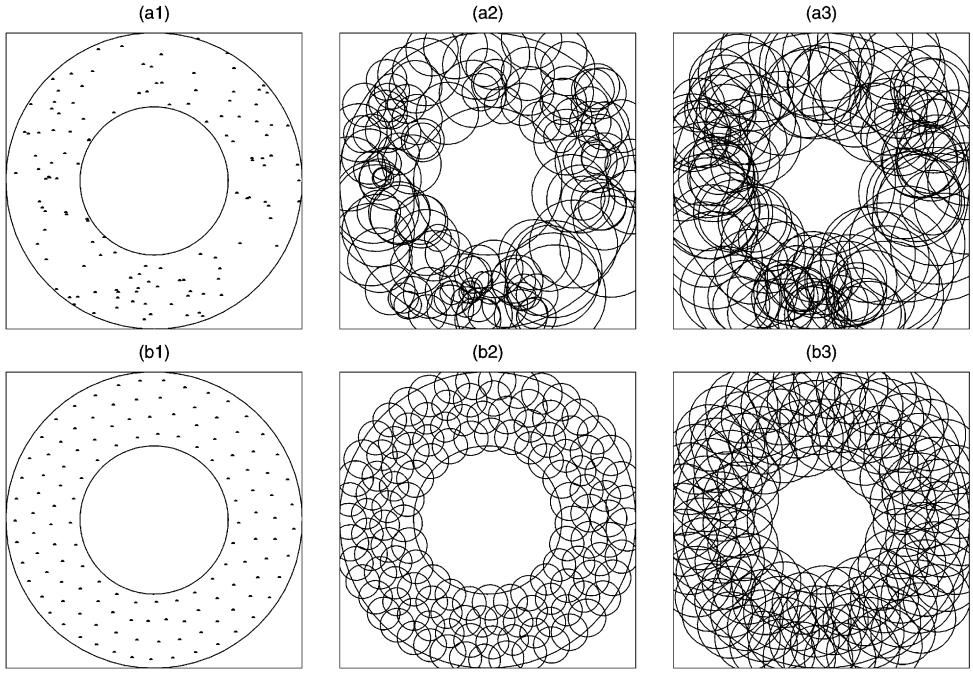

Fig. 8. The sets of 128 points in a non-convex domain (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function.

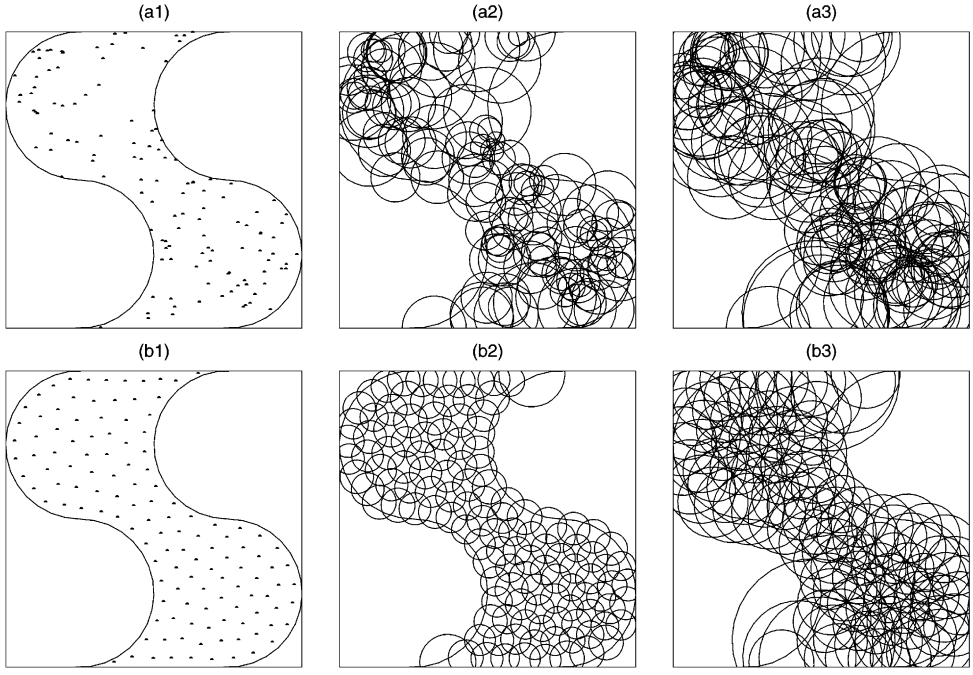

Fig. 9. The sets of 128 points in a non-convex domain (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function.

4.5. Three-dimensional domains

Algorithms 1–3 can be easily applied to any space dimension since the procedure is not related to the dimension of the space except for the concept of distances and the sorting process. Some numerical

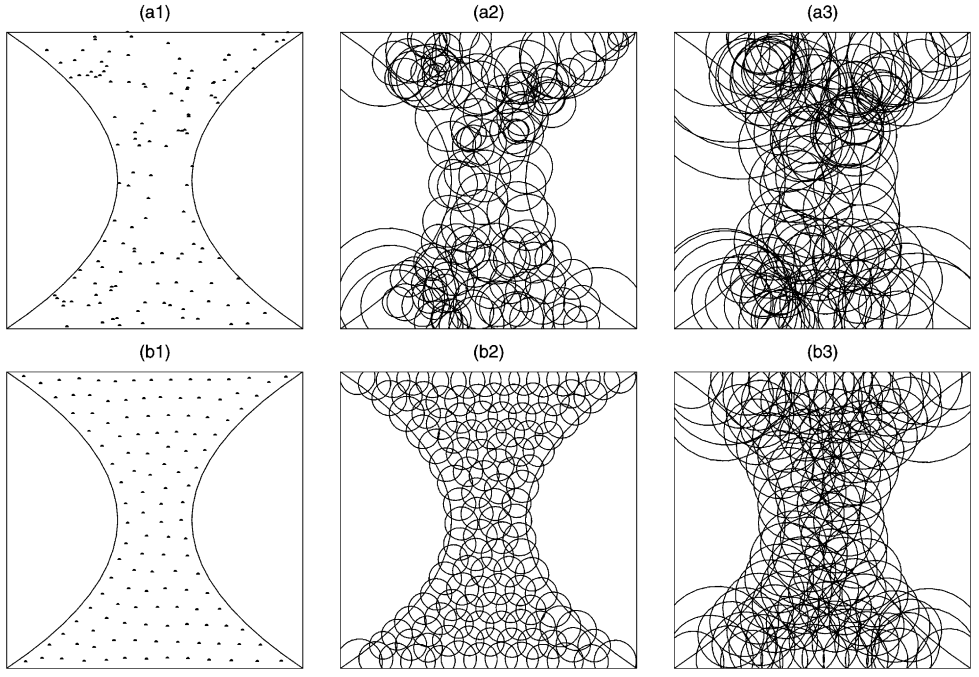

Fig. 10. The sets of 128 points in a non-convex domain (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function.

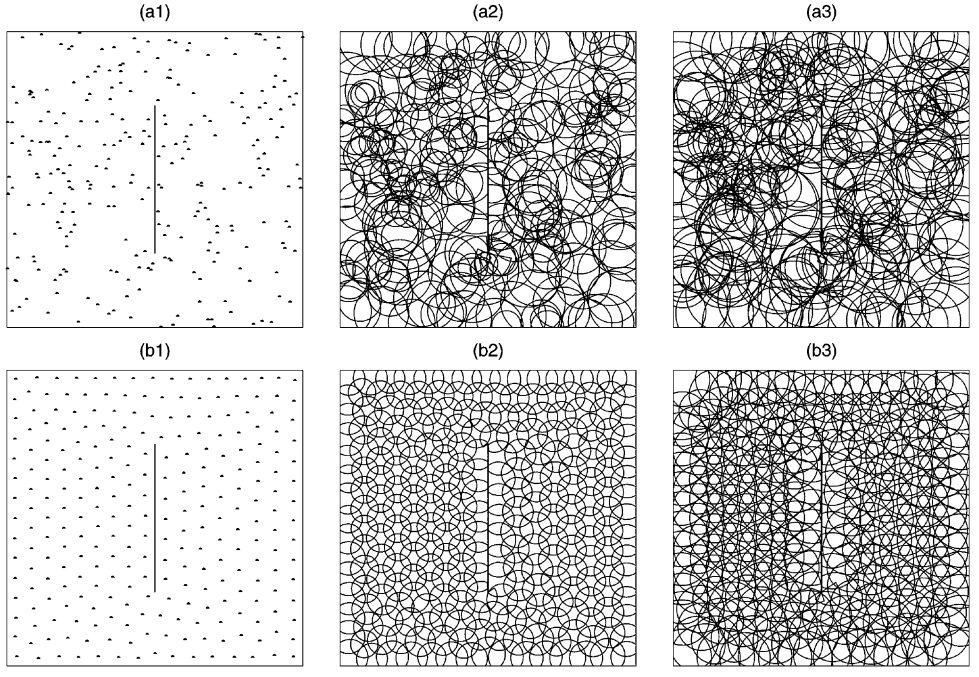

Fig. 11. The sets of 256 points in a domain with a slit (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function.

experiments with $n = 512$ points were performed in $\Omega = ([-1,1] \times [-0.5,0.5] \times [-1,1]) \cup ([0,1] \times [-0.5,0.5] \times [0,1])$. Since Ω is also non-convex, we used a rejection technique similar to that described in Section 4.3 for domains with relatively well-behaved boundaries. Due to the difficulty in visualizing

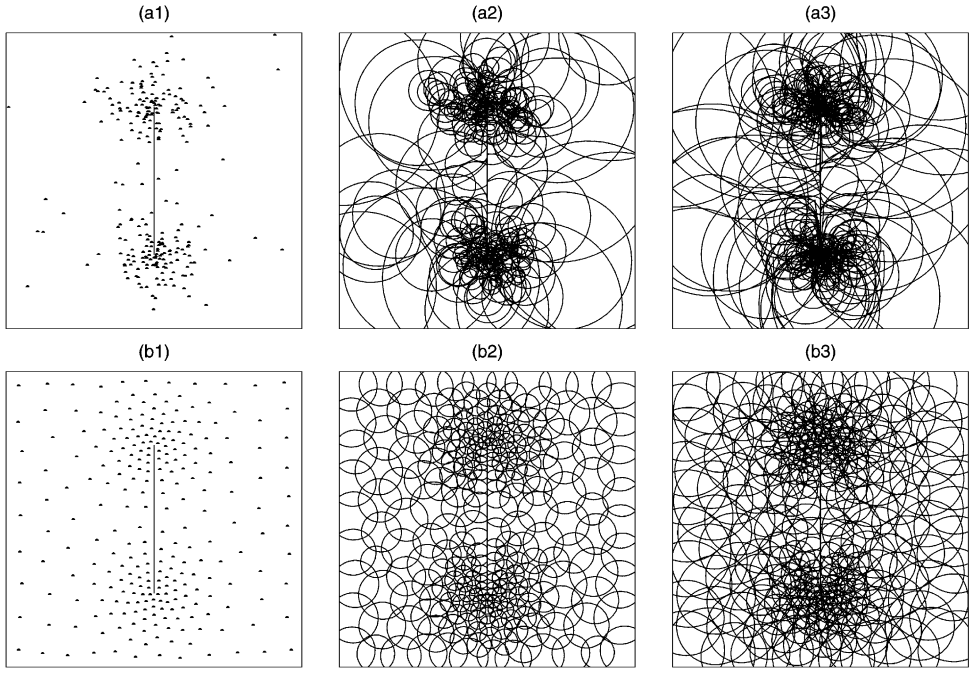

Fig. 12. The sets of 256 points in a domain with a slit (left) and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a non-uniform density function.

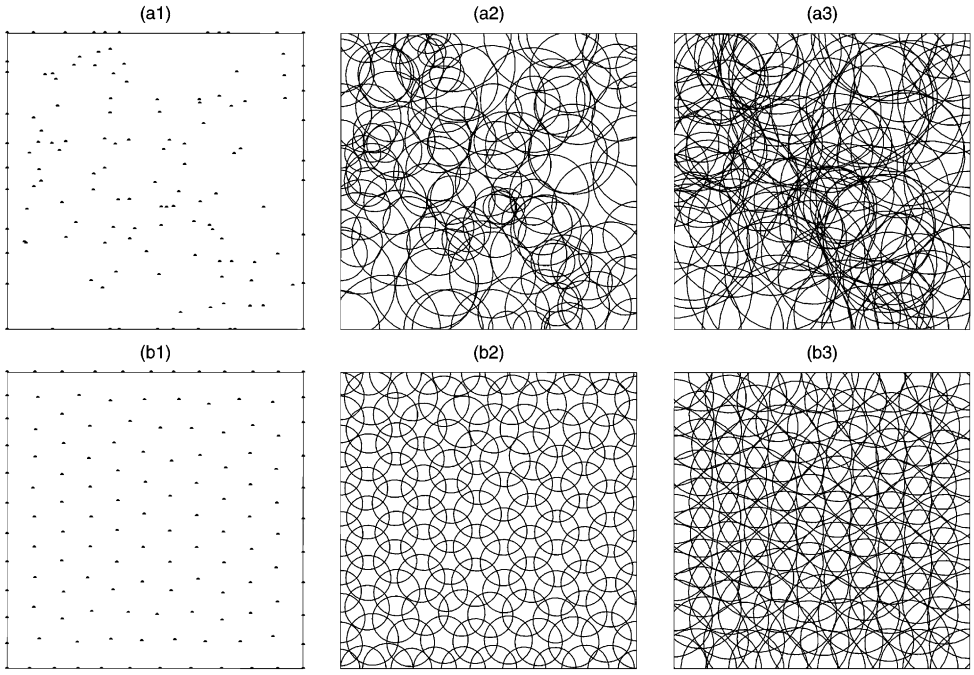

Fig. 13. The sets of 128 points in a square (left) under constraint and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function.

the spherical patches in three dimensionals, we here only give the scatter plots of the 512 points. Fig. 16 provides the plots for a uniform density function and Fig. 17 for the a non-uniform the density function

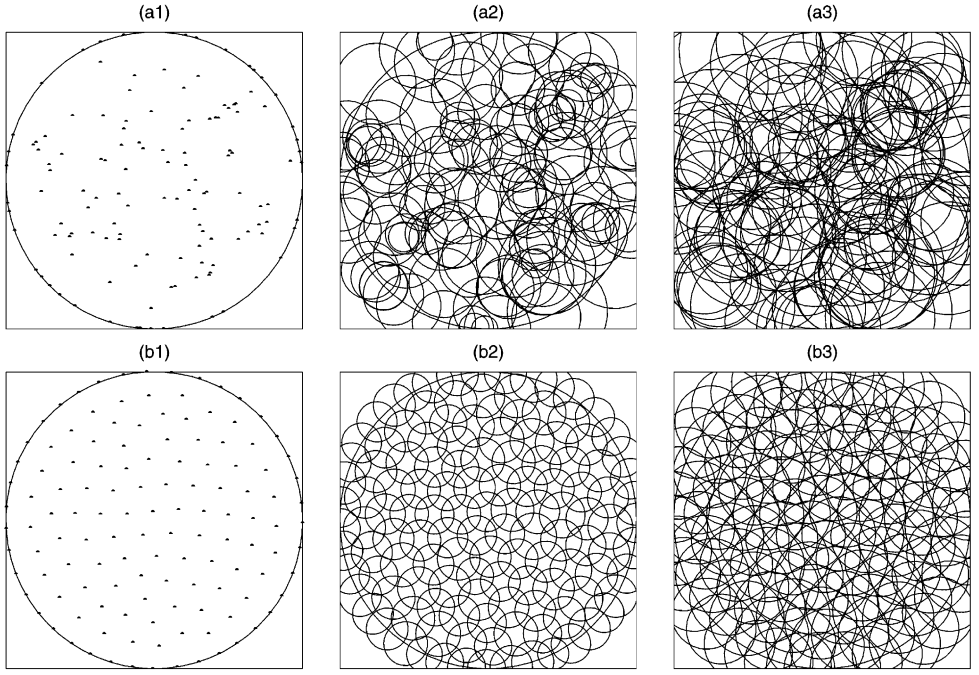

Fig. 14. The sets of 128 points in a circle (left) under constraint and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function.

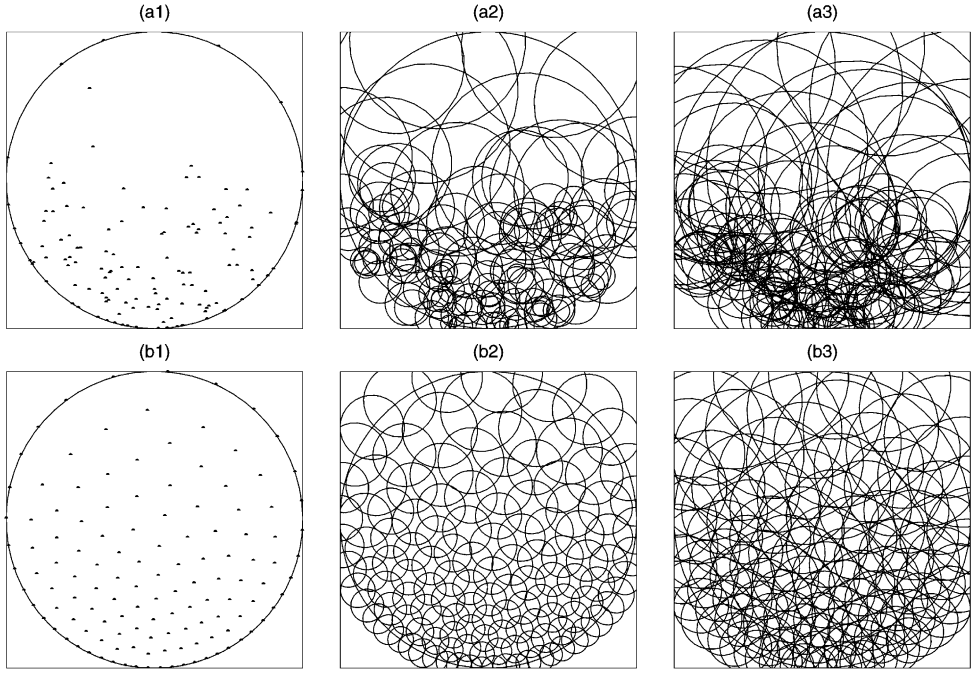

Fig. 15. The sets of 128 points in a circle (left) under constraint and the associated spherical patches determined by Algorithms 2 (middle) and 3 (with $p = 3$) (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $\rho(x, y) = \mathrm{e}^{-2.5(1+y)}$.

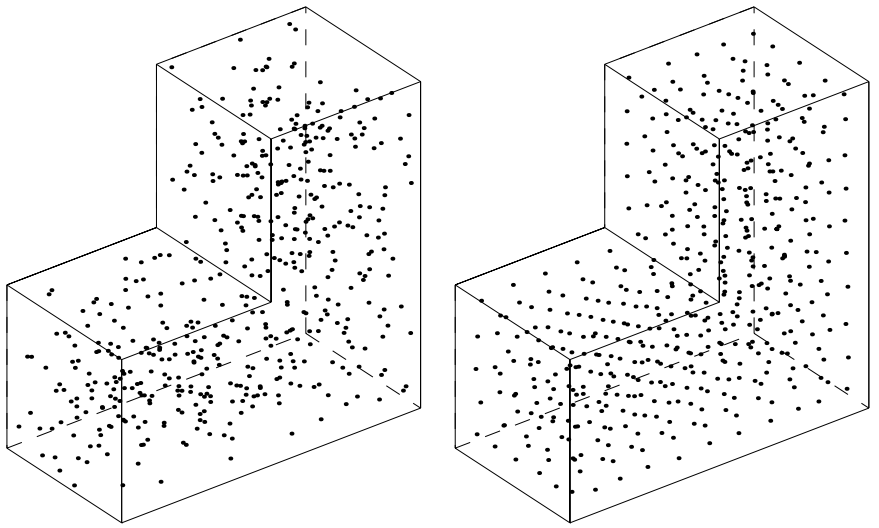

Fig. 16. The sets of 512 points in a three-dimensional non-convex domain for the Monte Carlo (left) and CVT (right) point selection methods for a uniform density function.

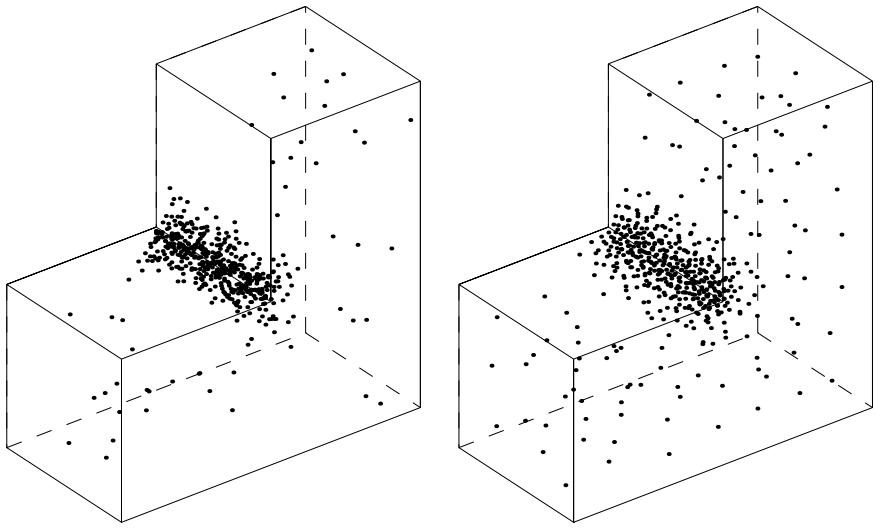

Fig. 17. The sets of 512 points in a three-dimensional non-convex domain for the Monte Carlo (left) and CVT (right) point selection methods for a non-uniform density function.

$$\rho(x,y,z) = \begin{cases} \mathrm{e}^{-20r} & \text{if } r < 0.4, \\ \mathrm{e}^{-8} & \text{otherwise,} \end{cases}$$

where $r = \sqrt{x^2 + z^2}$. The centroidal Voronoi point placement method again provides point sets of superior quality.

5. Concluding remarks

The new algorithms we have presented result in, for both uniform and non-uniform point distributions, high-quality point sets and high-quality support regions. Furthermore, being probabilistic in nature, the

algorithms are totally meshfree, i.e., they do not require, at any stage, the use of coarse or fine meshes. They also can be easily parallelized. In [9], a parallel implementation of Algorithm 1 for determining point placements is given which exhibits near perfect linear speed up respective to the number of processors. Algorithms 2 and 3 for support radii determination also possess obvious opportunities for parallelization.

In addition to testing our point placement and support radii determination algorithms in complex three dimensional domains, we are currently studying a number of other issues connected with the implementation and use of the algorithms. These include the following.

- Connecting the density function used for generating centroidal Voronoi point sets to a priori and a posteriori error estimates; then adaptive point generation algorithms could be easily defined.
- Using anisotropic (tensor valued) density functions for determining point distributions with special properties such as high aspect ratio spacings; these would be useful for, e.g., resolving boundary layers.
- Testing the use of the algorithms in function approximation methods, e.g., interpolation, and in Galerkin methods for the numerical solution of partial differential equations.
- Analyzing properties of the algorithms such as convergence behavior.

Acknowledgements

Supported by the National Science Foundation under grant number CCR-9988303.

References

[1] S. Arya, D. Mount, N. Netanyahu, R. Silverman, A. Wu, An optimal algorithm for approximate nearest neighbor searching, J. ACM 45 (1998) 891–923.
[2] I. Babuska, J. Melenk, The partition of unity finite element method: basic theory and applications, Comput. Methods Appl. Mech. Engrg. 139 (1996) 289–314.
[3] J. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (1975) 509–517.
[4] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: an overview and recent developments, Comput. Methods Appl. Mech. Engrg. 139 (1996) 3–47.
[5] T. Belytschko, Y. Lu, L. Gu, Element-free Galerkin methods, Int. J. Numer. Meth. Engrg. 37 (1994) 229–256.
[6] Y. Choi, S. Kim, Node generation scheme for meshfree method by Voronoi diagram and weighted bubble packing, in: Proceedings of Fifth US National Congress on Computational Mechanics, Boulder, 1999.
[7] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, SIAM Rev. 41 (1999) 637–676.
[8] Q. Du, M. Gunzburger, Grid generation and optimization based on centroidal Voronoi tessellations, Appl. Math. Comput., to appear.
[9] Q. Du, M. Gunzburger, L.-L. Ju, Probablistic methods for centroidal Voronoi tessellations and their parallel implementations, Parallel Comput., submitted.
[10] C. Duarte, J.T. Oden, Hp clouds – a meshless method to solve boundary value problems, Numer. Meth. PDE 12 (1996) 673–705.
[11] M. Griebel, M. Schweitzer, A particle-partition of unity method for the solution of elliptic, parabolic and hyperbolic PDEs, SIAM J. Sci. Comput. 22 (2001) 853–890.
[12] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, Int. J. Numer. Meth. Fluids 20 (1995) 1081–1106.
[13] X.-Y. Li, S.-H. Teng, A. Ungor, Biting: advancing front meets sphere packing, Int. J. Numer. Meth. Engrg. 49 (2000) 61–91.
[14] S. Lloyd, Least square quantization in PCM, IEEE Trans. Inform. Theory 28 (1982) 129–137.
[15] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: L. Le Cam, J. Neyman (Eds.), Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. I, University of California, Berkeley, 1967, pp. 281–297.
[16] A. Okabe, B. Boots, K. Sugihara, Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, Wiley, Chichester, 1992.
[17] H. Samet, The Design and Analysis of Spatial Data Structures, Addison-Wesley, Reading, MA, 1990.
[18] J. Spanier, E. Gelbard, Monte Carlo Principles and Neutron Transport Problems, Addison-Wesley, New York, 1969.
[19] J. Swegle, S. Attaway, F. Mello, D. Hicks, An analysis of the smoothed particle hydrodynamics, Tech. Rep. SAND93-2513, Sandia National Laboratories, Albuquerque, 1994.