Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2017) 478-491

ECCOMAS
Proceedia

UNCECOMP 2017
2nd ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Rhodes Island, Greece, 15–17 June 2017

VORONOI WEIGHTING OF SAMPLES IN MONTE CARLO INTEGRATION

Miroslav Vořechovský[1], Václav Sadílek[2] and Jan Eliáš[3]

[1] Brno University of Technology
Veveří 331/95, 602 00 Brno, Czech Republic
e-mail: vorechovsky.m@vut.cz, {sadilek.v, elias.j}@fce.vutbr.cz

Keywords: Voronoi tessellation, Reweighting, Monte Carlo integration, Weighted statistical estimators, Sampling plan.

Abstract. *The standard way to numerically calculate integrals such as the ones featured in estimation of statistical moments of functions of random variables using Monte Carlo procedure is to: (i) perform selection of samples from the random vector, (ii) approximate the integrals using averages of the functions evaluated at the sampling points. If the N_{sim} points are selected with an equal probability (with respect to the joint distribution function) such as in Monte Carlo sampling, the averages use equal weights $1/N_{\text{sim}}$. The problem with Monte Carlo sampling is that the estimated values exhibit a large variance due to the fact that the sampling points are usually not spread uniformly over the domain of sampling probabilities. One way to improve the accuracy would be to perform a more advanced sampling.*

The paper explores another way to improve the Monte Carlo integration approach: by considering unequal weights. These weights are obtained by transforming the sampling points into sampling probabilities (points within a unit hypercube), and subsequently by associating the sampling points with weights obtained as volumes of regions/cells around the sampling points within a unit hypercube. These cells are constructed by the Voronoi tessellation around each point. Supposedly, this approach could have been considered superior over the naive one because it can suppress inaccuracies stemming from clusters of sampling points.

The paper also explores utilization of the Voronoi diagram for identification of optimal locations for sample size extension.

Peer-review under responsibility of the organizing committee of UNCECOMP 2017.
doi: 10.7712/120217.5385.17023

1 INTRODUCTION

Monte Carlo estimation of statistical integrals is encountered in numerous applications. A typical example is the computer exploration of functions that feature random variables. These random variables form an N_{var}-dimensional vector, where N_{var} is the number of random variables considered. In computer experiments the first step is a selection of optimal sample set, i.e. selection of N_{sim} points from the N_{var} dimensional space. These points then form the sampling plan which is an $N_{\text{sim}} \times N_{\text{var}}$ matrix. The methods used for formulating the plan of experimental points are collectively known as Design of Experiments (DoE). The purpose of DoE is to provide a set of points lying inside a chosen *design domain* that are optimally distributed; the optimality of the sample depends on the nature of the problem. Various authors have suggested intuitive goals for good designs, including "good coverage", the ability to fit complex models, many levels for each factor/variable, and good projection properties. At the same time, a number of different mathematical criteria have been put forth for comparing designs.

The design of experiments is typically performed in a hyper-cubical domain of N_{var} dimensions, where each dimension/variable, U_v, ranges between zero and one ($v = 1, \ldots, N_{\text{var}}$). This *design domain* is to be covered by N_{sim} points as evenly as possible as the points within the design domain represent sampling probabilities. The probability that the i-th experimental point will be located inside some chosen subset of the domain must be equal to V_S/V_D, with V_S being the subset volume and V_D the volume of the whole domain (for unconstrained design $V_D = 1$). Whenever this is valid, the design criterion will be called *statistically uniform*. Moreover, each separate sampling plan should have the points spread evenly over the design domain. Even though such uniformity is conceptually simple and intuitive on a qualitative level, it is somewhat complicated to describe and characterize it mathematically. Though some problems do not require this uniformity, it is the crucial assumption in Monte-Carlo integration and its violation may lead to significant errors [5, 11].

There exist many other criteria of optimality of the sampling plan: e.g. the Audze-Eglājs (AE) criterion [1] later generalized into the so-called ϕ criterion, the Euclidean MaxiMin and MiniMax distance between points, various measures of discrepancy, criteria based on correlation (orthogonality), designs maximizing entropy and many others. It should also be noted that an experimental design can be also obtained via so-called "quasi-random" low-discrepancy sequences (deterministic versions of MC analysis) that can often achieve reasonably uniform sample placement in hypercubes (Niederreiter, Halton, Sobol', Hammersley, etc.).

As mentioned above, the selection of the sampling points is a crucial step when evaluating approximations to integrals as the ones performed in Monte Carlo simulations (numerical integration). In such applications, equal sampling probabilities inside the design domain are required.

In this article, it is assumed that the sampling points have already been selected and they are not spread optimaly over the design domain. A typical example may be a sample selected using crude Monte Carlo sampling. The article considers the possibility to improve quality of Monte Carlo estimation with such a given sample. The only possibility to improve the estimations of the integrals is to vary the weights associated with individual sampling points. Motivated by the MiniMax criterion of optimality [6], we explore the possibility to improve the quality of statistical estimations using Voronoi tessellation, i.e. a particular form of partitioning of the design domain around given sampling points. The *design domain* to be partitioned is the unit hypercube described above and therefore the volumes around individual sampling points represent weights (probabilities) to be used in the weighted averages that estimate the integrals.

2 STATISTICAL MOMENT ESTIMATION USING MONTE CARLO SAMPLING

As mentioned in the introduction, one of the frequent uses of DoE is *statistical sampling* for Monte Carlo integration. We present the application of statistical sampling to the problem of estimating statistical moments of a function of random variables. In particular, a deterministic function, $Z = g(\boldsymbol{X})$, is considered, which can be a computational model or a physical experiment. Z is the uncertain response variable (or generally a vector of the outputs). The vector $\boldsymbol{X} \in \mathbb{R}^{N_{\text{var}}}$ is considered to be a random vector of N_{var} continuous marginals (input random variables describing uncertainties/randomness) with a given joint probability density function (PDF).

Estimation of the statistical moments of variable $Z = g(\boldsymbol{X})$ is, in fact, an estimation of integrals over domains of random variables weighted by a given joint PDF of the input random vector, $f_{\boldsymbol{X}}(\boldsymbol{x})$. We seek the statistical parameters of $Z = g(\boldsymbol{X})$ in the form of the following integral:

$$\mathrm{E}[S[g(\boldsymbol{X})]] = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} S[g(\boldsymbol{x})] \, \mathrm{d}F_{\boldsymbol{X}}(\boldsymbol{x}) \tag{1}$$

where $\mathrm{d}F_{\boldsymbol{X}}(\boldsymbol{x}) = f_{\boldsymbol{X}}(\boldsymbol{x}) \cdot \mathrm{d}x_1 \, \mathrm{d}x_2 \cdots \mathrm{d}x_{N_{\text{var}}}$ is the infinitesimal probability ($F_{\boldsymbol{X}}$ denotes the joint cumulative density function) and where the particular form of the function $S[g(\cdot)]$ depends on the statistical parameter of interest. For example, to gain the mean value of $g(\cdot)$, $S[g(\cdot)] = g(\cdot)$; higher statistical moments of Z can be obtained by integrating polynomials of $g(\cdot)$. The probability of failure (an event defined as $g(\cdot) < 0$) is obtained in a similar manner: $S[\cdot]$ is replaced by the Heaviside function (or indicator function) $H[-g(\boldsymbol{X})]$, which equals one for a failure event ($g < 0$) and zero otherwise. In this way, the domain of integration of the PDF is limited to the failure domain.

In Monte Carlo sampling, which is the most prevalent statistical sampling technique, the above integrals are numerically estimated using the following procedure: (i) draw N_{sim} realizations of $\boldsymbol{X}$ that share the same probability of occurrence $1/N_{\text{sim}}$ by using its joint distribution $f_{\boldsymbol{X}}(\boldsymbol{x})$; (ii) compute the same number of output realizations of $S[g(\cdot)]$; and (iii) estimate the desired parameters as arithmetical averages. We now limit ourselves to independent random variables in vector $\boldsymbol{X}$. The aspect of the correct representation of the target joint PDF of the inputs mentioned in item (i) is absolutely crucial. Practically, this can be achieved by reproducing a *uniform distribution* in the design space (unit hypercube) that represents the space of sampling probabilities.

Assume now a random vector $\boldsymbol{U}$ that is selected from a multivariate uniform distribution in such a way that its independent marginal variables U_v, $v = 1, \dots, N_{\text{var}}$, are uniform over intervals $(0;1)$. A vector with such a multivariate distribution is said to have an "independence copula" [7]

$$C(u_1, \dots, u_{N_{\text{var}}}) = \mathrm{P}(U_1 \le u_1, \dots, U_{N_{\text{var}}} \le u_{N_{\text{var}}}) = \prod_{v=1}^{N_{\text{var}}} u_v \tag{2}$$

These uniform variables can be seen as sampling probabilities: $F_{X_v} = U_v$. The joint cumulative distribution function then reads $F_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_v F_{X_v} = \prod_v U_v$, and $\mathrm{d}F_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_v \mathrm{d}U_v$. The individual random variables can be obtained by inverse transformations

$$\{X_1, \dots, X_{N_{\text{var}}}\} = \{F_1^{-1}(U_1), \dots, F_{N_{\text{var}}}^{-1}(U_{N_{\text{var}}})\} \tag{3}$$

and similarly the realizations of the original random variables are obtained by the componentwise inverse distribution function of a point $\boldsymbol{u}$ (a realization of $\boldsymbol{U}$) representing a sampling

probability

$$\boldsymbol{x} = \{x_1, \ldots, x_{N_{\text{var}}}\} = \{F_1^{-1}(u_1), \ldots, F_{N_{\text{var}}}^{-1}(u_{N_{\text{var}}})\} \tag{4}$$

With the help of this transformation from the original to the uniform joint PDF, the above integral in Eq. (1) can be rewritten as

$$\begin{aligned} \mathrm{E}[S[g(\boldsymbol{X})]] &= \int_0^1 \ldots \int_0^1 S[g(\boldsymbol{x})] \; \mathrm{d}C(u_1, \ldots, u_{N_{\text{var}}}) \\ &= \int_{[0,1]^{N_{\text{var}}}} S[g(\boldsymbol{x})] \prod_{v=1}^{N_{\text{var}}} \mathrm{d}U_v \end{aligned} \tag{5}$$

so that the integration is performed over a unit hypercube with uniform unit density.

We now assume an estimate of this integral by the following statistic (the average computed using N_{sim} realizations of $\boldsymbol{U}$, namely the sampling points $\boldsymbol{u}_j$ $(j = 1, \ldots, N_{\text{sim}})$)

$$\mathrm{E}[S[g(\boldsymbol{X})]] \approx \frac{1}{N_{\text{sim}}} \sum_{i=1}^{N_{\text{sim}}} S[g(\boldsymbol{x}_i)] \tag{6}$$

where the sampling points $\boldsymbol{x}_i = \{x_{i,1}, \ldots, x_{i,v}, \ldots, x_{i,N_{\text{var}}}\}$ are selected using the transformation in Eq. (4), i.e. $x_{i,v} = F_v^{-1}(u_{i,v})$, in which we assume that each of the N_{sim} sampling points $\boldsymbol{u}_i$ $(i = 1, \ldots, N_{\text{sim}})$ were selected with the same probability of $1/N_{\text{sim}}$. Violation of the uniformity of the distribution of points $\boldsymbol{u}_i$ in the unit hypercube may lead to erroneous estimations of the integrals.

If the sampling points were not selected with respect to equal probabilities in the design domain, the possibility to improve the accuracy in Eq. (6) is to use weights different from $1/N_{\text{sim}}$. These weights reflect the probability content of the cells around individual sampling points

$$\mathrm{E}[S[g(\boldsymbol{X})]] \approx \frac{1}{W} \sum_{i=1}^{N_{\text{sim}}} S[g(\boldsymbol{x}_i)] \cdot w_i \tag{7}$$

where $W = \sum_{i=1}^{N_{\text{sim}}} w_i$ is the sum of weights for N_{sim} points (normalization). The proposed approach aims at finding appropriate weights that are calculated considering the spatial distribution of the points. In Monte Carlo sampling, for example, the sampling distribution may correspond to the joint distribution function (CDF) of the random vector, but the sampling strategy is so inefficient that the sample of N_{sim} point does not reproduce the CDF well. Reweighting of samples based on a true distribution of points seems to be a way to improve the accuracy. Obviously, unvisited regions of the design domain can not be explored by a nonuniform design.

In any case, partitioning the space into cells around the given sampling points may help to (i) reduce probabilities associated with points that are participating in clusters of points and, at the same time, (ii) identification of unexplored regions may help in adjusting the weights of the existing samples with respect to the volumes of regions they occupy, (iii) the identified unexplored regions can be used for sample size extension by new points in which the function can be additionally evaluated, if possible.

Voronoi tessellation has been selected for partitioning of the design space into volumes that are used as the weights w_i, $i = 1, \ldots, N_{\text{sim}}$. The following section describes the Voronoi tessellation procedures.

3 WEIGHTS OBTAINED AS VOLUMES OF VORONOI REGIONS

The weights associated with the design points are considered as volumes of Voronoi regions [2] computed on the sampling points. The Voronoi tessellation in N_{var}-dimensional space results in N_{sim} convex polyhedrons $\mathscr{V}_i$ that enclose all the points that are closer to i-th sampling point than any other. Defining the distance of point u from sampling point u_i as $d_i(u)$, the Voronoi region associated with i-th sampling point can be formally defined as

$$\mathscr{V}_i = \left\{ u \in \mathbb{R}^{N_{\text{var}}} \mid \forall\ j \neq i : d_i(u) \leq d_j(u) \right\} \tag{8}$$

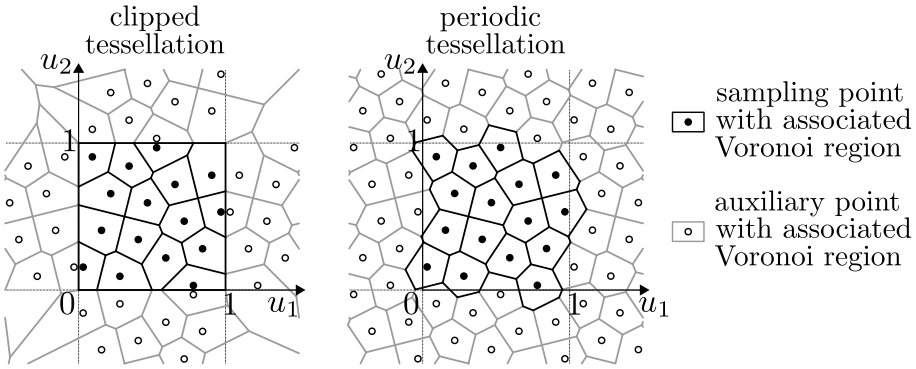

Figure 1: Example of clipped and periodic tessellation for $N_{\text{var}} = 2$ and $N_{\text{sim}} = 16$ with help of reflected and periodically repeated auxiliary points, respectively

Two alternatives of Voronoi tessellation that differ in the boundary regions are investigated:

- *clipped* Voronoi tessellation that is limited to the unit hypercube only

$$\mathscr{V}_i = \left\{ u \in \langle 0, 1 \rangle^{N_{\text{var}}} \mid \forall\ j \neq i : d_i(u) \leq d_j(u) \right\} \tag{9}$$

- *periodic* Voronoi tessellation which assumes that every sampling point is periodically repeated in the space along all the dimensions.

These two different concepts are demonstrated in Fig. 1. The reason for studying the *periodic* tessellation is that the authors have shown recently [5, 11] that the presence of boundaries in the hypercubical design domain cause problems. Briefly, one may think of a problem of packing (hyper)balls into a (hyper)cube. It is clear that the boundary is responsible for a kind of wall-effect. It has been shown [5, 11] that this problem can be removed by considering periodic extension of the design domain. The balls then permeate through the boundaries without interacting with them, see Fig. 1 right.

The *clipped* Voronoi diagrams [4, 14] are used mostly for construction of meshes and therefore available software to compute such tessellation is limited to two and three dimensional space. A similar situation exists for *periodic* Voronoi tessellation [13, 10]. In the field of design of experiments more than three variables (factors) can be present and therefore the tessellation must be performed in higher dimensions. In this contribution, Qhull software [3] is utilized for both clipped and periodic tessellations because it can compute Voronoi tessellation for arbitrary dimension. On the other hand, it cannot work directly with neither *clipped* nor *periodic* boundary condition and therefore simple tricks are used.

These tricks consist in manipulations of the design domain (together with the sampling points contained) by adding new design domains around it. In order to obtain the *clipped* structure,

the design domain is extended by reflecting the original design domain along each dimension. There are two reflections of the original unit interval along each dimension to obtain intervals $\langle -1, 0\rangle$ and $\langle 1, 2\rangle$. Therefore, the tessellation is performed on $N_{\text{sim}}\left(1+2^{N_{\text{var}}}\right)$ points. The use of reflection automatically provides edges between cells that coincide with the boundary of the original design domain and therefore the volumes outside the design domain can be ignored. The use of reflection to obtain clipped tessellation was proposed in [9].

The *periodic* structure is obtained by periodic extension (replication) of the original design domain along each direction and additionally the replication must be performed to obtain all the "corner" domains to fill a hypercube $\langle -1, 0\rangle^{N_{\text{var}}}$. Therefore, $N_{\text{sim}} \cdot 3^{N_{\text{var}}}$ points in total are used for the periodic tessellation.

The computational times needed for the both tessellation types can be substantially reduced if it involves only reflected or periodically repeated points that are close to the original hypercube, because only these points affects the tessellation inside the hypercube. Unfortunately, no effective algorithm has been developed yet to identify such points and therefore the full set of points must be involved for certainty.

In both alternatives, the weights for individual sampling points are the volumes of regions surrounding points. There are three algorithms available for the volume computation: (i) direct integration, (ii) Monte-Carlo integration and (iii) division into simplexes for which analytical formula is available. The first two algorithms are nicely elucidated in [8]. Here, we perform the third algorithm. Each Voronoi region is (with a help of the Qhull) divided into simplexes. Each simplex has $N_{\text{var}}+1$ vertices denoted v_i. The total volume of the region is simply the sum of simplex volumes, that are calculated based on the determinant of coordinate matrix.

$$V_{\text{simplex}} = \left| \frac{1}{N_{\text{var}}!} \begin{pmatrix} v_1 - v_0 \\ v_2 - v_0 \\ \vdots \\ v_{N_{\text{var}}} - v_0 \end{pmatrix} \right| \tag{10}$$

These volumes are used directly as weights of sampling points enclosed withing these cells.

4 FREQUENCY ANALYSIS OF WEIGHTS

It turns out to be important to see (i) whether the weights are very scattered compared to $1/N_{\text{sim}}$ and, (ii) whether their magnitude tend to depend on the position inside the domain. This is achieved by studying $N_{\text{run}} = 1000$ realizations of samples, each having N_{sim} points within an N_{var}-dimensional hypercube. For each sample, both types of Voronoi tessellation is constructed and the weights are statistically processed.

The results will be presented for two sampling schemes: the classical (crude) Monte Carlo sampling without any optimization (MC-RAND) and LHS (Latin Hypercube Sampling) optimized using the periodic criterion (LHS-PAE). PAE stands for an enhanced version of the Audze-Eglais criterion, see [5, 11].

Figure 2 shows one sample (N_{sim}=16) of a bivariate random vector U_v for both sampling schemes. For the two sampling schemes, both types of Voronoi diagrams (*clipped* and *periodic*) are constructed and visualized with colors depending on the area. The LHS-PAE sampling plans show more uniform distribution of points because the PAE-optimized LH-sampling better avoids clustering and limit the occurrence of empty regions. Therefore, the cells in LHS-PAE have similar volumes and the sampling points are closer to the centers of Voronoi regions. The small differences among weights in LHS-PAE with *periodic* tessellation suggest that weight-

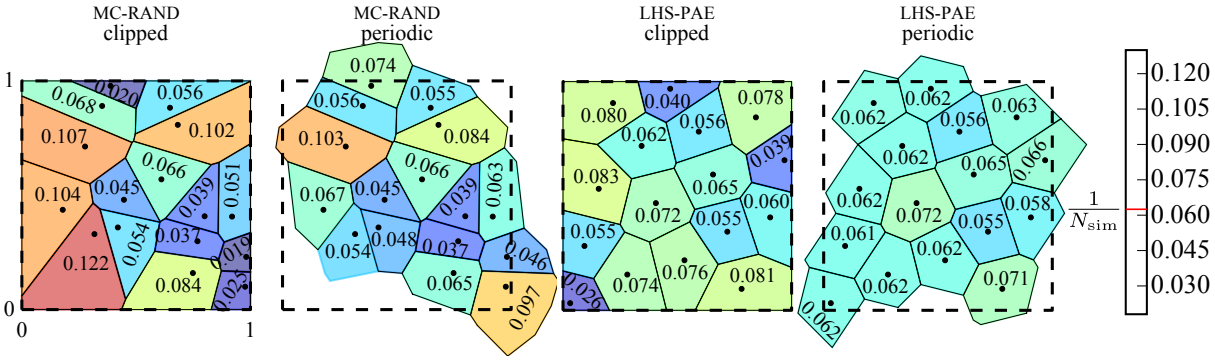

Figure 2: Voronoi weights for MC-RAND and LHS-PAE sampling plans ($N_{\mathrm{var}} = 2$, $N_{\mathrm{sim}} = 16$). Comparison of the *clipped* and *periodic* tessellations.

ing will not make much difference in comparison with integrals evaluated using equal weights $1/N_{\mathrm{sim}}$. The MC-RAND sampling plans suffer from point clustering and therefore, high variability in volumes of the Voronoi cells is observed. It should be noticed that the choice of tessellation (*clipped* vs. *periodic*) affects only the boundary regions while the central part of the hypercube is identical.

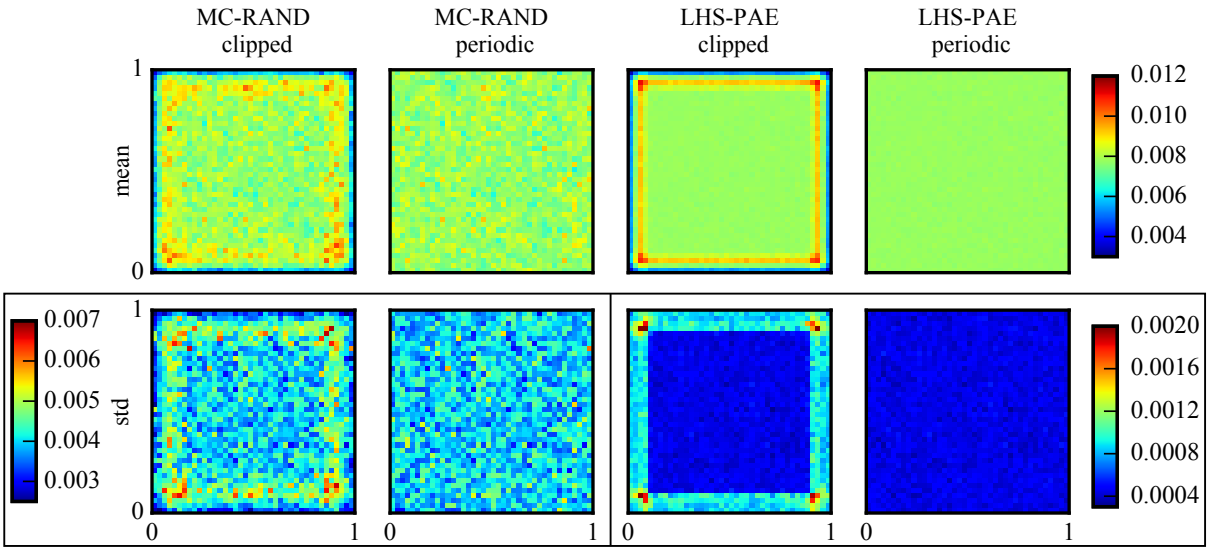

Figure 3: Bivariate histograms of the mean value and the standard deviation of cell volumes for both sampling plans and both tessellation alternatives ($N_{\mathrm{var}} = 2$, $N_{\mathrm{sim}} = 16$ and $N_{\mathrm{run}} = 1000$ realizations).

In order to judge about the spatial distribution of weights within the design domain, the above-mentioned N_{run}=1000 realizations of samples accompanied by Voronoi tessellations were prepared and for each spatial location, the mean value and standard deviation of weights occurring at that location have been calculated. The weights (volumes of Voronoi regions V_{simplex} in a hypercube) depend on the type of tessellation but they are independent of the sampling method (MC vs. LHS). The bivariate histograms in Fig. 3 document the dependency of the mean value and the standard deviation of Voronoi region volumes on the position of the sampling point in a square. In the case of *clipped* tessellation, both the mean value and the standard deviation of weights are not uniform in the hypercube. Three zones can be distinguished: (a) the boundary region where the mean value (shown in blue) of weights is underestimated. The boundary strip is followed/balanced by (b) zone parallel to the boundary where the weights are overestimated

(see the yellow to red color) and finally, (c) the bulk zone sufficiently far from the boundary, where the weights (volumes) are constant on average. The width of the two boundary zones is decreasing with increasing sample size N_{sim}.

Such a biased representation of different regions in the hypercube partitioned by the *clipped* tessellation has consequences in Monte Carlo integration. If the points are sampled uniformly, and that is indeed the case of both MC-RAND and LHS-PAE, errors are introduced due to introduction of nonuniform weighting. If the functions are sensitive to inaccuracies in representation of the boundary regions, their weighted MC integration may yield biased results. Therefore we conclude that the *clipped* tessellation generally should not be used for weighting in MC integration.

The *periodic* tessellation provides more promising bivariate histograms: no bias around the boundaries is visible for both MC-RAND and LHS-PAE sampling schemes. The statistics of the weights do not depend systematically on the position in the hypercube.

Numerical simulations focused on MC integration presented in Fig. 4 of our recent paper [12] revealed that reweighting the samples according to the volumes of the Voronoi cells in periodic space does not significantly improve the accuracy estimators compared to the standard estimations using equal weights $1/N_{\text{sim}}$. The conclusion was that the gain in accuracy was not worth the effort. Anyway, the periodicity of the Voronoi tessellation was found an important ingredient that guaranties that the Voronoi reweighting does not systematically bias the results.

Fig. 2 shows that good designs such as those obtained by LHS-PAE do not need any reweighting at all as the points have quite regular distribution in the the design domain. MC-RAND designs, on the other hand, can lead to Voronoi cells of quite different volumes. Unfortunately, the tessellation proposed so far does not deal with neither *clusters of points* nor *large empty spaces*.

5 ADAPTIVE REFINEMENT BY POINT CLUSTERING & IDENTIFICATION OF EMPTY REGIONS

In this section we present an enhanced weighting algorithm that is able to improve tessellation of a given point layout by an adaptive sequence of two different kinds of steps: (a) *grouping* of a pair of points that form a cluster (occupy relatively small region) and, (b) *insertion of dummy points* withing regions that are not occupied by the original sampling points. The dummy points help the tessellation to identify volumes of regions that are not represented by any point and thus the volumes must be subtracted form the total unit volume.

The process is performed by individual *steps* of either *point grouping* or *point insertion* and the decision of which step is taken is driven by given rules. The sequence of steps may evolve differently in various designs depending on the particular point layout. At every stage, the key number in the decision of what step to take is the *characteristic length* that we define as:

$$l_{\text{char}} = \frac{1}{\sqrt[N_{\text{var}}]{N_{\text{p}}}} \tag{11}$$

where N_{p} is the current number of points (including the dummy points and without the redundant points in clusters). The formula makes l_{char} the average distance to the nearest neighbor in and "ideal" design. The rationale behind this definition of l_{char} is a regular orthogonal grid of $N_{\text{p}} = N^{N_{\text{var}}}$ points in the unit hypercube, where the distance to the nearest neighbor is measured along any dimension and reads $1/N$. The number of points N_{p} is initially equal to the number of points in the design, N_{sim}, and after *insertion* of a new dummy point is increased by one and

after *grouping* of two points is decreased by one. After any of these steps is taken, the list of points and the value of l_{char} must be updated.

The characteristic length serves for comparison with two distances of the current design stage, see Fig. 4:

- the smallest inter-point distance, l_{min}, (measured in the periodic space), and
- the diameter of the largest empty N_{var}-dimensional hypersphere, l_{max}, (again in the periodic design domain).

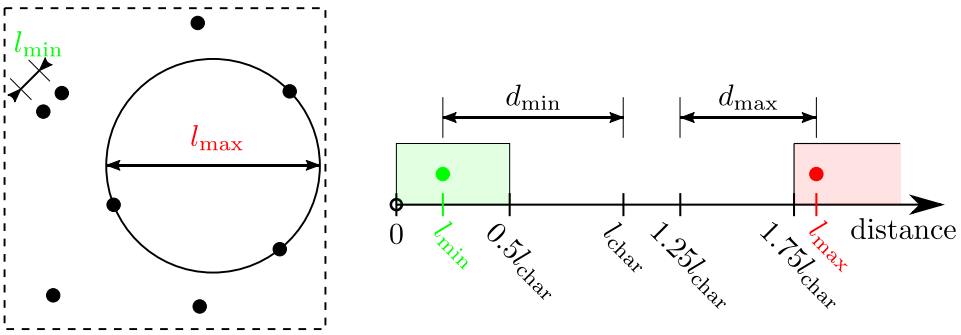

Figure 4: Left: Distances and regions used in conditions of an adaptive algorithm. Right: example design for $N_{\text{sim}} = 8$ points with dimensions of the smallest inter-point distance l_{min} and the largest incircle diameter l_{max}. This configuration in the first step will lead to grouping – insertion of the point in the middle of l_{min}.

These lengths are also calculated before every decision whether to perform the point *grouping* step or *insertion* step. With these lengths at hand, two parameters are calculated, see Fig. 4:

$$d_{\text{min}} = |l_{\text{min}} - l_{\text{char}}| \quad , \qquad d_{\text{max}} = |l_{\text{max}} - 1.25 l_{\text{char}}| \tag{12}$$

If $d_{\text{min}} > d_{\text{max}}$ the algorithm prefers *grouping* of the closest pair of points with the distance l_{min} by placing the new point in the centroid of the cluster. Otherwise ($d_{\text{min}} \leq d_{\text{max}}$), the algorithm prefers *insertion* of one dummy point into the center of the largest empty hypersphere of diameter l_{max}.

In any of the two cases, an *additional condition* have to be met prior to performing any of the two steps. For *point grouping* to be performed, the condition $l_{\text{min}} < 0.5 l_{\text{char}}$ must be fulfilled, otherwise the algorithm terminates the adaptive process. Similarly, *point insertion* is made only if the sphere diameter satisfies: $l_{\text{max}} > 1.75 l_{\text{char}}$, otherwise the algorithm terminates.

Regarding the grouping step, we mention that any of the two grouped points may already represent a cluster of points. Therefore the points have weights equal to the number of point they represent already (initially they have unit weights). This weighting guarantees that if a cluster of more than two original points occurs, the inserted point after grouping is in the centroid of the whole cluster. The additional conditions serve for halting the process and the constants are tuned manually such that the algorithm leads to quite uniform distribution of Voronoi cells with points relatively close to the centroids of the Voronoi cells (the points are good repentants of their cells). Three kinds of points are available at the end of the adaptive process:

- the original sampling points that have not been clustered (see the black solid circles in Fig. 5). They are associated with weights according to the volumes of Voronoi cells they occupy.
- the centroids of clusters (see the green solid circles in Fig. 5). The original points within a single cluster have share weights equal to the volume of the corresponding cell divided by the number points inside it.

- the dummy points (see the red solid circles in Fig. 5). They have zero weights as they do not represent any original point. These points will be considered as good candidates for sample size extension.

Fig. 5 shows examples of Voronoi diagrams obtained with the adaptive algorithm, applied to initial designs obtained by four different techniques. In the same figure, they are compared with the basic periodic Voronoi diagrams as described in Sec. 3, see the top row. The weights of points are represented by the darkness of blue color. Green points represent clusters of the original (white) points.

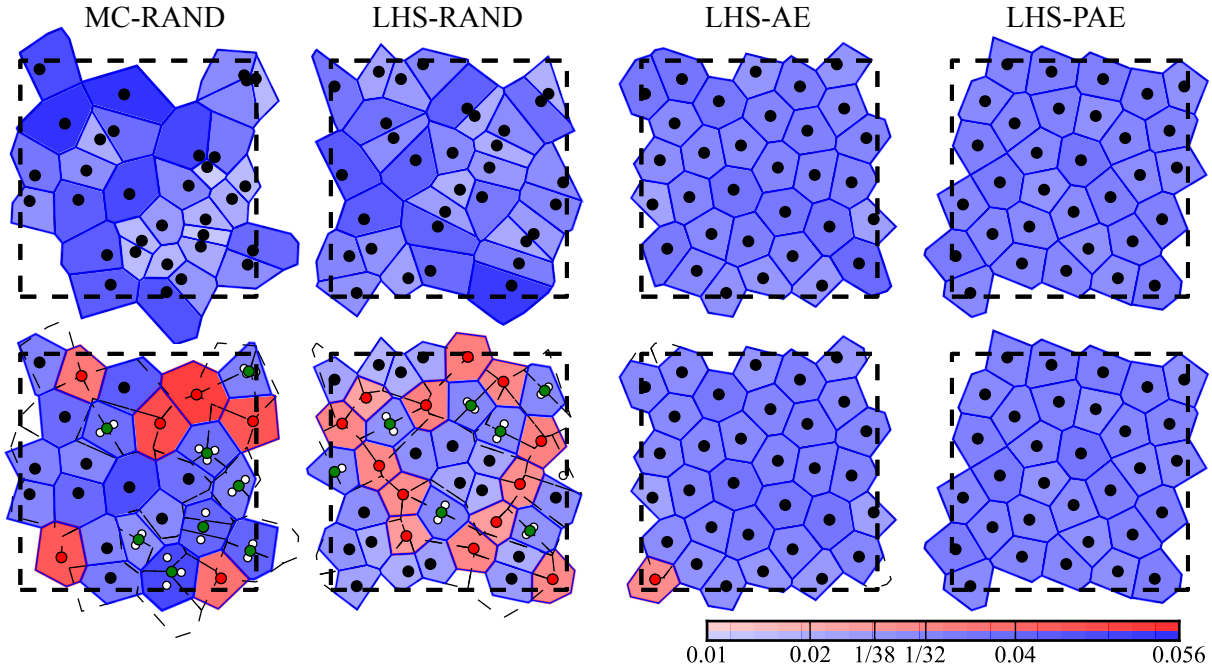

Figure 5: Examples of Voronoi tessellations of various designs with the initial sample size of $N_{\text{sim}} = 32$ points. Top row: periodic tessellation. Bottom row: adaptive tessellation with identified clusters and the "dummy" points.

In the basic setting of the adaptive algorithm, the red points are dummy points and the associated red areas are excluded. In other words, the existence of the red areas signalize that the sum of weights (volumes), W, is less than one. The adaptive algorithm basically redistribute the total unit volume among the cells (including the red ones) such that the volumes are almost identical and the points tend to be close to the centroids of the cells. The fact that the blue regions have similar volumes means that the weights among samples are similar, approximately $1/N_{\text{p}}$.

One of the outcomes of the proposed adaptive algorithm is that the points inside the red areas can be used as candidate points for *sample size extension*. They are placed approximately in the centers of the largest empty hypersphere's and therefore they are points for the design refinement. The next section studies two aspects of the proposed adaptive Voronoi weighting, namely

(i) whether the reweighting of existing N_{sim} points leads to improvement in Monte Carlo integration (blue regions in Fig. 5 bottom are considered), and

(ii) how much gain in accuracy and variance reduction is achieved by extending the sample with the proposed "dummy points", evaluating the studied function and considering them in addition to the existing N_{sim} points with the adaptive Voronoi weights (both the blue and the red regions in Fig. 5 bottom are considered).

6 NUMERICAL EXAMPLES OF MC INTEGRATION & DISCUSSION

This section studies whether weighting in MC integrals based on the Voronoi tessellation improves the quality of the estimates. Three basic transformations $g(\boldsymbol{X})$ of standard independent Gaussian random variables X_v, $v = 1,\ldots,N_{\text{var}}$ have been selected for the numerical study. The following equation array presents formulas of the three functions (first column), the analytical solutions for the mean values (second column) and the standard deviations (third column):

$$Z_{\text{sum}} = g_{\text{sum}}(\boldsymbol{X}) = \sum_{v=1}^{N_{\text{var}}} X_v \qquad \mu_{\text{sum}} = 0 \qquad \sigma_{\text{sum}} = \sqrt{2} \tag{13}$$

$$Z_{\text{exp}} = g_{\text{exp}}(\boldsymbol{X}) = \sum_{v=1}^{N_{\text{var}}} \exp(-X_v^2) \qquad \mu_{\text{exp}} = \frac{\sqrt{3}}{3} N_{\text{var}} \qquad \sigma_{\text{exp}} = \sqrt{N_{\text{var}}}\sqrt{\frac{\sqrt{5}}{5} - \frac{1}{3}} \tag{14}$$

$$Z_{\text{prod}} = g_{\text{prod}}(\boldsymbol{X}) = \prod_{v=1}^{N_{\text{var}}} X_v \qquad \mu_{\text{prod}} = 0 \qquad \sigma_{\text{prod}} = 1 \tag{15}$$

Two unoptimized sampling schemes, MC-RAND and LHS-RAND, have been used to prepare $N_{\text{run}} = 1000$ sampling plans for various sample sizes N_{sim} ranging from 2 to 512. Only bivariate cases are studied: $N_{\text{var}} = 2$.

The performance of the approaches will be demonstrated by showing their ability to estimate the mean value and standard deviation of the transformed variable $Z = g(\boldsymbol{X})$. The estimated mean value and standard deviation are denoted as $\bar{\mu}_Z$ and $\bar{\sigma}_Z$, respectively, and can be estimated in the spirit of Eq (7) as:

$$\bar{\mu}_Z = \frac{1}{W} \sum_{i=1}^{N_{\text{sim}}} g(\boldsymbol{x}_i) \cdot w_i \tag{16}$$

$$(\bar{\sigma}_Z)^2 = \frac{1}{W} \sum_{i=1}^{N_{\text{sim}}} \left[g(\boldsymbol{x}_i) - \bar{\mu}_Z\right]^2 \cdot w_i \tag{17}$$

where $W = \sum_{i=1}^{N_{\text{sim}}} w_i$ is the sum of weights for N_{sim} points.

Three approaches to the weighting in Monte-Carlo type numerical integration were studied:

1. *uniform* weights that assign each design point a constant weight $w_i = 1/N_{\text{sim}}$,
2. *V*oronoi weights from *periodic* tessellation. Weights are equal to the "volumes" of cells obtained by the *adaptive* algorithm ($W \leq 1$)
3. *V*oronoi weights from *periodic* tessellation. Weights are equal to the "volumes" of cells obtained by the *adaptive* algorithm ($W = 1$) and considering the additional samples (the red regions).

We only used the *periodic* tessellations as the *clipped* tessellations provide systematically wrong results [12].

The results of numerical study are presented in Fig. 6 for all three functions, two estimated statistical moments, two sampling schemes and two alternatives of weighting. The second alternative is not presented because the adaptive algorithm without the additional points leads to approximately equal weights $1/N_{\text{p}}$ for all N_{sim} samples and thus has no effect in Eq. (17). The results are almost identical with those obtained with weight equal to $1/N_{\text{sim}}$ (approach 1).

The third technique improves the accuracy of average estimates and it also significantly decreases the variance of the estimator compared to alternative standard equal weighting (alternative 1). This improvement comes at a price of additional effort associated with the new

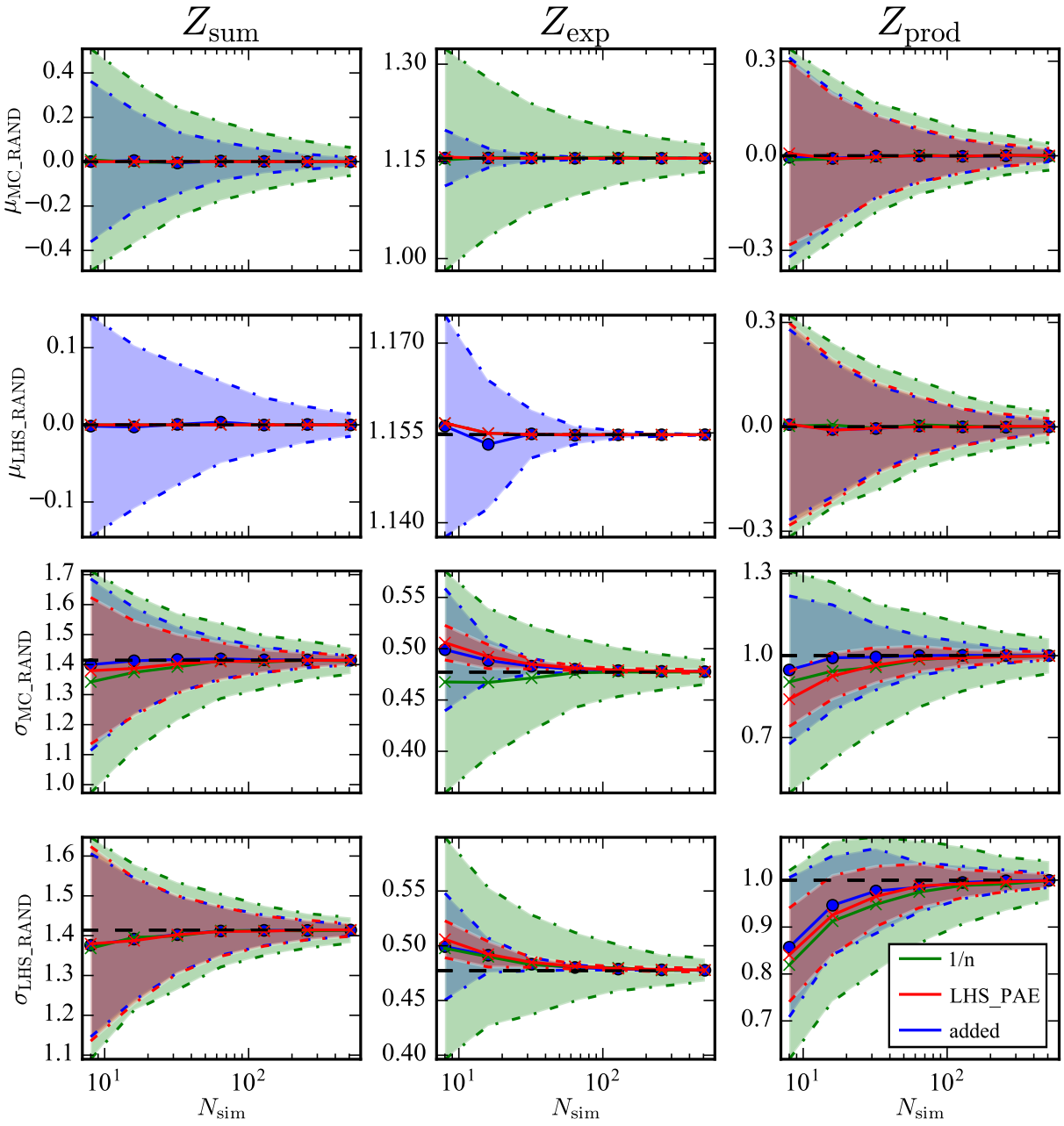

Figure 6: Convergence of the estimated mean values and standard deviations of the three transformed variables Z_{sum}, Z_{exp} and Z_{prod}, computed for $N_{\mathrm{var}} = 2$. Two weighting alternatives are compared for two initial sampling schemes: MC-RAND and LHS-RAND. LHS-PAE is used as a reference technique.

samples. We have found that the number of additional samples is such that for small designs is around 20% of N_{sim} and this percentage increases to approximately 33% for large sample sizes. In order to show how much gain for the additional sample size is obtained, we have compared the results with those obtained by employing LHS-PAE designs.

Each alternative is represented by a solid line showing the average estimations $\pm$ one sample standard deviation (a scatter-band shown by the shaded area); both computed using the $N_{\mathrm{run}} = 1000$ realizations.

The mean values obtained by LHS-schemes for Z_{sum} and Z_{exp} has no scatter at all. This is due to the fact that for these additive functions, the LH-samples are transformed to the exact

mean value of each separate variable as these variables are sampled regularly with respect to probabilities. The pairing does not matter. Therefore, the estimations using LH-designs can not improved by adding new points.

In most cases, however, the standard equal weights $1/N_{\text{sim}}$ provide the largest variance of the estimators; employing LHS instead of MC helps to reduce the variance; and the best results are obtained with LHS-PAE. The accuracy of existing LHS and MC samples is always improved by using the extended sample size with weighting obtained using the proposed adaptive algorithm.

7 CONCLUSIONS

Various alternatives of Voronoi tessellation was studied in an attempt to improve the accuracy of Monte-Carlo integration schemes for small N_{sim} by weighting individual sampling points. The weights were obtained as volumes of the Voronoi cells – the regions surrounding the sampling points in the design domain (unit hypercube).

Weighting using the *clipped* Voronoi tessellation (a tessellation limited to the design domain) was found inapplicable due to problems related to the presence of boundaries of the unit hypercube. The tessellation results in systematic appearance of underestimated regions near the boundaries followed by regions with over-weighted regions. The *periodic* tessellation removes the systematic bias.

Pure reweighting the sample using Voronoi weights does not help much to increase the accuracy. The proposed adaptive algorithm that identify and remove unvisited regions does not help much despite its ability to find clusters of points and decrease their influence accordingly.

The proposed adaptive algorithm, however, seems to drastically improve the results by adding up to 33% of additional point to the design. Results obtained initially with unoptimized designs seems to improve significantly at the price of the additional evaluations at the new points. The performance of a such an extended sample is almost as good as performance obtained with designs that were optimized in advance. Therefore, the proposed technique can be viewed as a sophisticated sample size extension technique.

ACKNOWLEDGEMENT

This work has been supported by the Grant agency of the Czech Republic under projects Nos. GA16-22230S and GA15-07730S. This support is gratefully acknowledged.

REFERENCES

[1] P. Audze and V. Eglājs. New approach for planning out of experiments. *Problems of Dynamics and Strengths*, **35**:104–107, 1977. (in Russian).

[2] F. Aurenhammer. Voronoi diagrams–a survey of a fundamental geometric data structure. *ACM Computing Surveys*, **23**(3):345–405, 1991.

[3] C. Bradford Barber, D.P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical software*, **22**(4):469–483, 1996.

[4] T.M.Y. Chan, J. Snoeyink, and C.-K. Yap. Output-sensitive construction of polytopes in four dimensions and clipped voronoi diagrams in three. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '95, pages 282–291, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.

[5] J. Eliáš and M. Vořechovský. Modification of the audzeeglajs criterion to achieve a uniform distribution of sampling points. *Advances in Engineering Software*, **100**:82–96, 2016.

[6] M.E. Johnson, L.M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, **2**(26):131–148, 1990.

[7] R.B. Nelsen. *An Introduction to Copulas*, volume **XIV** of *Springer Series in Statistics*. Springer, 2nd edition, 2006. Originally published as volume 139 in the series "Lecture Notes Statistics".

[8] H.L. Ong, H.C. Huang, and W.M. Huin. Finding the exact volume of a polyhedron. *Advances in Engineering Software*, **34**(6):351–356, 2003.

[9] L. Pronzato and W.G. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, **22**(3):681–701, 2012.

[10] C.H. Rycroft. Voro++: A three-dimensional voronoi cell library in c++. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **19**(4):041111, 2009.

[11] M. Vořechovský and J. Eliáš. Improved formulation of Audze–Eglājs criterion for space-filling designs. In *Proc. of 12th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP*, 2015.

[12] M. Vořechovský, V. Sadílek, and J. Eliáš. Application of voronoi weights in monte carlo integration with a given sampling plan. In S. Freitag, R.L. Muhanna, and R.L. Mullen, editors, *REC 2016 the 7th International Workshop on Reliable Engineering Computing*, pages 441–452, 2016.

[13] D.-M. Yan, K. Wang, B. Lévy, and L. Alonso. Computing 2D periodic centroidal voronoi tessellation. In *8th International Symposium on Voronoi Diagrams in Science and Engineering*, 2011.

[14] D.-M. Yan, W. Wang, B. Lévy, and Y. Liu. Efficient computation of clipped voronoi diagram for mesh generation. *Computer-Aided Design*, **45**(4):843–852, 2013. Geometric Modeling and Processing 2010.