Journal of Civil Engineering and Management
ISSN 1392-3730 / eISSN 1822-3605
2018 Volume 24 Issue 5: 410–423
https://doi.org/10.3846/jcem.2018.5189

EVALUATION OF PAIRWISE DISTANCES AMONG POINTS FORMING A REGULAR ORTHOGONAL GRID IN A HYPERCUBE

Václav SADÍLEK, Miroslav VOŘECHOVSKÝ*

Institute of Structural Mechanics, Faculty of Civil Engineering, Brno University of Technology, Czech Republic

Received 22 March 2018; accepted 19 June 2018

Abstract. Cartesian grid is a basic arrangement of points that form a regular orthogonal grid (ROG). In some applications, it is needed to evaluate all pairwise distances among ROG points. This paper focuses on ROG discretization of a unit hypercube of arbitrary dimension. A method for the fast enumeration of all pairwise distances and their counts for a high number of points arranged into high-dimensional ROG is presented. The proposed method exploits the regular and collapsible pattern of ROG to reduce the number of evaluated distances. The number of unique distances is identified and frequencies are determined using combinatorial rules. The measured computational speed-up compared to a naïve approach corresponds to the presented theoretical analysis. The proposed method and algorithm may find applications in various fields. The paper shows application focused on the behaviour of various distance measures with the motivation to find the lower bounds on the criteria of point distribution uniformity in Monte Carlo integration.

Keywords: full factorial design, design of experiments, pairwise distances, Audze-Eglājs criterion, optimization, periodic space.

Introduction

The measuring of distances is fundamental to geospatial analysis (Tobler 1970). It is closely related to the concept of route. For example, the package *gdistance* (van Etten 2012) was designed to determine grid-based distances and routes, and to be used in combination with other packages available within R (R Core Team 2016). Another application of distance distributions between points in spatial processes can be found when modelling connectivity in wireless mobile systems such as cellular, ad-hoc and sensor networks. Since connectivity can be expressed as a function of the distance between nodes, distance distributions between points in spatial processes are of special importance (Moltchanov 2012).

The study of point distribution is also very relevant to the field of design of experiments (DoE) and statistical sampling for Monte Carlo integration. The present paper primarily concerns the distribution of points relevant to statistical sampling for computer experiments (Iman, Conover 1980; Morris, Mitchell 1995), where the optimal placement of sampling points is an as yet unsolved problem.

In computer experimentation (Sacks *et al.* 1989), which is a powerful tool for the investigation of problems encompassing the randomness of observed phenomena, the task is to prepare a plan of the simulations that should be performed, i.e. what is known as the Design of Experiments (DoE) should be carried out. A similar task is included in the *response surface method* first introduced by Box (1954) and since then improved and adapted in various ways (e.g. Bucher, Bourgund 1990; Gupta, Manohar 2004; Hamzah *et al.* 2017), where training points have to be placed appropriately.

In Monte Carlo integration it is desirable to minimize the number of simulations (point count N_{sim}) while making sure statistical estimates remain of high quality. This is achieved by the uniform filling of the design domain, which is a unit N_{var}-dimensional hypercube (N_{var} is the number of input random variables), and the appropriate transformation of points of such a *sampling plan* in accordance with the required probability distribution and mutual dependencies among the inputs. Individual simulations are then represented by design points placed within the hypercube. Designs for the placement of points inside the design domain that supposedly distribute the points uniformly are known as space filling designs (see e.g. Damblin *et al.* 2013).

*Corresponding author. E-mail: *vorechovsky.m@vutbr.cz*

Designs with a general N_{sim} and N_{var} that are supposed to fill the hypercube uniformly are usually sought for via heuristic optimization (Morris, Mitchell 1995; Mortazavi *et al.* 2017; Siddiquea, Adelib 2016; Vořechovský, Novák 2009). There are many different criteria that are designed to measure this uniformity and are subject to minimization during such optimization. One of the criteria that accentuate the space-filling property of the final design is the Audze-Eglãjs (AE) criterion (Audze, Eglãjs 1977). Another example is the *periodic* version of the AE criterion, hereinafter referred to as the PAE criterion (Eliáš, Vořechovský 2016), which removes a major flaw of the AE criterion. Many other criteria have been defined with the aim of ensuring the uniform filling of a given space. They are often based on the evaluation of discrepancy, e.g. Centered L_2-discrepancy (Fang, Ma 2001), Wrap-Around L_2-discrepancy (Fang, Ma 2001). Maximin or miniMax criteria (Husslage 2006; Johnson *et al.* 1990) may also be used. The criteria used for the illustration of application in this article are the Audze-Eglãjs (AE) criterion (Audze, Eglãjs 1977; Bates *et al.* 2003), or its generalization into the ϕ criterion (Morris, Mitchell 1995) and the PAE criterion (Eliáš, Vořechovský 2016).

The effective control of an optimization algorithm is often conditional upon the knowledge of the lower bound of the optimization criterion (the minimum that can be reached during optimization) and the mean value. Various means of optimization exist. For example, heuristic algorithms based on columnwise-pairwise exchanges (the shuffling of pre-sampled coordinates) (Li, Wu 1997; Morris, Mitchell 1995; Vořechovský, Novák 2009; Chen *et al.* 2016; Huang *et al.* 2016) can drastically profit from the knowledge of the minimum (or maximum in the case of maximization) of the objective function (e.g. the AE criterion).

The available methods of filling a unit hypercube with points are basically divided into two types: deterministic and stochastic. Historically, non-random lattices (such as ROGs, triangular lattices or hexagonal lattices) have also been used as models for a number of physical and environmental phenomena, e.g. elements of crystals, the placement of seedlings in the landscape, the locations of facilities in cities, etc. (Chu 2006). Therefore, the evaluation of distances in regular lattices may be important in these fields of expertise as well. Deterministic methods place the design points in regular patterns and may ensure perfect space-filling. One such example is full factorial design (Chudoba *et al.* 2013; Montgomery 2006). It is a basic design that, in repeatable computer experiments, has only one representant for each location and explores combinations of all factors with all levels for that factor.

In this paper, we consider such a Cartesian grid (regular orthogonal grid) of N_{sim} points in a *design domain* that is a unit hypercube of dimension N_{var} (sometimes denoted as $[0,1]^{N_{\text{var}}}$). In particular, we study the pairwise distances among all pairs of these N_{sim} points. The list of the distances featured in such a design is needed in the evaluation of many criteria concerning the optimality of that design (Maximin, miniMax, AE, PAE, ϕ criterion) (Audze, Eglãjs 1977; Bates *et al.* 2003; Eliáš, Vořechovský 2016; Husslage 2006; Johnson *et al.* 1990; Morris, Mitchell 1995). Due to the regularity and perfect space-filling of certain deterministic designs it is reasonable to expect these designs to provide a criterion value that is the minimum or close to the minimum for a given configuration $(N_{\text{sim}}, N_{\text{var}})$. The ROG designs studied in this paper are presumed to be optimal or near-optimal designs as regards many criteria related to uniformity, regularity, discrepancy and space-fillingness. It is supposed that they provide conservative estimates of the lower bounds on the design optimality criteria.

Since these deterministic designs are restricted just to specific numbers of design points ($N_{\text{sim}} = N^{N_{\text{var}}}$, N being a natural number expressing the number of different input values for each random input variable), the article also anticipates the possibility of interpolation of the lower bound for an arbitrary N_{sim} of a stochastic design.

This paper presents a method for the fast enumeration of all pairwise distances among points arranged into a regular orthogonal grid (ROG) in a unit hypercube and is organized as follows. Section 1 describes the ROG arrangement of points. Section 2 provides an analysis of distances in ROG and identifies main transformations of simple distances identified by simple combinatorial operations. Section 3 presents the speed-up compared to naïve evaluation of distances. Section 4 presents an application of the

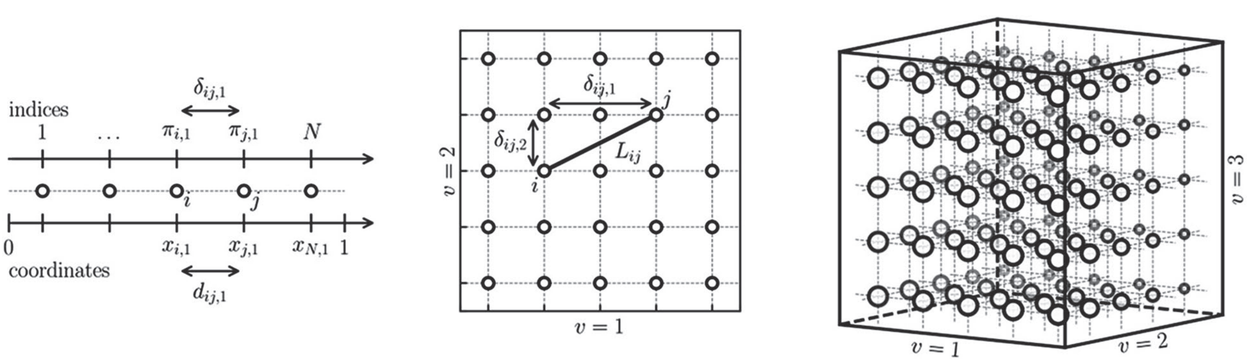

Figure 1. Regular grids of five points ($N = 5$) in unit cubes of $N_{\text{var}} = 1, 2$ and 3 dimensions

method in determining the bounds of optimality criterion and design of computer experiments.

1. Unit hypercube filled with regular orthogonal grid of points

A Grid or Mesh is defined as smaller shapes formed after the discretization of a geometric domain. A *regular grid* is defined as a tessellation of N_{var} -dimensional Euclidean space by congruent parallelotopes (e.g. bricks). Several types of gridding systems exist, e.g. hexagonal grids or triangular grids generalized to arbitrary dimensions. This paper considers a rectangular domain (a unit hypercube) and its division into a regular Cartesian grid. A Cartesian grid (ROG) of points is a special case where the elements are equal lines, squares, cubes, etc.

We consider a ROG of points arranged in the unit hypercube arranged such that the number of points reads:

$$N_{\text{sim}} = N^{N_{\text{var}}}, \tag{1}$$

where N is the number of equidistant coordinates along each individual dimension.

It corresponds to a tessellation of a unit hypercube into N_{sim} identical cubes; the points are placed in the intersections of lines passing through their centroids. In this arrangement, the points form an orthogonal grid inside a unit hypercube the dimension of which is N_{var}. Examples of such hypercubes are a line ($N_{\text{var}} = 1$), a square ($N_{\text{var}} = 2$), a cube, etc. In this paper, ROG is considered such that the number of *equidistant points*, N, is identical along each edge. Figure 1 shows examples of such ROGs for various dimensions. An i th point of ROG is a row vector with N_{var} coordinates:

$$\mathbf{x}_i = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,v}, \ldots, x_{i,N_{\text{var}}} \right). \tag{2}$$

The coordinates are found within the unit hypercube: $0 \le x_{i,v} \le 1$. All points then form a matrix $\mathbf{x}$ (also referred to as the *sampling plan* in Section 4):

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_i \\ \vdots \\ \mathbf{x}_{N_{\text{sim}}} \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,v} & \cdots & x_{1,N_{\text{var}}} \\ x_{i,1} & \cdots & x_{i,v} & \cdots & x_{i,N_{\text{var}}} \\ x_{N_{\text{sim}},1} & \cdots & x_{N_{\text{sim}},v} & \cdots & x_{N_{\text{sim}},N_{\text{var}}} \end{bmatrix}. \tag{3}$$

The notation used in this paper is such that the number or symbol following the comma in the lower index stands for the dimension $v \in \langle 1, N_{\text{var}} \rangle$.

When all the coordinates are multiplied by number N then the sampling plan becomes a matrix of *rank numbers*, $\boldsymbol{\pi}$. In this way the ROG is scaled from the unit hypercube in to a hypercube of length N. Individual points form row vectors of indices: $\boldsymbol{\pi}_i = \left(\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,v}, \ldots, \pi_{i,N_{\text{var}}} \right)$, where $i = 1, \ldots, N_{\text{sim}}$ and $\pi_{i,v} \in \{1, 2, \ldots, N\}$, see Figure 1 left – the rank numbers are denoted on the top axis.

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_{1,1} & \cdots & \pi_{1,v} & \cdots & \pi_{1,N_{\text{var}}} \\ \pi_{i,1} & \cdots & \pi_{i,v} & \cdots & \pi_{i,N_{\text{var}}} \\ \pi_{N_{\text{sim}},1} & \cdots & \pi_{N_{\text{sim}},v} & \cdots & \pi_{N_{\text{sim}},N_{\text{var}}} \end{bmatrix}. \tag{4}$$

The uniform distribution of points within the unit hypercube is considered with the coordinates:

$$x_{i,v} = \frac{\pi_{i,v} - \frac{1}{2}}{N} \quad \begin{matrix} 1 \le i \le N_{\text{sim}} \\ 1 \le v \le N_{\text{var}} \end{matrix}. \tag{5}$$

The pairwise difference of points i and j is a row vector:

$$\boldsymbol{\delta}_{ij} = \boldsymbol{\pi}_i - \boldsymbol{\pi}_j, \tag{6}$$

that consists of differences of rank numbers along individual dimensions: $\delta_{ij,v} = \pi_{i,v} - \pi_{j,v}$. Using these differences in index numbers, the Euclidean distance between points i and j reads:

$$L_{ij} = \sqrt{\sum_{v=1}^{N_{\text{var}}} \left(x_{i,v} - x_{j,v} \right)^2} = \frac{1}{N} \sqrt{\sum_{v=1}^{N_{\text{var}}} \delta_{ij,v}^2}. \tag{7}$$

The number of distances between all pairs of N_{sim} points is generally:

$$N_{\text{p}} = \binom{N_{\text{sim}}}{2} = \frac{N_{\text{sim}} \left(N_{\text{sim}} - 1 \right)}{2}. \tag{8}$$

This number is $\mathcal{O}(N_{\text{sim}}^2)$ which implies its rapid growth with increasing discretization N and the dimension N_{var}. The calculation of distances between all pairs of points in ROG can be simplified.

2. Analysis of distances in ROG

The ROG is a collapsible arrangement and therefore the projections of pairwise distances along individual dimensions imply repeated vectors of index differences $\boldsymbol{\delta}_{ij}$. It is possible to obtain a list of all possible *vectors of index differences*, $\boldsymbol{\delta}_{ij}$, by considering only a limited list of vectors, $\boldsymbol{\delta}_t$. This list contains vectors that exhaust *unique* vectors $\boldsymbol{\delta}_{ij}$ irrespective of the order of elements. The number of such *types* of vectors is denoted as N_{c}. The vectors $\boldsymbol{\delta}_t$ are indexed by "t" and their list is selected so as to represent distances from the point $\boldsymbol{\pi}_1 = (1, \ldots, 1)$. The point $\boldsymbol{\pi}_1$ has the lowest possible indices and is placed in the "bottom left corner" of the hypercube. Therefore, we can write the vectors as:

$$\boldsymbol{\delta}_t \equiv \boldsymbol{\delta}_{1t} = \boldsymbol{\pi}_t - \boldsymbol{\pi}_1, \quad 1 \le t \le N_{\text{c}}. \tag{9}$$

The considered set of points $\boldsymbol{\pi}_t$ is selected to exhaust all points that fulfil:

$$\{ t : \pi_{t,v} \ge \pi_{t,w} \ \text{ for } \ 1 \le v < w \le N_{\text{var}} \}. \tag{10}$$

In other words, all vectors $\boldsymbol{\pi}_t$ form all possible non-increasing sequences of indices. Figure 2 illustrates the sit-

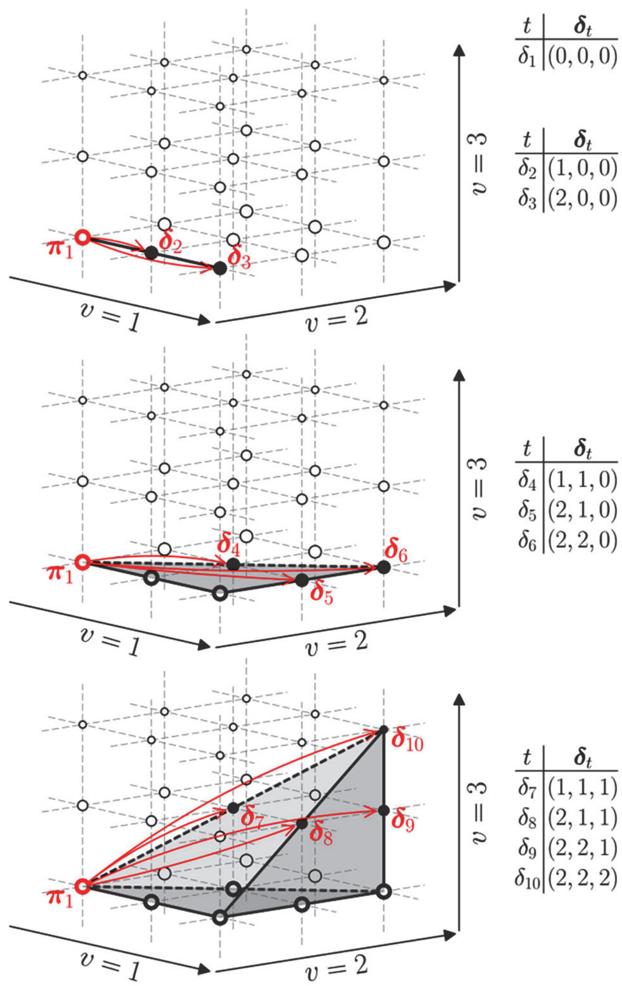

Figure 2. Enumeration of all vectors $\boldsymbol{\delta}_t$ and the corresponding visualization in a 3D grid with $N=3$ points along each dimension

uation for $N=3$ and $N_{\text{var}}=3$. The full list of vectors $\boldsymbol{\delta}_t$ is divided into three lists of vectors of dimensions 1, 2 and 3. It can be seen that the points in increasing dimensions progressively form a simplex.

We are now interested in obtaining the total number of such vectors $\boldsymbol{\delta}_t$. For a given design with (N, N_{var}), the number is equal to the number of ways to place N types of elements into a vector of length N_{var} (number of multisubsets). These are combinations with repetition:

$$N_{\text{c}} = \left(\!\!\binom{N}{N_{\text{var}}}\!\!\right) = \binom{N_{\text{var}}+N-1}{N_{\text{var}}} = \frac{(N_{\text{var}}+N-1)!}{N_{\text{var}}!(N-1)!}. \quad (11)$$

As can be seen from Figure 2, when $N = N_{\text{var}} = 3$, the number of vector types is $N_{\text{c}} = 10$. Eqn (11) can be simplified by decomposing the factorial in the numerator and cancelling the $(N-1)!$ terms in the numerator and denominator:

$$N_{\text{c}} = \frac{\prod_{i=0}^{N_{\text{var}}-1} (N+i)}{N_{\text{var}}!}. \quad (12)$$

Now, it is easy to write the bounds on N_{c} by taking either $i=0$ (lower bound) or $i = N_{\text{var}}/2$ (upper bound):

$$\frac{N_{\text{sim}}}{N_{\text{var}}!} = \frac{N^{N_{\text{var}}}}{N_{\text{var}}!} \lesssim N_{\text{c}} \lesssim \frac{\left(N+\frac{N_{\text{var}}}{2}\right)^{N_{\text{var}}}}{N_{\text{var}}!}. \quad (13)$$

The lower bound on the left hand side provides a useful estimate that quickly tends towards N_{c} with increasing N. It helps to show that the number of vector types is much lower than the total number of pairs, N_{p}. A direct comparison of Eqn (8) with the lower bound in Eqn (13) reveals that while N_{p} is proportional to N_{sim}^2, the number N_{c} is proportional to N_{sim} only. This fact is documented in Figure 4 (left), where the number of pairs, N_{p}, is independent of dimension and steeply grows with increasing N_{sim}. The number N_{c} slightly depends on N_{var}; however, the slope is half of that for N_{p}.

To show a more complex example of vectors $\boldsymbol{\delta}_t$, Table 1 presents all vectors $\boldsymbol{\delta}_t$ for $N=5$ and $N_{\text{var}}=3$. The number of pairwise distances in such a grid is $N_{\text{p}} = 7{,}750$, while the number of unique types of vectors is $N_{\text{c}} = 35$. The table is accompanied by the corresponding Euclidean point distances in a unit hypercube, L_t (squared and multiplied by N^2). Eqn (7) can be reused to calculate the Euclidean lengths using the index differences $\delta_{ij,v} = \delta_{t,v}$:

$$N \cdot L_t = \|\boldsymbol{\delta}_t\| = \sqrt{\sum_{v=1}^{N_{\text{var}}} \delta_{t,v}^2}. \quad (14)$$

With all possible vector types, $\boldsymbol{\delta}_t$, and the associated distances, L_t, we are now interested in obtaining the total number of occurrences of each vector type, denoted as n_t. One can show that this number is a product of three coefficients:

$$n_t = n_t^{\text{p}} \times n_t^{\text{d}} \times n_t^{\text{n}}, \quad 2 \le t \le N_{\text{c}}, \quad (15)$$

where the three coefficients n_t^{p}, n_t^{d} and n_t^{n} represent three types of spatial transformations of $\boldsymbol{\delta}_t$: rotations, reflections and translations. Note that we index the vector types from $t=2$; this is because $\boldsymbol{\delta}_1$ is the null-vector. One can check that the number of pairwise distances (Eqn (8)) equals to the sum of all n_t's:

$$N_{\text{p}} = \sum_{t=2}^{N_{\text{c}}} n_t = \binom{N_{\text{sim}}}{2}. \quad (16)$$

The coefficients are derived in the following subsections.

2.1. Number of permutations n_t^{p} of integer difference vectors

When considering a distance corresponding to a certain pattern $\boldsymbol{\delta}_t$, it is important to count the number of ways to achieve this distance from the point $\boldsymbol{\pi}_1$. In particular, n_t^{p} corresponds to the number of ways to rotate the vector around the point $\boldsymbol{\pi}_1$ which is equivalent to renumbering the dimensions. In other words, one must consider how many times the differences $\delta_{t,v}$ can be permuted. To make

Table 1. Enumeration of all vectors $\boldsymbol{\delta}_t$ with their corresponding numbers n_t and the associated lengths in a unit hypercube for a 3D grid ($N_{\text{var}} = 3$) with $N = 5$ points along each dimension

t	$\boldsymbol{\delta}_t$	n_t^{p}	n_t^{d}	n_t^{n}	n_t	$\left(N \cdot L_t\right)^2$	$\left(N \cdot \overline{L_t}\right)^2$
1	(0, 0, 0)	–	–	–	–	–	–
2	(1, 0, 0)	3	1	100	300	1	1
3	(2, 0, 0)	3	1	75	225	4	4
4	(3, 0, 0)	3	1	50	150	9	4
5	(4, 0, 0)	3	1	25	75	16	1
6	(1, 1, 0)	3	2	80	480	2	2
7	(2, 1, 0)	6	2	60	720	5	5
8	(3, 1, 0)	6	2	40	480	10	5
9	(4, 1, 0)	6	2	20	240	17	2
10	(2, 2, 0)	3	2	45	270	8	8
11	(3, 2, 0)	6	2	30	360	13	8
12	(4, 2, 0)	6	2	15	180	20	5
13	(3, 3, 0)	3	2	20	120	18	8
14	(4, 3, 0)	6	2	10	120	25	5
15	(4, 4, 0)	3	2	5	30	32	2
16	(1, 1, 1)	1	4	64	256	3	3
17	(2, 1, 1)	3	4	48	576	6	6
18	(3, 1, 1)	3	4	32	384	11	6
19	(4, 1, 1)	3	4	16	192	18	3
20	(2, 2, 1)	3	4	36	432	9	9
21	(3, 2, 1)	6	4	24	576	14	9
22	(4, 2, 1)	6	4	12	288	21	6
23	(3, 3, 1)	3	4	16	192	19	9
24	(4, 3, 1)	6	4	8	192	26	6
25	(4, 4, 1)	3	4	4	48	33	3
26	(2, 2, 2)	1	4	27	108	12	12
27	(3, 2, 2)	3	4	18	216	17	12
28	(4, 2, 2)	3	4	9	108	24	9
29	(3, 3, 2)	3	4	12	144	22	12
30	(4, 3, 2)	6	4	6	144	29	9
31	(4, 4, 2)	3	4	3	36	36	6
32	(3, 3, 3)	1	4	8	32	27	12
33	(4, 3, 3)	3	4	4	48	34	9
34	(4, 4, 3)	3	4	2	24	41	6
35	(4, 4, 4)	1	4	1	4	48	3

an example, $\boldsymbol{\delta}_2 = \left(1,0,0\right)$ corresponds to unit distance from $\boldsymbol{\pi}_1$ along the first dimension. However, the same type of vector can also be placed along the remaining two dimensions ($\left(0,1,0\right)$ and $\left(0,0,1\right)$), because we consider the metric to be isotropic. Basically, one has to count the number of different arrangements of indexes $\delta_{t,v}$, some of which might be identical. In other words, one has to count the number of *permutations with repetition*:

$$n_t^{\text{p}} = \frac{N_{\text{var}}\,!}{n_{t,0}\,!\cdot n_{t,1}\,!\cdots n_{t,N-1}\,!} = \frac{N_{\text{var}}\,!}{\prod\limits_{k=0}^{N-1}\left[n_{t,k}\,!\right]}, \tag{17}$$

where $n_{t,k}$ are the numbers of repetitions of index differences k in vector $\boldsymbol{\delta}_t$. These numbers of repetitions can be counted as:

$$n_{t,k} = \sum_{v=1}^{N_{\text{var}}} \mathbf{1}\left(\delta_{t,v} - k\right), \tag{18}$$

where $\mathbf{1}(x)$ is an indicator function returning one if there is a match, i.e. a repetition ($x = 0$). Otherwise ($x \neq 0$), the indicator function returns zero.

To explain how the formula works, we return to the example of vector $\boldsymbol{\delta}_2 = \left(1,0,0\right)$. The total number of permutations of these three index differences is $N_{\text{var}}\,!$, see the numerator in Eqn (17). However, since the zero difference is featured more than once, some arrangements in the permutations are identical. In particular, the zero difference is repeated twice and therefore one has to consider that permutations with repetition only count for *different* arrangements. The numerator in Eqn (17) must be divided by the numbers of arrangements of indistinguishable differences.

One has to go through the list of all possible index differences, $k \in \{0,1,\ldots,N-1\}$, that can occur in $\boldsymbol{\delta}_t$ and count the number of occurrences for each k . In this example, the zero difference yielding $n_{2,0} = 2$ is featured twice, the unit difference ($n_{2,1} = 1$) once and an index difference of two or higher ($n_{2,2} = n_{2,3} = n_{2,4} = 0$) zero times. Therefore, the denominator reads $2!\cdot 1!\cdot 0!\cdot 0!\cdot 0! = 2$. Finally, the number of distinguishable arrangements of $\boldsymbol{\delta}_2 = \left(1,0,0\right)$ is therefore $n_2^{\text{p}} = 3!/2 = 3$, see Table 1.

Figure 3 (left) illustrates that $\boldsymbol{\delta}_{11} = \left(3,2\right)$ a 2D design has the same distance from $\boldsymbol{\pi}_1$ as a vector $\left(2,3\right)$, so that $n_{11}^{\text{p}} = 2$.

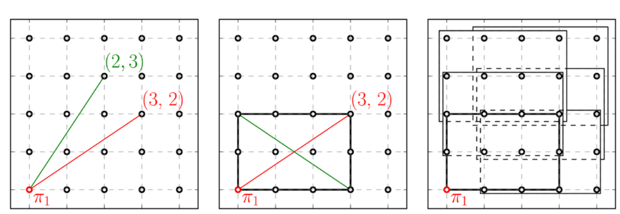

Figure 3. Illustration of the three types of transformation corresponding to n_t^{p} , n_t^{d} and n_t^{n} for a grid with $N = 5$ and $N_{\text{var}} = 2$

2.2. Number of space diagonals (r -agonals) n_t^{d}

The second coefficient, n_t^{d} , corresponds to the number of reflections of each vector type, $\boldsymbol{\delta}_t$. The vector $\boldsymbol{\delta}_t$ forms a space diagonal (r -agonal, i.e. the longest diagonal) of a hyperrectangle (sometimes called an n -orthotope or just a box). The dimension of such a diagonal $\boldsymbol{\delta}_t$ can be quantified as the number of nonzero index differences:

$$r_t = \sum_{v=1}^{N_{\text{var}}} \mathbf{1}\left(\delta_{t,v}\right), \qquad 1 \leq r_t \leq N_{\text{var}}, \tag{19}$$

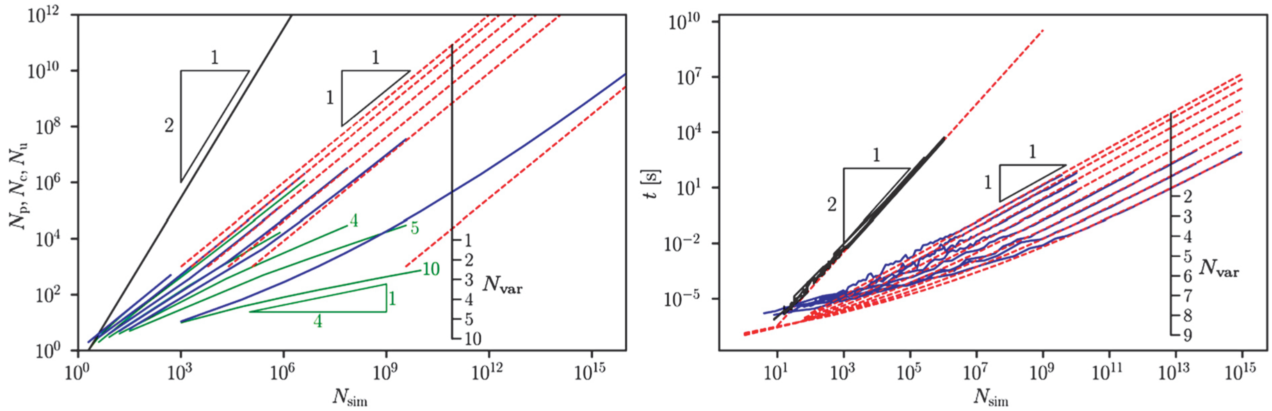

Figure 4. Left: number of pairwise distances. The number of all pairs, N_p, (black line) is compared with the number N_c of vectors $\boldsymbol{\delta}_t$ (blue lines) and the number of unique distances N_u (green line). The red dashed lines show the approximated formula $N_{sim} / N_{var}!$ from Eqn (13); Right: computing times (black/blue lines). The enumeration of all N_p pairwise distances takes much longer than the enumeration of N_c distance pairs and their counts. The red dashed lines show Eqns (26) and (27)

where $\mathbf{1}(x)$ is an indicator function returning one for nonzero integer differences $\delta_{i,v}$ and zero otherwise. For a given dimension, r_t, the number of various space diagonals in a hyper-rectangle is:

$$n_t^{\mathrm{d}} = 2^{r_t - 1}. \tag{20}$$

For example, the number of space diagonals for vectors $\boldsymbol{\delta}_6 = (1,1,0)$ and $\boldsymbol{\delta}_7 = (2,1,0)$ forming *diagonals* of a square/rectangle is equal to $r_6^{\mathrm{d}} = r_7^{\mathrm{d}} = 2$ and the number of *triagonals* in 3 dimensional space with vectors $\boldsymbol{\delta}_{16} = (1,1,1)$ or $\boldsymbol{\delta}_{17} = (2,1,1)$ (cube/cuboid) is $n_{16}^{\mathrm{d}} = n_{17}^{\mathrm{d}} = 4$.

Figure 3 (centre) illustrates that a two-dimensional vector $\boldsymbol{\delta}_{11} = (3,2)$ is one of two possible diagonals in the same rectangle, so that of $n_{11}^{\mathrm{d}} = 2$.

2.3. Number of hyperrectangles n_t^{n}

The last coefficient, n_t^{n}, corresponds to all possible translations of a "sub-hyperrectangle" within the hypercube. The coefficient is a product of the numbers of ways to translate the hyperrectangle along each dimension:

$$n_t^{\mathrm{n}} = \prod_{v=1}^{N_{var}} \left(N - \delta_{t,v} \right). \tag{21}$$

The vector $\boldsymbol{\delta}_{11} = (3,2)$ featured in Figure 3 (left and centre) is shown to have $n_{11}^{\mathrm{n}} = 6$ possible placements within the square shown in Figure 3 (right).

2.4. Histogram of pairwise distances

Equations (14) and (15) enable the evaluation of all pairwise distances (vector types $\boldsymbol{\delta}_t$) and their counts, n_t, in an ROG. We note that the list of vectors $\boldsymbol{\delta}_t$ may not represent the list of *unique* pairwise distances. The reason is that the sum of the squared projections of several different vectors $\boldsymbol{\delta}_t$ may be the same, see e.g. vectors $\boldsymbol{\delta}_4$ and $\boldsymbol{\delta}_{20}$ in Table 1. To obtain the most highly condensed list of different lengths and their frequencies possible, we propose sorting the vectors according to length and grouping the identical ones. The numbers of unique distances are plotted by green lines in Figure 4 (left).

In Figure 5, histograms of all distances L, squared distances L^2 and distances raised to the power $N_{var} + 1$ are presented for various dimensions N_{var}. The reason for presenting $L^{N_{var}+1}$ will become clear in Section 4. These histograms were obtained for a high number of points, N_{sim}.

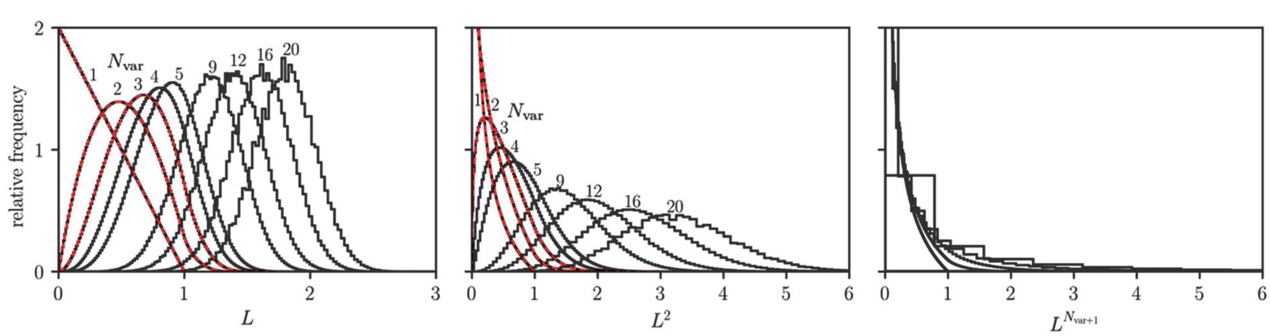

Figure 5. Probability density functions (histograms) of a random pairwise distance (left), squared distance (middle) and a distance raised to $N_{var} + 1$ (right) in a hypercube of dimension N_{var}. The red lines are the exact density functions for a random pair of points

When the number of points grows high ($N_{sim} \to \infty$), the average distances between two points can be seen as the distance between two points chosen at random inside a unit hypercube. The solution is trivial for $N_{var} = 1$ ($E[L] = 1/3$). For $N_{var} = 2$, the mean distance reads $(2+\sqrt{2})/15 + \ln(1+\sqrt{2})/3 \approx 0.521405434$. This number is a known constant available in the On-Line Encyclopedia of Integer Sequences (OEIS), which is published electronically at https://oeis.org, 2010, Sequence "A091505" (OEIS 2010). Its explicit expression has already been provided by Ghosh (1951). For $N_{var} = 3$, the mean distance reads $\frac{4+17\sqrt{2}}{105} - \frac{\pi}{15} - \frac{2}{35}\sqrt{3} + \ln(1+\sqrt{2})/5 + 2\ln(2+\sqrt{3})/5 \approx$ 0.661707181. This average distance is known as the Robbins constant (Robbins, Bolis 1978), which is available in the OEIS as Sequence "A073012" (OEIS 2010). Average distances are available for dimensions $N_{var} = 4,5,6,7$ and 8 in the OEIS as Sequences "A103983, A103984, A103985, A103986 and A103987" (OEIS 2010). They read 0.7776656535, 0.8785309152, 0.9689420830, 1.0515838734 and 1.1281653402, respectively (see also Bailey *et al.* 2007; Weisstein n.d.). Table 1 in Anderssen *et al.* (1976) provides approximate values for $N_{var} = 9$ and 10: 1.19985 and 1.26748.

The distributions of distances L for increasing dimension N_{var} tend towards Gaussian distribution. Anderssen *et al.* (1976) provided bounds for the average distance D depending on the dimension, N_{var}:

$$\frac{\sqrt{N_{var}}}{3} \le E[L] \le \sqrt{\frac{N_{var}}{6}} \sqrt{\frac{1}{3}\left[1 + 2\sqrt{1 - \frac{3}{5N_{var}}}\right]}. \quad (22)$$

We have found a tighter upper bound on the mean value that possesses correct asymptotic properties: $E[L]$ tends towards $\sqrt{N_{var}/6}$ as $N_{var} \to \infty$. The suggested bound reads:

$$E[L] \lessapprox \sqrt{\frac{N_{var} - 1/3}{6}}, \quad (23)$$

and provides an exact value for $N_{var} = 1$ ($E[L] = \frac{1}{3}$). Gates (1985) derived an asymptotic formula:

$$E[L] \approx \sqrt{\frac{N_{var}}{6}} \left(1 - \frac{7}{40\,N_{var}} - \frac{65}{896 N_{var}^2} + \ldots\right). \quad (24)$$

This formula is already very accurate for $N_{var} > 2$.

The variance for $N_{var} = 1$ reads simply $D[L] = 1/18$. For higher N_{var}, it is found within 1/16 and 1/18 and tends towards 7/120 as $N_{var} \to \infty$.

From this analysis, it is clear that the Euclidean distance of two points picked randomly from a unit hypercube becomes virtually identical and deterministic in very high dimensions. The standard deviation stays approximately constant with increasing dimension N_{var} while the mean value keeps growing. The coefficient of variation of a random distance L is therefore asymptotically proportional to $1/\sqrt{N_{var}}$. In such a case the distance contrast decreases and it is said that the distances concentrate (Aggarwal *et al.* 2001; Flexer, Schnitzer 2015). This *distance concentration in high-dimensional spaces* might lead to undesired effects in some applications such as the optimization of the design of experiments (Audze, Eglājs 1977; Eliáš, Vořechovský 2016).

The squared distance, L^2, has a trivial mean value, $N_{var}/6$, and variance, $7N_{var}/180$. The central limit theorem tells us that the square of the distance is almost normally distributed for large N_{var}.

3. Speed-up and implementation details

The proposed algorithm delivers the list of pairwise distances and their counts quicker than a naïve approach that simply evaluates the distances for all pairs of points. The reason for this is that the number of vector types in an ROG is considerably smaller than the number of all pairs.

The ratio between the number of pairs among the N_{sim} points in the unit hypercube, N_p (see Eqn (8)), and the number of unique type vectors, N_c (Eqns (11), (12)), provides a hint about the speed-up associated with using the proposed methodology. Asymptotically the ratio reads:

$$\frac{N_p}{N_c} = \frac{N_{var}!}{2}\left(N_{sim} - 1\right) = \frac{N_{var}!}{2}\left(N^{N_{var}} - 1\right). \quad (25)$$

We have implemented both the naïve approach and the suggested approach in C language and the computing times for various N_{sim} and N_{var} have been measured. Figure 4 (right) displays the computing time of the naïve approach, $t(N_p)$, and the time taken by the suggested approach, $t(N_c)$. It is no surprise that the times taken by the naïve approach are proportional to the number of pairs, N_p. The measured times are almost completely independent of the dimension, N_{var}, and the following formula provides an excellent approximation of the time in seconds:

$$t(N_p) = N_p \cdot C_1 \approx N_{sim}{}^2 \cdot \frac{C_1}{2}, \quad (26)$$

where the constant C_1 has been obtained by fitting the times measured with our hardware (Intel Core i7-860 2.8 GHz) as $C_1 = 7 \cdot 10^{-9}$ [sec].

The computing times obtained with the proposed algorithm seem to be proportional to the number of vectors processed, N_c, and they also depend linearly on the dimension:

$$t(N_c) = N_c N_{var} \cdot C_2. \quad (27)$$

The constant $C_2 = 1.5 \cdot 10^{-8}$ [sec] has been obtained by fitting the times measured using the same hardware and compiler as for the naïve algorithm.

By using the lower bound from Eqn (13) one can conclude that the computing time is asymptotically linear in N_{sim}:

$$t(N_c) = \frac{N_{sim}}{(N_{var} - 1)!} \cdot C_2, \quad (28)$$

see the triangles in Figure 4 (right), while N_p is quadratic in N_{sim}. For high N_{sim} and N_{var}, the speed-up is dramatic:

$$\frac{t(N_{\mathrm{p}})}{t(N_{\mathrm{c}})} \approx (N_{\mathrm{var}} - 1)!\, N_{\mathrm{sim}} \cdot 0.2\,. \quad (29)$$

The ratio between these two numbers increases very fast with an increasing number of simulations.

The analysis deals with the evaluation of distances and does not take into account the preparation of input arrays of coordinates x_{ij} or vector types $\boldsymbol{\delta}_i$.

The prototype of an implementation of a function for the evaluation of pairwise distances and the number of their occurrences in a unit hypercube was made in Python programming language (van Rossum *et al.* 1991). The Python source code is presented in the Appendix.

4. Application

As mentioned in the Introduction, one of the possible applications of the proposed method and algorithm is in the field of the optimal placement of points used in Monte Carlo integration, which is one possible application of the Design of Experiments.

Monte Carlo sampling is the most general technique for estimation of probabilistic integrals, such as those representing the statistical moments of functions of random variables (e.g. Kala *et al.* 2017; Strauss *et al.* 2017; Vahdatirad *et al.* 2015), sensitivity analyses or reliability analyses (see e.g. Kong *et al.* 2013; Liao *et al.* 2015).

Monte Carlo integration approximates an integral of a function f as an average:

$$\int_{[0,1]^{N_{\mathrm{var}}}} f(\mathbf{x})\mathrm{d}\mathbf{x} \approx \frac{1}{N_{\mathrm{sim}}} \sum_{i=1}^{N_{\mathrm{sim}}} f(\mathbf{x}_i)\,, \quad (30)$$

where $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_{\mathrm{sim}}}\}$ is a set of points in the *design space* $[0,1]^{N_{\mathrm{var}}}$ (a space of sampling probabilities). These points can be transformed into points in real space, where the function is evaluated. The ratio $1/N_{\mathrm{sim}}$ is a weight that is supposed to be identical for all points as their chance of being selected is supposed to be equal. Therefore, the points must be uniformly distributed throughout the unit hypercube. The way of selecting individual points from the design domain influences the quality of the approximation. According to Koksma-Hlawka inequality (Fang, Ma 2001; Niederreiter 1992), the error of such an approximation depends on the character of the examined function (its bounded variation, $V(f)$) and the star discrepancy of the set of integration points, $D^{\star}_{N_{\mathrm{sim}}}(\mathbf{x}_1, \ldots, \mathbf{x}_{N_{\mathrm{sim}}})$:

$$\left| \frac{1}{N_{\mathrm{sim}}} \sum_{i=1}^{N_{\mathrm{sim}}} f(\mathbf{x}_i) - \int_{[0,1]^{N_{\mathrm{var}}}} f(\mathbf{x})\mathrm{d}\mathbf{x} \right| \le V(f) \cdot D^{\star}_{N_{\mathrm{sim}}}(\mathbf{x}_1, \ldots, \mathbf{x}_{N_{\mathrm{sim}}}). \quad (31)$$

As the character of the examined function is not under the analyst's control, the only way to decrease the upper bound of the approximation error is to reduce the discrepancy of the set $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_{\mathrm{sim}}}\}$. Therefore, uniform designs are sought. These can be obtained by optimizing an originally random design (either Monte Carlo or Latin Hypercube Sampling) using various optimality criteria, either some type of discrepancy criterion or some other criterion aimed at ensuring the uniformity of the final design. The latter type of criterion is usually based on distances between pairs of points in the design domain (traditionally the inter-site distances are considered).

A very general example of such a criterion is the ϕ criterion. This criterion was first defined by Morris and Mitchell (1995), see also Damblin *et al.* (2013), Pronzato and Müller (2012):

$$\phi = \left[\sum_{\substack{i,j=1\ldots N_{\mathrm{sim}} \\ i<j}} L_{ij}^{-p} \right]^{\frac{1}{p}}, \quad (32)$$

L_{ij} being the inter-site Euclidean distance of points i and j defined by Eqn (7).

A limit case of the ϕ criterion for $p \to \infty$ is the Maximin criterion (Johnson *et al.* 1990) (sometimes called the mindist criterion):

$$\phi_{\mathrm{Mm}} = \min_{\substack{i,j=1\ldots N_{\mathrm{sim}} \\ i\neq j}} L_{ij}\,. \quad (33)$$

The value of this criterion must be maximized to obtain a better design.

A specific case of the ϕ criterion, in which $p = 2$, is the Audze-Eglãjs (AE) criterion defined earlier in Audze and Eglãjs (1977) as:

$$E^{\mathrm{AE}} = \frac{1}{N_{\mathrm{p}}} \sum_{i=1}^{N_{\mathrm{sim}}} \sum_{j=i+1}^{N_{\mathrm{sim}}} \frac{1}{L_{ij}^2}\,. \quad (34)$$

The use of this criterion has been presented in, e.g., Bates *et al.* (2003), Fuerle and Sienz (2011), Husslage *et al.* (2011), Janouchová and Kučerová (2013), Kovalovs and Rucevskis (2011), Liefvendahl and Stocki (2006), Vu *et al.* (2014). Note that the original formulation of the criterion does not feature standardization by the number of pairs considered (division by N_{p}).

An improved version of the AE criterion, called the Periodic Audze-Eglãjs (PAE) criterion, has recently been published in Eliáš and Vořechovský (2016). It removes a major flaw of the standard AE criterion related to the presence of boundaries:

$$E^{\mathrm{PAE}} = \frac{1}{N_{\mathrm{p}}} \sum_{i=1}^{N_{\mathrm{sim}}} \sum_{j=i+1}^{N_{\mathrm{sim}}} \frac{1}{\overline{L}_{ij}^2}\,, \quad (35)$$

here: $\overline{L}_{ij}$ is not the Euclidean distance of points in the design space, but a distance redefined as the distance between point i and the nearest image of point j in a periodically extended space. Taking the nearest distance corresponds to folding and gluing the design domain N_{var} times – compare L and $\overline{L}$ in Figure 6. The new definition of this length (modification of Eqn (7)) stands:

$$\overline{L}_{ij}^2 = \sum_{v=1}^{N_{\mathrm{var}}} \left[\min\left(\Delta_{ij,v}, 1 - \Delta_{ij,v}\right) \right]^2, \quad (36)$$

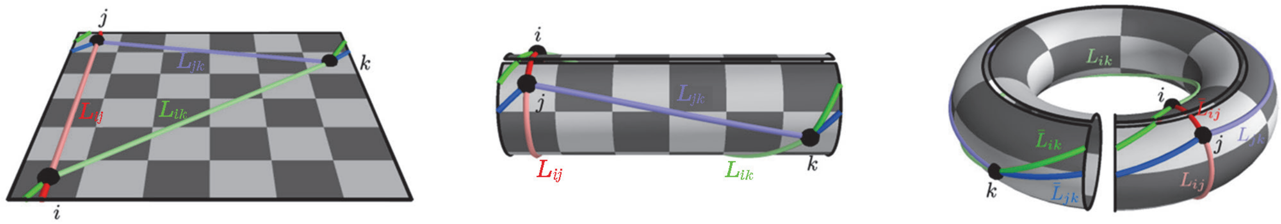

Figure 6. Illustration of the modified metric (periodic distance $\overline{L}$) in the N_{var} times "folded & glued" 2D design domain

where the projected distance along direction ν reads

$$\Delta_{ij,\nu} = |x_{i,\nu} - x_{j,\nu}| \ . \tag{37}$$

This distance may also be defined in terms of $\overline{\delta}$ in a similar manner as to δ as presented in Eqn (7), then:

$$\overline{L}_{ij}^2 = \frac{1}{N^2} \sum_{\nu=1}^{N_{\text{var}}} \overline{\delta}_{ij,\nu}^2 \ , \tag{38}$$

where:

$$\overline{\delta}_{ij,\nu} = \begin{cases} \min(\delta, N-\delta) & \text{for } N \text{ even} \\ \min(\delta, N-\delta-1) & \text{for } N \text{ odd} \end{cases} . \tag{39}$$

The rightmost column in Table 1 presents the shortest squared distances $\overline{L}_{ij}^2$ in a periodic space (multiplied by N^2). These can be directly compared with the standard squared inter-site distances L_{ij}^2 used in the AE criterion.

The proposed algorithm for the evaluation of pairwise distances can be used without any modification. However, to get a list of unique distances in a periodic space, the list of distances must be sorted according to the periodic length and some vector types must be merged as they become identical to others. The number of unique periodic distances in a periodic space is roughly one half of the standard distances in a hypercube.

The modified formulation of AE and PAE criteria (Eqns (34) and (35)) can be viewed as the computation of the *average* inverse squared distance $1/L_{ij}^2$ (or $1/\overline{L}_{ij}^2$). The convergence of the average distance measures is studied for the orthogonal grid in Figure 7 (bottom left). We argue that this criterion, which can also be understood as the potential energy of a system of charged particles (Eliáš, Vořechovský 2016), changes its character for various dimensions N_{var} and also for various numbers of points, N_{sim} . In a 1D situation, the energy tends towards infinity *linearly* with increasing point count $N_{\text{sim}} = N$. In 2D ($N_{\text{var}} = 2$), the energy tends towards infinity as $\ln(N_{\text{sim}})$. Such a divergence is not a power law and therefore various sample sizes may yield dissimilar optimal patterns (self-similarity would be manifested through a power law dependence of the criterion on the point count). For 3D and higher N_{var} , the energy tends towards a constant for increasing N_{sim} , see Figure 7 (bottom left). This means that for a given $N_{\text{var}} > 1$, a higher number of points yields a different proportion between the long-range and short-range interactions. This may not be desirable behaviour as the criterion in high dimensions and also for a high number of points becomes insensitive to local clusters of points: it becomes dominated by long-range interactions.

Therefore, we also consider a modification of the criterion where the inverse distances are raised to a power that might be dependent on the problem dimension, N_{var} . The definition of such a criterion is analogous to the ϕ criterion but we suggest a reason for the selection of the distance power, p :

$$E_{\text{n}}^{\text{AEp}} = \frac{1}{N_{\text{p}}} \sum_{i=1}^{N_{\text{sim}}} \sum_{j=i+1}^{N_{\text{sim}}} \frac{1}{L_{ij}^p} \ , \tag{40}$$

and equivalently, the periodic version:

$$E_{\text{n}}^{\text{PAEp}} = \frac{1}{N_{\text{p}}} \sum_{i=1}^{N_{\text{sim}}} \sum_{j=i+1}^{N_{\text{sim}}} \frac{1}{\overline{L}_{ij}^p} \ . \tag{41}$$

It is suggested that the power be at least $p = N_{\text{var}} + 1$. Why? With this power, the interaction is dominated by short-range forces. The convergence of the potential energy $E_{\text{n}}^{\text{AEp}}$, or better $E_{\text{n}}^{\text{PAEp}}$, towards infinity for a uniform distribution of points is a power law. Such a convergence signalizes the self-similarity of the problem (the absence of a length scale). In other words, a zoom into a sufficiently representative subdomain has all the features of the full design and the energy value can be easily scaled from the value corresponding to the smaller zoom.

This can be shown by studying the behaviour of the radial part of the integral of the potential over the volume V of an N_{var} dimensional domain. The potential energy for a uniform design reads:

$$I = \int_{N_{\text{var}}} \frac{1}{L^p} \mathrm{d}^{N_{\text{var}}} V \ . \tag{42}$$

After transformation into polar coordinates, one writes:

$$I = \int_{N_{\text{var}}} \varphi \, \mathrm{d}^{N_{\text{var}}-1} V \, |J| \frac{1}{L^p} \mathrm{d}L \ , \tag{43}$$

where $|J|$ is the Jacobian. The volume element is thereby given as:

$$\mathrm{d}^{N_{\text{var}}} V = L^{N_{\text{var}}-1} \mathrm{d}L \cdot \mathrm{d}\varphi \prod_{i=1}^{N_{\text{var}}-2} \sin^{N_{\text{var}}-1-i}(\varphi_i) \ . \tag{44}$$

Therefore, the integral is performed over the product $L^{N_{\text{var}}-1-p}$. Performing just the radial integration leads to

$$I_r = \int \frac{L^{N_{\text{var}}-1}}{L^p} \mathrm{d}L \ . \tag{45}$$

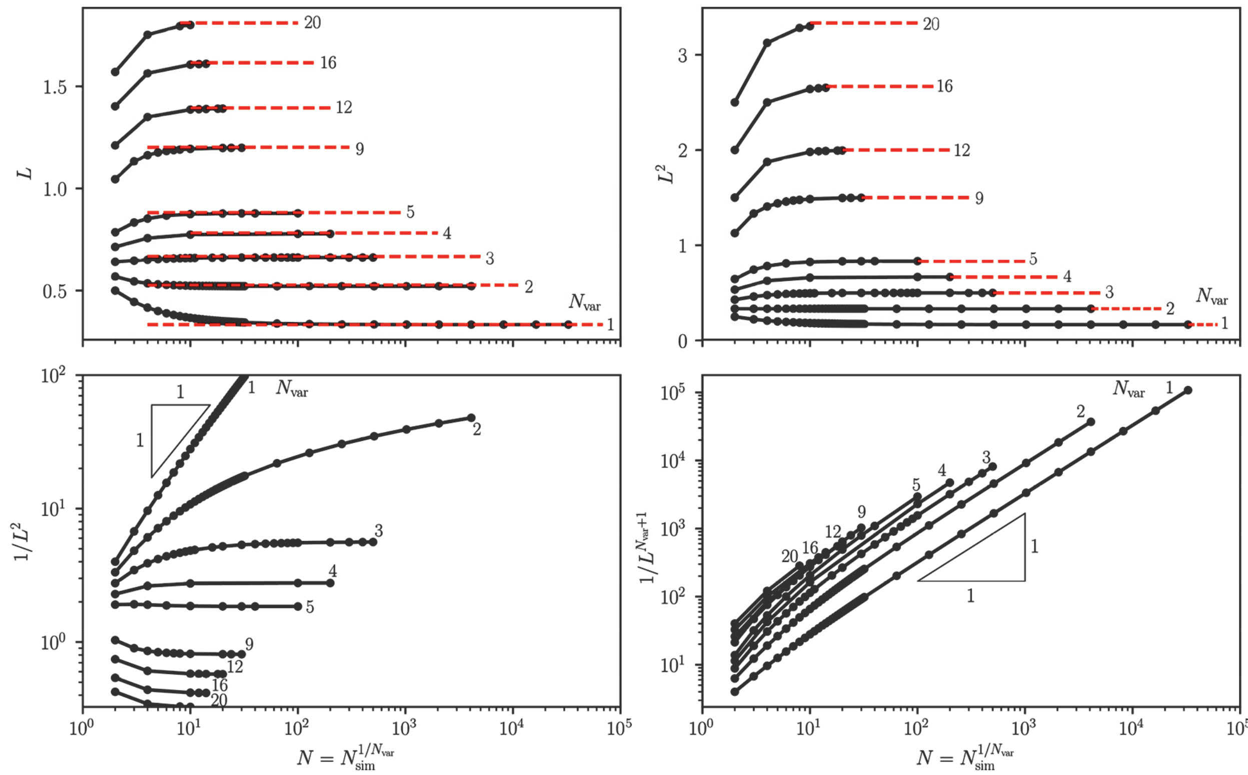

Figure 7. Averages of various powers of inter-site distances between all pairs of points.
The dashed lines show asymptotic solutions enumerated in Section 2.4

For $p = 2$ as used in the AE criterion, we obtain the behaviour described above. Using $p = N_{var}$ leads to $I_r = \int 1/L\,dL = \ln(L)$, which diverges logarithmically, and the interaction is still long-range. Using $p = N_{var} + 1$ yields $I_r = \int 1/L^2\,dL = 1/L$, which is the desired asymptotic behaviour driven by short-range interaction. Using higher powers only increases the (asymptotically constant) ratio between short-range and long-range interactions.

Figure 5 (right) shows a qualitative change in the histograms of lengths featured in the criterion. From Figure 7 right bottom, it can be seen that the mean value of $1/L^{N_{var}+1}$ diverges linearly with increasing N, which is a common behaviour for any hypercube dimension, N_{var}. The power law divergence of the criterion (Eqn (40)) obtained for ROG confirms the behaviour of the radial part of the integral, i.e. the short-range interaction dominates the potential energy, see Figure 7 (right bottom).

Using a different power in the definition of the criteria does not necessitate the re-evaluation of the list of vector types and pairwise distances. The same holds for the selection of the definition of the metric itself (see the criticism of the Euclidean length at the end of Section 2.4). Instead, a fractional norm instead the ubiquitous Euclidean norm can be used. Changing the distance norm is independent of the presented algorithm and fast evaluation of the norm histogram can help when studying behaviour of various norms. When a different distance norm is used or a different power, p, the list of various vectors and their counts can be used to quickly enumerate any criterion of the ROG.

Conclusions

The paper presents a simple algorithm for the exact evaluation of pairwise distances among all pairs of points forming a regular orthogonal grid of points within a unit hypercube. The algorithm provides the distances and their counts considerably faster than a naïve algorithm based on the evaluation of all pairs of points. The speed-up is enabled by the regular structure of the orthogonal grid.

The paper shows how the algorithm can be used in the study of the behaviour of criteria concerning regularity, discrepancy or space-fillingness. In particular, since the regular orthogonal grid can be considered to be close to optimal point placement, the algorithm can be used for the fast estimation of the lower bound of various distance-based criteria of design optimality for the possible sample sizes.

Acknowledgements

The authors acknowledge the financial support provided by the Czech Science Foundation under projects Nos. GA16-22230S and GA15-07730S.

The authors would like to thank Dr. Godehard Sutmann for a valuable discussion about the physics of interacting particles.

Funding

This work was supported by the Czech Science Foundation under Grants Nos. 16-22230S and GA15-07730S.

Disclosure statement

Authors declare that they have not any competing financial, professional, or personal interests from other parties.

References

Aggarwal, C. C.; Hinneburg, A.; Keim, D. A. 2001. On the surprising behavior of distance metrics in high dimensional space, in *Proceedings of the 8th International Conference "Database Theory – ICDT 2001"*, 4–6 January 2001, London, UK, 420–434. https://doi.org/10.1007/3-540-44503-X_27

Anderssen, R. S.; Brent, R. P.; Daley, D. J.; Moran, A. P. 1976. Concerning $\int_0^1 \cdots \int_0^1 \sqrt{x_1^2 + \ldots + x_k^2} dx_1 \cdots dx_k$ and a Taylor series method, *SIAM Journal on Applied Mathematics* 30: 22–30. https://doi.org/10.1137/0130003

Audze, P.; Eglājs, V. 1977. New approach for planning out of experiments, *Problems of Dynamics and Strengths* 35: 104–107.

Bailey, D. H.; Borwein, J. M.; Crandall, R. E. 2007. Box integrals, *Journal of Computational and Applied Mathematics* 206(1): 196–208. https://doi.org/10.1016/j.cam.2006.06.010

Bates, S. J.; Sienz, J.; Langley, D. S. 2003. Formulation of the Audze–Eglais uniform Latin hypercube design of experiments, *Advances in Engineering Software* 34(8): 493–506. https://doi.org/10.1016/S0965-9978(03)00042-5

Box, G. E. P. 1954. The exploration and exploitation of response surfaces: Some general considerations and examples, *Biometrics* 10(1): 16–60. https://doi.org/10.2307/3001663

Bucher, C. G.; Bourgund, U. 1990. A fast and efficient response surface approach for structural reliability problems, *Structural Safety* 7(1): 57–66. https://doi.org/10.1016/0167-4730(90)90012-E

Chen, H.; Huang, H. Z.; Lin, D. K. J.; Liu, M. Q. 2016. Uniform sliced Latin hypercube designs, *Applied Stochastic Models in Business and Industry* 32: 574–584. https://doi.org/10.1002/asmb.2192

Chu, D. P. 2006. Distance between random points in two rectangular cities, *Communications in Statistics – Simulation and Computation* 35(2): 257–276. https://doi.org/10.1080/03610910600591818

Chudoba, R.; Sadílek, V.; Rypl, R.; Vořechovský, M. 2013. Using Python for scientific computing: an efficient and flexible evaluation of the statistical characteristics of functions with multivariate random inputs, *Computer Physics Communications* 184(2): 414–427. https://doi.org/10.1016/j.cpc.2012.08.021

Damblin, G.; Couplet, M.; Iooss, B. 2013. Numerical studies of space-filling designs: optimization of Latin Hypercube Samples and subprojection properties, *Journal of Simulation* 7(4): 276–289. https://doi.org/10.1057/jos.2013.16

Eliáš, J.; Vořechovský, M. 2016. Modification of the Audze–Eglājs criterion to achieve a uniform distribution of sampling points, *Advances in Engineering Software* 100: 82–96. https://doi.org/10.1016/j.advengsoft.2016.07.004

Fang, K.-T.; Ma, C.-X. 2001. Wrap-around L_2-discrepancy of random sampling, Latin Hypercube and uniform designs, *Journal of Complexity* 17(4): 608–624. https://doi.org/10.1006/jcom.2001.0589

Flexer, A.; Schnitzer, D. 2015. Choosing ℓ^p norms in high-dimensional spaces based on hub analysis, *Neurocomputing* 169: 281–287. https://doi.org/10.1016/j.neucom.2014.11.084

Fuerle, F.; Sienz, J. 2011. Formulation of the Audze–Eglais uniform Latin hypercube design of experiments for constrained design spaces, *Advances in Engineering Software* 42(9): 680–689. https://doi.org/10.1016/j.advengsoft.2011.05.004

Gates, D. J. 1985. Asymptotics of two integrals from optimization theory and geometric probability, *Advances in Applied Probability* 17(4): 908–910. https://doi.org/10.1017/S0001867800015470

Ghosh, B. 1951. Random distances within a rectangle and between two rectangles, *Bulletin of Calcutta Statistical Association* 43: 17–24.

Gupta, S.; Manohar, C. S. 2004. An improved response surface method for the determination of failure probability and importance measures, *Structural Safety* 26(2): 123–139. https://doi.org/10.1016/S0167-4730(03)00021-3

Hamzah, M. O.; Golchin, B.; Woodward, D. 2017. A quick approach for rheological evaluation of warm asphalt binders using response surface method, *Journal of Civil Engineering and Management* 23(4): 475–486. https://doi.org/10.3846/13923730.2016.1210216

Huang, H. Z.; Lin, D. K. J.; Liu, M. Q.; Yang, J. F. 2016. Computer experiments with both qualitative and quantitative variables, *Technometrics* 58: 495–507. https://doi.org/10.1080/00401706.2015.1094416

Husslage, B. G. M.; Rennen, G.; van Dam, E. R.; den Hertog, D. 2011. Space-filling Latin hypercube designs for computer experiments, *Optimization and Engineering* 12(4): 611–630. https://doi.org/10.1007/s11081-010-9129-8

Husslage, B. G. M. 2006. *Maximin designs for computer experiments*: PhD thesis. CentER, Tilburg University.

Iman, R. C.; Conover, W. J. 1980. Small sample sensitivity analysis techniques for computer models with an application to risk assessment, *Communications in Statistics – Theory and Methods* A9(361–926): 1749–1842.

Janouchová, E.; Kučerová, A. 2013. Competitive comparison of optimal designs of experiments for sampling-based sensitivity analysis, *Computers & Structures* 124: 47–60. https://doi.org/10.1016/j.compstruc.2013.04.009

Johnson, M. E.; Moore, L. M.; Ylvisaker, D. 1990. Minimax and maximin distance designs, *Journal of Statistical Planning and Inference* 2: 131–148. https://doi.org/10.1016/0378-3758(90)90122-B

Kala, Z.; Valeš, J.; Jönsson, J. 2017. Random fields of initial out of straightness leading to column buckling, *Journal of Civil Engineering and Management* 23(7): 902–913. https://doi.org/10.3846/13923730.2017.1341957

Kong, D.; Lu, S.; Frantzich, H.; Lo, S. M. 2013. A method for linking safety factor to the target probability of failure in fire safety engineering, *Journal of Civil Engineering and Management* 19(sup1): S212–S221. https://doi.org/10.3846/13923730.2013.802718

Kovalovs, A.; Rucevskis, S. 2011. Identification of elastic properties of composite plate, in *Proceedings of Annual Conference on Functional Materials and Nanotechnologies – FM&NT*, Vol. 23. IOP Publishing, Ltd. https://doi.org/10.1088/1757-899X/23/1/012034

Li, W. W.; Wu, C. F. J. 1997. Columnwise-pairwise algorithms with applications to the construction of supersaturated designs, *Technometrics* 39(2): 171–179. https://doi.org/10.2307/1270905

Liao, K.-W.; Lu, H.-J.; Wang, C.-Y. 2015. A probabilistic evaluation of pier-scour potential in the Gaoping River Basin of

Taiwan, *Journal of Civil Engineering and Management* 21(5): 637–653. https://doi.org/10.3846/13923730.2014.890650

Liefvendahl, M.; Stocki, R. 2006. A study on algorithms for optimization of Latin hypercubes, *Journal of Statistical Planning and Inference* 136(9): 3231–3247. https://doi.org/10.1016/j.jspi.2005.01.007

Moltchanov, D. 2012. Distance distributions in random networks, *Ad Hoc Networks* 10(6): 1146–1166. https://doi.org/10.1016/j.adhoc.2012.02.005

Montgomery, D. C. 2006. *Design and analysis of experiments*. 8th ed. John Wiley & Sons.

Morris, M. D.; Mitchell, T. J. 1995. Exploratory designs for computational experiments, *Journal of Statistical Planning and Inference* 43(3): 381–402. https://doi.org/10.1016/0378-3758(94)00035-T

Mortazavi, A.; Toğan, V.; Nuhoğlu, A. 2017. Weight minimization of truss structures with sizing and layout variables using integrated particle swarm optimizer, *Journal of Civil Engineering and Management* 23(8): 985–1001. https://doi.org/10.3846/13923730.2017.1348982

Niederreiter, H. 1992. *Random number generation and Quasi-Monte Carlo methods*. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611970081

OEIS. 2010. *The on-line encyclopedia of integer sequences* [online], [cited 16 September 2016]. Available from Internet: http://oeis.org

Pronzato, L.; Müller, W. G. 2012. Design of computer experiments: space filling and beyond, *Statistics and Computing* 22(3): 681–701. https://doi.org/10.1007/s11222-011-9242-3

R Core Team. 2016. *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Robbins, D. P.; Bolis, T. S. 1978. Average distance between two points in a box, *American Mathematical Monthly* 85(4): 277–278. https://doi.org/10.2307/2321177

Sacks, J.; Schiller, S. B.; Welch, W. J. 1989. Designs for computer experiments, *Technometrics* 31(1): 41–47. https://doi.org/10.1080/00401706.1989.10488474

Sadílek, V.; Vořechovský, M. 2017. *ROG – Regular Orthogonal Grid* [online], [cited 18 September 2018]. Available from Internet: https://github.com/kelidas/ROG

Siddiquea, N.; Adelib, H. 2016. Applications of gravitational search algorithm in engineering, *Journal of Civil Engineering and Management* 22(8): 981–990. https://doi.org/10.3846/13923730.2016.1232306

Strauss, A.; Wan-Wendner, R.; Vidovic, A.; Zambon, I.; Yu, Q.; Frangopol, D. M.; Bergmeister, K. 2017. Gamma prediction models for long-term creep deformations of prestressed concrete bridges, *Journal of Civil Engineering and Management* 23(6): 681–698. https://doi.org/10.3846/13923730.2017.1335652

Tobler, W. R. 1970. A computer movie simulating urban growth in the Detroit region, *Economic Geography* 46: 234–240. https://doi.org/10.2307/143141

Vahdatirad, M. J.; Bayat, M.; Andersen, L. V.; Ibsen, L. B. 2015. Probabilistic finite element stiffness of a laterally loaded monopile based on an improved asymptotic sampling method, *Journal of Civil Engineering and Management* 21(4): 503–513. https://doi.org/10.3846/13923730.2014.890660

van Etten, J. 2012. *gdistance: Distances and routes on geographical grids*. Department of Mathematics, KTH, Stockholm, Sweden.

van Rossum, G.; Hettinger, R.; Diedrich, J. 1991. *The Python programming language*. Prentice Hall.

Vořechovský, M.; Novák, D. 2009. Correlation control in small sample Monte Carlo type simulations I: A Simulated Annealing approach, *Probabilistic Engineering Mechanics* 24(3): 452–462. https://doi.org/10.1016/j.probengmech.2009.01.004

Vu, H. M.; Forth, J. P.; Dao, D. V.; Toropov, V. V. 2014. The use of optimisation for enhancing the development of a novel sustainable masonry unit, *Applied Mathematical Modelling* 38(3): 853–863. https://doi.org/10.1016/j.apm.2013.07.026

Weisstein, E. W. n.d. *Hypercube line picking* [online], [cited 16 September 2016]. Available from Internet: http://mathworld.wolfram.com/HypercubeLinePicking.html

Appendix

The prototype of the described algorithm written in Python programming language uses the standard Python libraries `math`, `functools` and `itertools`. The package `itertools` is used to create an iterator for N_c types of difference vectors $\boldsymbol{\delta}_t$ using the function `combinations_with_replacement`. The function `rog_lenghts` for evaluation of pairwise distances in ROG takes three parameters. The first two arguments `n`, `nvar` are required to define the hypercube parameters. The last keyword argument `periodic` enables evaluation of distances in a periodically extended space as described in Section 4. The function returns two lists: i) `lengths` contains pairwise distances and ii) `counts` contains their counts.

Re-implementation in C language yields significant speed-up and enables the evaluation of larger designs. Python and C codes for evaluation of pairwise distances are available in the Github repository (Sadílek, Vořechovský 2017).

```
import math
import itertools
from functools import reduce

def rog_lengths(n, nvar, periodic=False):
    '''
    Evaluate pairwise distances among points forming
    regular orthogonal grid in a hypercube.

    Parameters
    ----------
    n : int
        number of equidistant points
        along an individual dimension
    nvar : int
        number of input random variables
        (dimension of a hypercube)
    periodic : bool, optional
        evaluate lengths in a periodically extended space.
        Default is False.

    Returns
    -------
    lengths : list of floats
        pairwise distances among points
    counts : list of ints
        number of distances of the same type

    Examples
    --------
    >>> n = 3
    >>> nvar = 2
    >>> rog_lengths(n, nvar, periodic=False)
    ([0.3333333333333333, 0.6666666666666666, 0.47140452079103173, \
0.7453559924999299, 0.9428090415820635], [12, 6, 8, 8, 2])
     >>> rog_lengths(n, nvar, periodic=True)
     ([0.3333333333333333, 0.3333333333333333, 0.47140452079103173, \
0.47140452079103173, 0.47140452079103173], [12, 6, 8, 8, 2])
    '''
    lengths = []
    counts = []
    # prepare iterator for difference vectors delta_t
    deltas = itertools.combinations_with_replacement(range(n), nvar)
    next(deltas) # skip the first null vector (0,...,0)
    # loop over difference vectors delta_t
    for delta_t in deltas:
        # sum of squared differences
        if periodic:
            # update delta_t for periodic space eq.(39)
            h = [(dt - n // 2) > 0 for dt in delta_t]
```

```
            delta_t_pae = [abs(hi * n - dt) for hi, dt in zip(h, delta_t)]
            length_t = sum([dt ** 2 for dt in delta_t_pae])
        else:
            length_t = sum([dt ** 2 for dt in delta_t])

        # number n_t^n    eq. (21)
        ntn = reduce(lambda x, y: x * y, [n - dt for dt in delta_t])
        # number n_t^p    eq. (17)
        ntp = math.factorial(nvar)
        for k in set(delta_t):
            # frequencies of differences in the vector delta_t
            ntp //= math.factorial(sum([(dt - k) == 0 for dt in delta_t]))
        # number n_t^d eq. (20)
        ntd = 2 ** (sum([dt > 0 for dt in delta_t]) - 1)
        # number n_t eq. (15)
        nt = ntn * ntp * ntd

        lengths.append(length_t)
        counts.append(nt)
    # calculate real lengths in unit hypercube eq. (7) or (38)
    lengths = [l ** 0.5 / float(n) for l in lengths]
    return lengths, counts
```