

Quasi-Monte Carlo: halftoning in high dimensions?

Kenneth M. Hanson

Los Alamos National Laboratory, Los Alamos, New Mexico 87545

ABSTRACT

The goal in Quasi-Monte Carlo (QMC) is to improve the accuracy of integrals estimated by the Monte Carlo technique through a suitable specification of the sample point set. Indeed, the errors from N samples typically drop as N^{-1} with QMC, which is much better than the $N^{-1/2}$ dependence obtained with Monte Carlo estimates based on random point sets. The heuristic reasoning behind selecting QMC point sets is similar to that in halftoning, that is, to spread the points out as evenly as possible, consistent with the desired point density. I will outline the parallels between QMC and halftoning, and describe an halftoning-inspired algorithm for generating a sample set with uniform density, which yields smaller integration errors than standard QMC algorithms in two dimensions.

Keywords: Quasi-Monte Carlo, Monte Carlo integration, low-discrepancy sequences, Halton sequence, Sobel sequence, halftoning, direct binary search, minimum visual discrepancy, Voronoi analysis

1. INTRODUCTION

The goal of the standard Monte Carlo (MC) technique¹ is to estimate the integral of a function over a specified M -dimensional domain from evaluations of the function at points that are randomly chosen within that domain. The objective in Quasi-Monte Carlo² (QMC) is to improve those estimates through a suitable specification of the sample point set. It has been shown that the errors from N samples for a fixed number of dimensions typically fall off as N^{-1} with QMC, much more quickly than with MC, namely, $N^{-1/2}$.

Digital halftoning is the process of creating a pattern of black dots on a white background to create the illusion of a gray-scale image.^{3,4} One of the principal goals in halftoning is avoid introducing undesirable texture into the rendered image, which is typically caused by clumping of the dots, or uneven dot placement that accompanies random dot distributions. In a sense, QMC has the same goal, whether it is implicitly or explicitly stated. The clumpiness in random point distributions also exists in standard Monte Carlo, and lead to lower sampling efficiency than more uniformly distributed point distributions. One observes that in regions of uniform low density, halftoned images seem to have characteristics deemed desirable in QMC.

The heuristic reasoning behind selecting QMC point sets is similar to that in halftoning, that is, to avoid clumping of the points (dots). The visual similarities between the patterns generated in halftoning and QMC lead one to speculate whether halftoning techniques might provide some useful lessons for quasi-Monte Carlo, or visa versa? I will demonstrate that a method for generating point sets, which is inspired by basic concepts used in halftoning, yields more accurate estimates of 2D integrals than is obtained with some standard QMC sequences. Of course, halftoning is conducted in only two dimensions. It must be kept in mind that the same algorithms that work for 2D may not work for higher dimensions. I will discuss the implications for higher dimensions and other potential approaches to enhanced QMC methods.

The objective of the present study is to find improved QMC techniques to reduce the number of function evaluations needed to achieve a specified accuracy in the estimate of an integral. For example, the function to be integrated may depend on a simulation of a complex physical process that might take several hours, or even several days, to calculate on the fastest computers available. In such a situation, the time required to generate an appropriate set of samples is inconsequential. Incidentally, intelligently selected point sets can be used for purposes other than integration, for example, for performing sensitivity analysis of computer models.^{5,6}

Further author information: Full address: MS D413, Los Alamos National Laboratory, (for packages, include street; Bikini Atoll Road), Los Alamos, New Mexico 87545, E-mail: kmh@lanl.gov, Telephone: 1 505 667 1402, Web: <http://public.lanl.gov/kmh/>



Figure 1. An example of a digital halftone image generated with the direct-binary-search algorithm taken from Ref. 8, which shows the high quality that is achievable with state-of-the-art halftone rendering of gray-scale images. (©IEEE)

2. HALFTONING

The goal of the halftoning process is to render a gray-scale image, subject to whatever limitations are present in the printing or display process. Given the wide variety of constraints in printing technologies, a similarly wide variety of halftoning techniques exist.⁴ For the purpose of the present study, I will focus on a single type of rendering, referred to as digital halftoning, in which the printing process is only capable of putting black dots on a white surface.^{3,7} The constraints on this process can include the dot size, the minimum distance between dots, etc. Because halftoning is used in commercial printers, which need to print pages rapidly, a large fraction of the published work on halftoning is devoted to finding ways to speed up the halftoning process, through use of look-up tables, for example. The trade-off between speed and rendition quality becomes a critical design issue.

Figure 1 shows an excellent example of a high-quality halftoned image.⁸ For the purpose of demonstrating the method, the dots in this image are much larger than would normally be the case in any good publication or printing situation. Specifically, the spacing between pixels in this image correspond to about 70 dots per inch, whereas modern laser printers boast resolutions of 1200 dpi and greater. To see what this image would look like when produced by a 1200 dpi laser printer and viewed at a normal viewing distance of 30 cm, you would have to view it from a distance of about 5 meters! Ideally, the dots should be small enough that they are not visually resolved by the observer. The resolution limits of the human eye effectively blur the dots so that the dot pattern is perceived as the intended image with continuous gray levels.

2.1. Model for Visual Discrepancy

Figure 1 was produced using the direct-binary-search (DBS) technique, which will be described in the following section.^{7,8} As in many halftoning techniques, DBS is based on minimizing the perceived difference between the halftone image and the original gray-scale image that it is supposed to represent. Because the judge of the quality of the halftoned image is a human observer, halftoning algorithms are often based on properties of the human visual system (HVS).

In one simple description of the HVS,^{3,4} it is assumed that the effective modulation transfer function (MTF) for the eye is proportional to an exponential of $-c|f|$, where f is the radial spatial frequency on the observed

page, and the factor c is related to the distance of the observer from the page. The 2D inverse Fourier transform of this MTF yields a blur function of the form

$$h(r) \propto (w^2 + r^2)^{-3/2}, \quad (1)$$

where r is the radial distance in the printed image. The width of this 2D Cauchy distribution is characterized by the parameter w . The full-width at half maximum (FWHM) of the radial profile of this distribution is $1.553 w$. An important feature of $h(r)$ is that it has a long tail, behaving like r^{-3} for large r .

Assuming a position-invariant blur function $h(x, y)$, the difference between the two perceived images is the convolution:

$$e(x, y) = h * [d - g] = \int h(x - x', y - y') [d(x', y') - g(x', y')] dx' dy', \quad (2)$$

where $*$ denotes the convolution operation, $d(x, y)$ is the dot image, and $g(x, y)$ is the original gray-scale image to be rendered. Because the convolution redistributes intensities, it is necessary to specify what to use outside of the domains of images d and g . This topic is not often mentioned in discussions of halftoning but will be dealt with in Sect. 4. To quantify the perceived discrepancy between the halftone image and the actual gray-scale image, the most-often-used cost function is the total power in the error image

$$\varphi = \int_R |e(x, y)|^2 dx dy. \quad (3)$$

A variety of simplifying assumptions go into this formula, but it seems to be adequate for producing halftone images of high visual quality.

2.2. Direct Binary Search Algorithm

The DBS algorithm^{7,8} is a specific approach to minimizing the HVS-motivated cost function given by Eq. (3). The following is a simplified description of the DBS algorithm. It is assumed that the halftone dot pattern is represented in terms of a discretized image in which each pixel represents a dot and has one of two values, either black or white. An initial pattern of dots is produced in which the density of the dots is approximately proportional to the gray-scale image being rendered. Any one of several methods may be used to generate this initial image, for example, thresholding of a set of random numbers. In this iterative algorithm, each pixel in the image is considered one at a time. The change in φ produced by swapping the pixel's value with each of its eight nearest neighbors is calculated. The effect on φ of toggling, or reversing, the pixel's value is also calculated. If any of these options results in a decrease in φ , the change that reduces φ the most is kept. One pass through all pixels in the halftone image is counted as one iteration. The number iterations can vary from a half dozen to many times that, depending on the characteristics of the initial image and the stopping criterion. Although the DBS algorithm yields halftoned images of excellent quality, it requires intensive calculation.⁸

If the model for visualization of the halftone image presented in the preceding section were strictly followed, the parameter w , which controls the amount of visual blur, would be determined by the assumed distance of the observer from the printed image. In practice, however, w is determined by practical considerations related to the convergence of the algorithm. It has been observed that the best choice for the width may depend on dot density.^{8,9} In a sense, the HVS model is often taken only as a guide; significant departures from it may be considered if they yield better looking halftone images.

3. QUASI-MONTE CARLO

In standard Monte Carlo techniques,¹ one evaluates integrals on the basis of a set of point samples. The integral of a function $f()$ of the parameter vector \mathbf{x} is estimated as

$$\int_R f(\mathbf{x}) d\mathbf{x} = \frac{V_R}{N} \sum_{i=1}^N f(\mathbf{x}_i), \quad (4)$$

where R indicates the domain of integration in M dimensions, V_R is the volume of R , and the N samples \mathbf{x}_i are randomly drawn from a uniform probability density function defined over R .

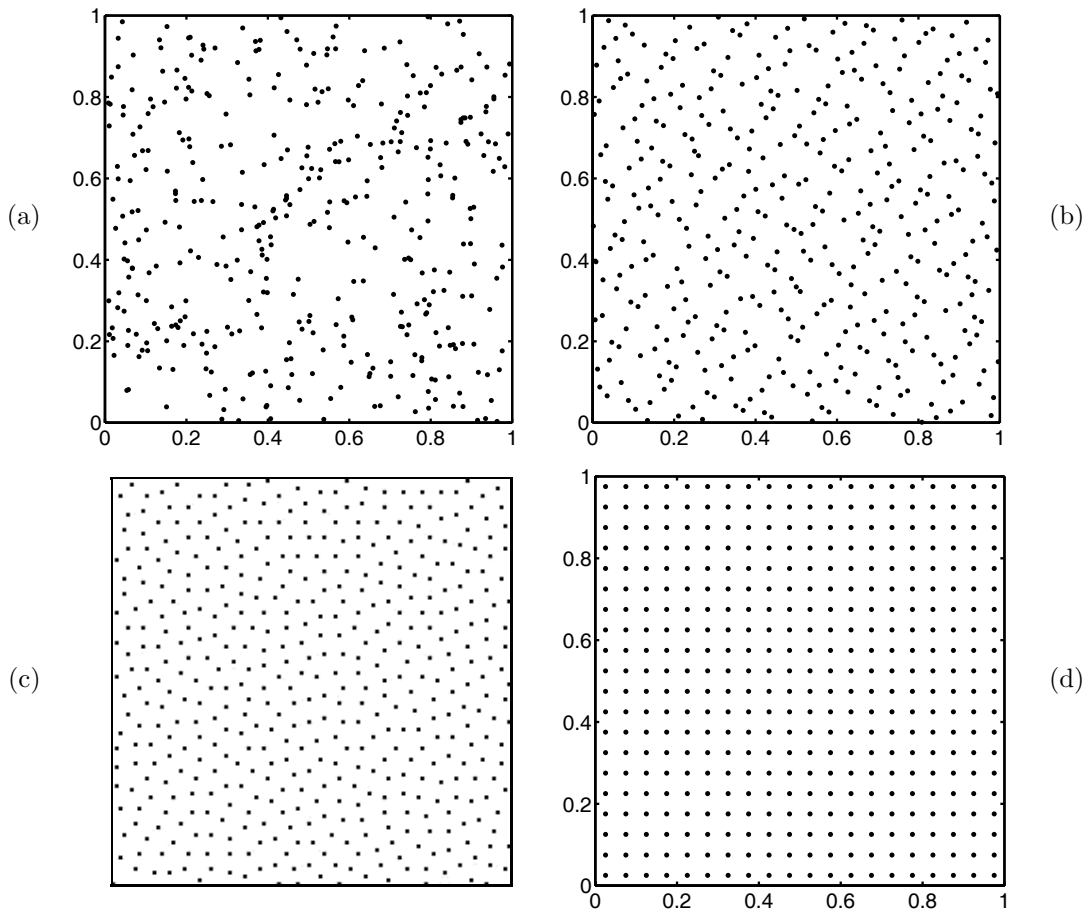


Figure 2. Four examples of different kinds of sample sets, each consisting of 400 points, which can be used to estimate an integral over the unit square using Eq. (4). (a) A random point set, each point in which is obtained by randomly drawing values for x and y from a uniform distribution from 0 to 1. (b) The first 400 points from the Halton sequence [2,3]. (c) Subsample taken from the sky region in the halftoning example shown in Fig. 1. (d) A regular array of points arranged on a square grid.

The objective of the quasi-Monte Carlo technique is to reduce the number of function evaluations needed to obtain a given accuracy in a Monte Carlo type of integration, and to accelerate its convergence as N increases, a goal that is typically achieved.¹⁰ One useful feature of QMC is that any number of samples can be generated. Furthermore, an arbitrary number of additional samples can be added to an existing set of samples.

Figure 2 shows four different sets of points that cover the unit square in 2D. Panel (a) shows a set of random numbers that represents the type of point distribution that would be used in classical MC. The points in panel (c) are taken from the low-density portion of the sky near the top of Fig. 1. This dot pattern is observed to be somewhat similar to the patterns seen in 2D point sets that used in quasi-Monte Carlo, as displayed in Figs. 2b, 3a, and 3c. It is known¹¹ that by using quadrature methods, which typically rely on uniformly spaced sampling on a Cartesian square grid, shown in panel (d), the rms error drops as N^{-1} . For classical MC, the rms error drops more slowly, as $N^{-1/2}$.

It is interesting to compare the four patterns in Fig. 2, and consider the fact that the accuracy for integrating `func2`, defined in Eq. (8), using these point sets is (a) 2.5%, (b) 0.5%, (c) 0.14%*, and (d) 0.09%. It seems that the more uniformly distributed the points, the better they are for MC-style integration.

*This rms error is actually for the point set produced by the MVD algorithm, and shown in Fig. 5a, which is virtually indistinguishable in its general character from Fig. 2c.

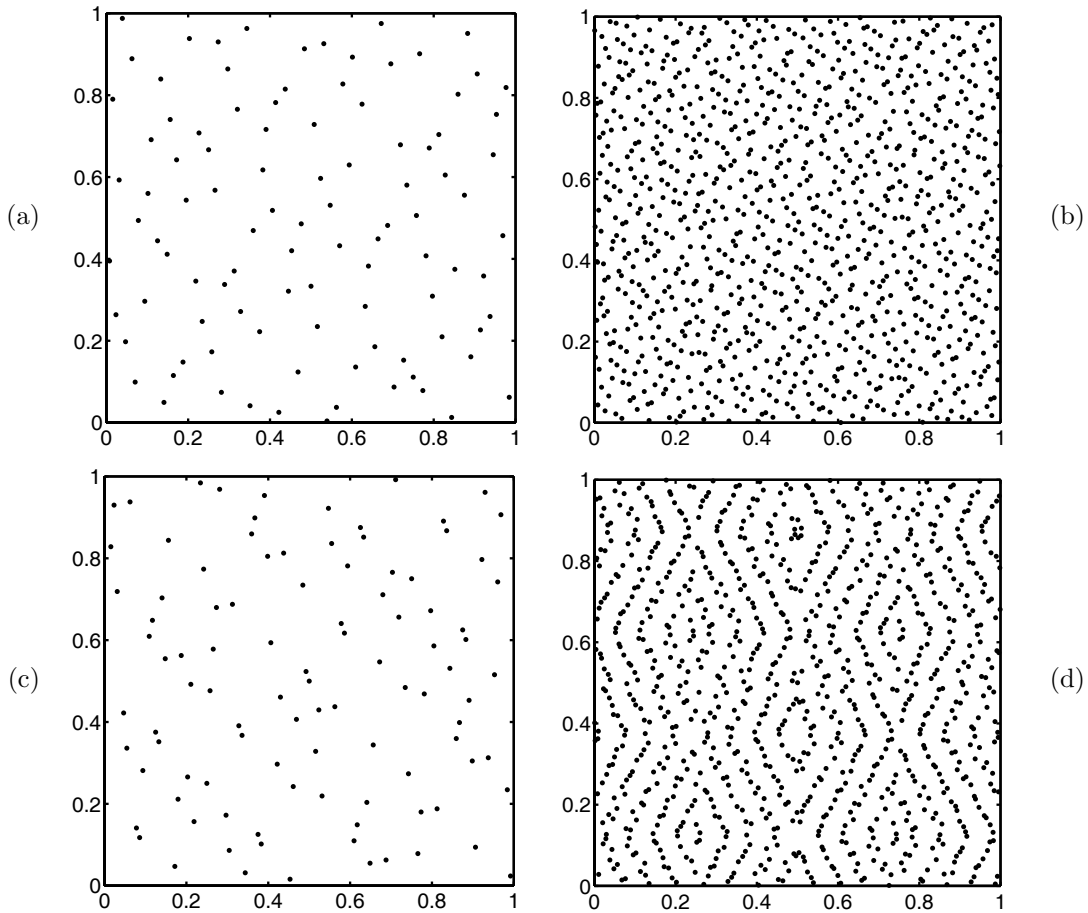


Figure 3. Examples of a few standard sequences used in quasi-Monte Carlo. The panels in the top row display (a) the first 100 and (b) first 1000 points from the Halton sequence based on the prime numbers 2 and 3. The bottom row shows (c) the first 100 and (d) first 1000 points from the Sobel sequence. The patterns seen in both examples for 100 points appear somewhat similar to those in halftoning, as least when compared to random dot patterns. However, visually disturbing patterns develop in both sequences as the number of points increases, especially for the Sobel sequence.

With most standard QMC sequences, the point sequences are the same from one time to the next. The algorithms for generating the points are deterministic, and typically depend on the prime numbers, which are typically small primes. This situation is quite different than that for the generation of pseudo-random numbers, where the sequence is usually different (and independent) each time it is requested. Of course, by setting the ‘seed’ to the same value, the same sequence can be obtained again, but this is under the user’s control.

A metric that is often used in the QMC field is the local discrepancy, the L_2 norm of which is, in 2D

$$D_2 = \left\{ \int_U \left[\frac{n(x, y)}{N} - A_0(x, y) \right]^2 dx dy \right\}^{\frac{1}{2}}, \quad (5)$$

where $A_0(x, y)$ is the area of the rectangle with one corner anchored at the origin and its opposite corner at (x, y) , and $n(x, y)$ is the number of samples that lie inside the same rectangle out of a total of N samples. The integration is over the unit square, designated by U . This quantity is clearly a measure of the uniformity of the distribution of the sample points. D_2 , and other similarly defined discrepancy measures, are useful because they form the basis for several upper bounds on the integration error for specified classes of functions.² The definition of D_2 may be made more robust by employing an average over rectangles anchored at all four corners of the

unit square.¹² One thing that seems odd about this metric is that it is based on rectangles with horizontal and vertical edges, which seems to emphasize the separable characteristics of point sets and may be not well suited for nonseparable functions. A metric with more isotropic properties might be more appropriate.

QMC sequences are typically obtained by combining low-discrepancy one-dimensional sequences.² Sequences in M dimensions are generated by forming an M -tuple $(x_1, x_2, x_3, \dots, x_M)$ and using a unique 1D low-discrepancy sequence for each coordinate. Numerous QMC sequences have been proposed. Figure 3 shows some 2D sequences based on two standard QMC sequences, the Halton and the Sobel, which were generated using MatLab.¹³ The Halton sequences shown in panels (a) and (b) are based on the prime numbers 2 and 3. For example, the Halton sequence based on the prime number 3 is $1/3, 2/3, 1/9, 4/9, 7/9, 2/9, 5/9, 8/9, 1/27, 10/27, 19/27$ etc. Other primes would yield different patterns. However, when the same primes are used, the sequences are perfectly repeatable and the same 2D pattern are obtained. The Sobel sequence is based on primitive polynomials.¹¹

One argument for using MC instead of sampling on a regular grid (quadrature method) is to avoid a coherent interaction between the sample set and the unknown function being integrated.¹¹ Consider for example, the case when the integrand is a periodic function with the same period as the spacing between sample points. Then the estimated value of the integral depends critically on the position of the sample points relative to the structure in the integrand. A regular point pattern will yield the same result every time, if the same pattern is used. A random sample pattern used in Monte Carlo will yield different results every time, so one is less likely to be fooled. Another well-known reason for using MC in high dimensions is that the number of points required by quadrature methods becomes too large.¹¹

Although there are similarities between QMC and halftoning patterns, there seems to have been little intersection of the two fields of research. One possible exception is the work by Ohbuchi and Aono¹⁴ in which they used QMC sequences to improve rendering of surfaces and shadows in a 3D scene. It should be mentioned that many techniques other than QMC and MC exist for conducting efficient sensitivity studies on computer models, the intended application of the present work. Some of these are Latin hypercube sampling,¹⁵ stratified sampling,¹ and orthogonal arrays.⁶

4. MINIMUM VISUAL DISCREPANCY FOR POINT SETS

Taking a cue from the DBS algorithm, I now describe an algorithm for minimizing the visual discrepancy (MVD) between a set of points and an image with uniform density. Starting with some arbitrary point pattern, the MVD algorithm considers each point in the set in a randomly permuted order. Instead of comparing the blurred dot image to the blurred gray-scale image, as in Eq. (3), it is better to reference the blurred dot image to its own mean value by computing its variance:

$$\psi = \int_R |e(x, y)|^2 dx dy - \left(\int_R e(x, y) dx dy \right)^2, \quad (6)$$

where e is the convolution of d with h , as in Eq. (2), but with $g = 0$. This self referencing is possible because the number of points is fixed. To minimize ψ , the following iterative algorithm is employed. Each point is considered, one at a time, in a randomly ordered way. The value of ψ is calculated for fixed-length steps in eight different directions, along the axes and along the 45° diagonals. If any of these steps results in a decrease in ψ , that step yielding the smallest ψ is accepted. After all points in the point set have been considered, which is called one iteration, the step length is decreased by a factor and the process is repeated until a predetermined number of iterations K have been performed. The total number of iterations required to achieve satisfactory point distributions ranges from five to about 15.

The step length for the first iteration is chosen on the basis of how evenly distributed the points are in the initial pattern. It is specified in terms a characteristic distance between points for the final, uniformly distributed point set, taken in two dimensions as $a = \sqrt{N}$, for N points. In the present study, the initial step length is chosen between $0.2a$ to $0.4a$, dependent on the uniformity of the initial point pattern. The larger value is used when the initial pattern is a random point set, the smaller when starting with the Halton sequence, for example. In practice, the step length is adjusted so that approximately half the points considered get moved to lower ψ

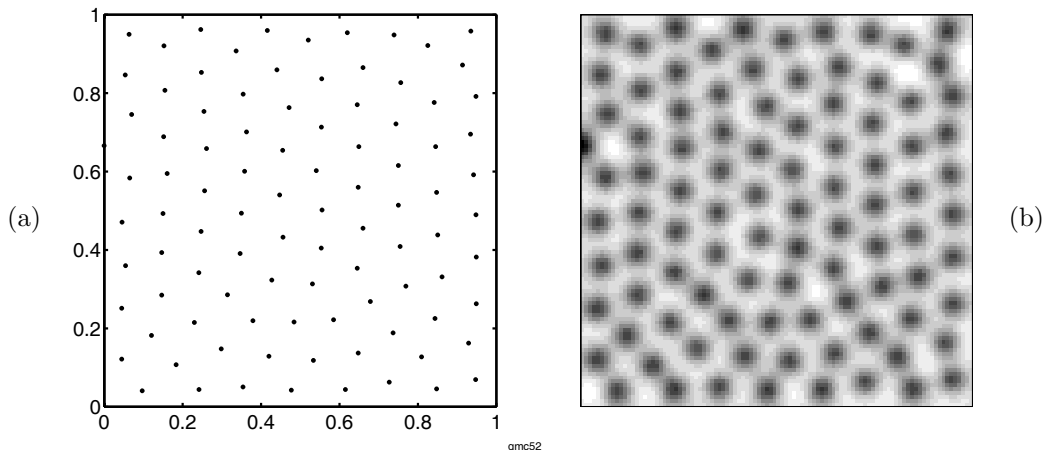


Figure 4. (a) This distribution of 100 points is obtained by minimizing the visual discrepancy from a uniform density level, and (b) the visualized point image, that is, the point image on the left blurred by the human visual system. The MVD algorithm is based on minimizing the variance in this blurred image. The rms deviation for the blurred image, relative to its mean value, is 1.1%.

values. In the last iteration of the prescribed number of iterations K , the final step length is normally about $0.1 a$.

An important aspect of the MVD algorithm is the specification and calculation of ψ , the function to be minimized. Through experimentation, I have found that the radial FWHM of the blur function Eq. (1) is best chosen to be on the order of the expected distance between points a . A value of $w \approx a/2$ seems to work well. Larger values do not keep points away from each other strongly enough. Another detail in the calculation of ψ concerns the convolution. The convolution operation redistributes intensities from inside the unit square to the outside, and visa versa. It therefore becomes necessary to specify the values of the image, not only inside the unit square, but also outside. In this study, the image is extended with a value equal to the average point density inside the unit square. Auxiliary runs show that this choice minimizes the error in the integration tests described in the next section. The convolution operations are performed using standard Fast Fourier Transform methods. The image sizes used in this study are typically 200×200 pixels to cover the unit square, and are extended to 256×256 pixels for the FFT.

The result of the blurring operation of the point (dot) image, stated in Eq. (6), is shown in Fig. 4. This point distribution is obtained by using the above optimization algorithm starting with the 100-point Halton sequence [2,3], shown in Fig. 3a. In this example, $a = 0.1$, and $w = 0.05$, which results in a radial profile with a FWHM of 0.0767. I have found that it is helpful to start with a point pattern that has relatively good coarse-scale uniformity. For example, starting with a Halton sequence of points or a stratified sample set¹¹ makes it easier to minimize ψ compared to starting with a random distribution. Similar observations have been made about the operation of centroidal Voronoi tessellation algorithms.¹⁶ Other examples of point sets obtained with MVD are presented in Fig. 5. Figure 5a looks very similar to Fig. 2c, the DBS example.

Minimizing ψ amounts to minimizing the variance in Fig. 4b. It is fairly clear that to obtain the lowest variance in the blurred image, the points must be distributed so no gaps exist in the pattern. Consequently, for the optimal pattern the points will be spread out as far apart and as uniformly as possible. One might expect regular patterns to minimize ψ . In fact, it is possible to identify regular patterns in the point sets produced by the MVD algorithm, as for example, in Figs. 4a and 5. The local patterns tend to be principally hexagonal, but do not possess a long-range order. The situation is similar to that in solid-state physics in which carefully fabricated materials produce single crystals with well-defined lattice structures. While the MVD and the DBS halftoning algorithm do not produce such regular arrays, there is definitely a tendency to do so. The MVD structures have defects similar to those seen in crystals, e.g., dislocations and interstitial defects. Actually, the patterns produced by the MVD and BDS algorithms more closely resemble glassy structures, obtained by quickly freezing a molten material, than crystalline structures.

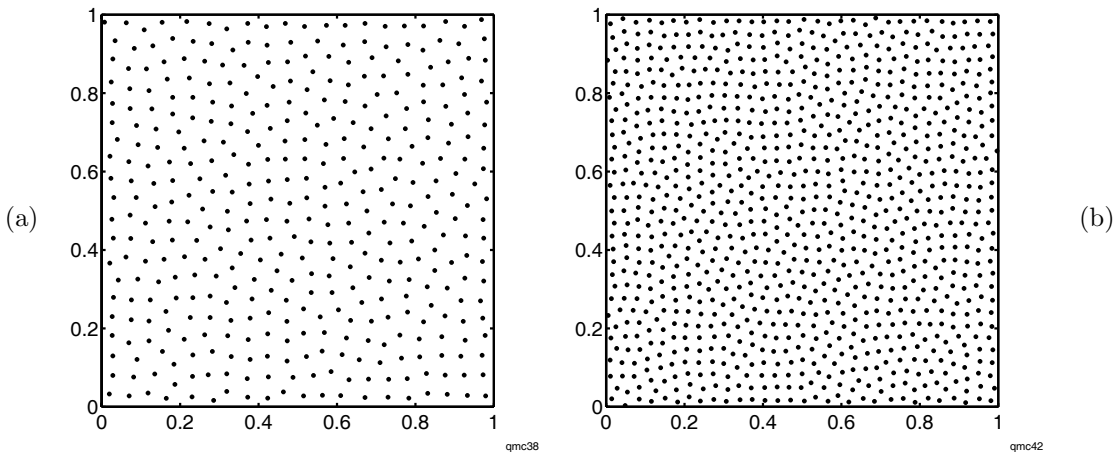


Figure 5. Point sets obtained with the MVD algorithm containing (a) 400 points and (b) 1000 points.

It should be kept in mind that the MVD algorithm tends to find a local minimum in ψ , not the global minimum. Single crystals represent the global minimum in energy. It may be possible to show that similar regular structures occur at the global minimum in ψ . Auxiliary tests indicate that regular square and hexagonal arrays of points tend to have the lowest observed values of ψ . However, initializing MVD with slightly perturbed versions of these patterns does not generally result in the same regular pattern being regained. It seems that irregularities in the point distribution create barriers that the simple downhill optimization algorithm presently used does not overcome. Other choices for the blur kernel or smaller values for w may make it easier to come closer to the global minimum. On the other hand, it is not clear that reaching the global minimum is desirable, as commented on in the Discussion section.

The optimization approach in MVD moves each point a limited distance. As with many iterative approaches to optimization, the high-frequencies in the image (corresponding to small scales) tend to be optimized at the beginning of the optimization procedure. The low frequencies (coarse scales) often take many iterations to be optimized. Thus, one might think that a multiscale approach to this optimization procedure could help speed up long-range adjustment of the point patterns to reach a suitable local minimum. For example, in the early iterations, one might use a relatively large value of w in the blur function (1) and move groups of neighboring points instead of single points. An alternative would be to start with point patterns that possess coarse-scale uniformity, for example, a QMC sequence or stratified sampling. Even a regular pattern, such as the square-grid pattern shown in Fig. 2d or a hexagonal pattern might be used to initialize MVD, provided that the positions of each point are randomly perturbed a suitable, but small, distance.

5. INTEGRATION TESTS

The performance of the various kinds of point sets for MC-like integration may be tested for selected functions. The following two separable functions are used in this study

$$\text{func1}(\mathbf{x}) = \prod_i |4(x_i - x_i^0)|, \quad (7)$$

$$\text{func2}(\mathbf{x}) = \prod_i \exp\{-2|x_i - x_i^0|\}, \quad (8)$$

where \mathbf{x} represents a position vector, \mathbf{x}^0 is the center of the distribution, and $i = 1, 2$ for two dimensions. The mean-square error is determined for the integral estimated using Eq. (4) by averaging the squared error over \mathbf{x}^0 values that uniformly cover the unit square. This average is accomplished using the standard MC technique and is based on 1000 randomly chosen \mathbf{x}^0 positions. The first function (7) is an inverted pyramid that rises linearly from zero at \mathbf{x}^0 . The second function (8) peaks at \mathbf{x}^0 , and gradually drops toward zero with a FWHM of 0.693 in both the x_1 and x_2 directions.

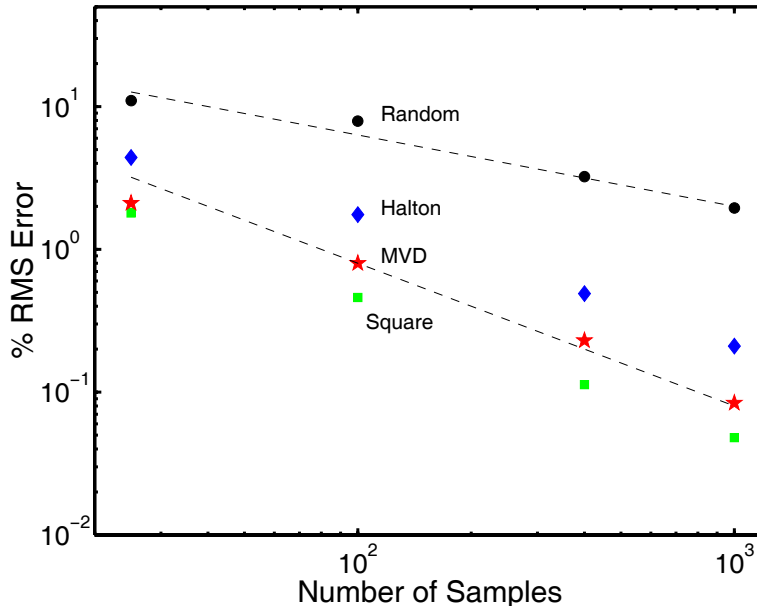


Figure 6. Plot of the per-cent rms error in Monte Carlo evaluations of integrals of the two-dimensional function `func2`, given in Eq. (8), for various kinds point sets versus the number of samples, N . The largest errors occur for the random sample set, shown as dots. The rms errors are smaller for the Halton sequence (diamonds), the minimum visual discrepancy (MVD) algorithm (stars), and the square grid (squares). The top line shows the convergence behavior of the rms error expected for standard MC, namely $N^{-1/2}$; the bottom line shows the behavior often achieved by quasi-MC sample sets, N^{-1} .

Figure 6 summarizes the results of the integration test of the function `func2` (8). The first and most important observation is that the random point set yields substantially poorer integration accuracy than the other point sets. Its rms error drops like $N^{-1/2}$, indicated by the upper line, as is expected for Monte Carlo integration. The Halton sequence, a mainstay in the QMC field, performs much better. The MVD algorithm yields rms errors that are generally better than those for the Halton sequence by at least a factor of two. For 100 points, MVD provides approximately ten times lower error than random sampling, and at 1000 samples it is at least 20 times better. The general dependence of the error for these three kinds of points sets is N^{-1} . For expensive function evaluations, these improvements represent substantial savings.

Table 1 summarizes the results of this study. The random and MVD sampling methods produce a different sampling pattern each time they are generated. Their entries in the table are, however, for a single realization for each point on the graph. The ranking of the rms errors for the various point sets consistently is, from largest to smallest, random, Halton, MVD, and square. The discrepancy D_2 , given in Eq. (5), does not correlate very well with the integration accuracy. In particular, the values for D_2 for the square pattern are always somewhat greater than those of the Halton and MVD sample sets, and yet the square pattern almost always produced the smallest rms error. Furthermore, D_2 is often smaller for the Halton sequence than for the MVD pattern, even though MVD always has smaller rms error. One can conclude that the D_2 discrepancy for a point set is not a very good indicator of its integration accuracy, at least for the functions considered here.

One way to look at Eq. (4) for MC integration is that each function value in the sum is representative of a volume element surrounding it. A useful way to partition the domain of integration into neighborhoods surrounding each sample point is through Voronoi analysis.^{16–18} A simple, if somewhat unorthodox way to perform Voronoi analysis, is based on a Monte Carlo technique. One randomly draws a set of K points $\{\mathbf{X}_k\}$ from a uniform distribution defined over the domain of interest. The distance is computed between each point \mathbf{X}_k and each point in the sample set $\{\mathbf{Z}_j\}$ being analyzed. It is assigned to the closest \mathbf{Z}_j . The set of points that are closest to \mathbf{Z}_j belong to its Voronoi region, the number of which, divided by K , is an estimate of the fractional volume of that Voronoi region.

Table 1. Results of integration tests for func1 and func2 for various kinds of point sets. The columns list the type of point set, the number of samples N , the % rms error for integrals of func1 and func2, N times the discrepancy D_2 , and the % rms deviation in the areas of the Voronoi polygons. The relative accuracy of the rms integration error entries is about 2%.

Sample set	N	% rms error		ND_2	% rms dev in V areas
		func1	func2		
Random	25	19.7	11.0	1.75	61.
Halton	25	8.6	4.4	0.92	20.
MVD	25	5.4	2.1	0.70	12.
Square	25	2.5	1.8	1.19	0.4
Random	100	14.9	7.92	5.60	60.
Halton	100	2.6	1.75	1.29	26.
MVD	100	1.8	0.80	1.01	10.
Square	100	0.6	0.46	2.36	1.
Random	400	6.25	3.23	8.44	58.
Halton	400	0.73	0.49	1.52	22.
MVD	400	0.47	0.23	1.96	6.
Square	400	0.16	0.11	4.72	2.
Random	1000	2.37	1.95	8.49	55.
Halton	1000	0.31	0.21	1.52	25.
MVD	946	0.17	0.08	3.22	13.
Square	961	0.07	0.05	7.60	3.

From the equal weighting of the function value at each sample point in Eq. (4), one would suppose that for a good sample set, the Voronoi areas should all be about equal. The relative rms deviations of the Voronoi areas, calculated on the basis of 10^6 samples, are listed in Table 1. It is seen that the rank order for the rms deviation in the Voronoi areas is consistently the same as for the rms error. However, in further tests involving other types of point sets and other integration test functions, the correlation between these two quantities is not upheld. Thus, the uniformity of the Voronoi areas does not seem to be uniquely related to the integration error. In addition, one might expect that the average second moment of the radius should be minimum for a point set with good sampling properties. This condition is attained with a centroidal Voronoi tessellation.^{16, 17} Unfortunately, this quantity does not seem to be correlated with integration error either.

6. DISCUSSION

The ultimate aim of the present work is to improve on present techniques for analyzing the sensitivity of computer models outputs to numerous input parameters. The goal therefore is to develop point sets in high dimensions, which possess desirable convergence properties for integration estimates, comparable to, or better than QMC sequences. As argued earlier, it seems desirable for the sample points to be spatially distributed somewhat randomly. The sample sets should also be independent from one sequence to the next. Another desirable trait is for the sample sets to be easily augmentable, so that additional points may be easily generated, if deemed necessary.

The MVD algorithm, based on the ideas behind halftoning has been shown to achieve very good performance for integrating two simple functions. On the other hand, the MVD algorithm, per se, may be impossible to

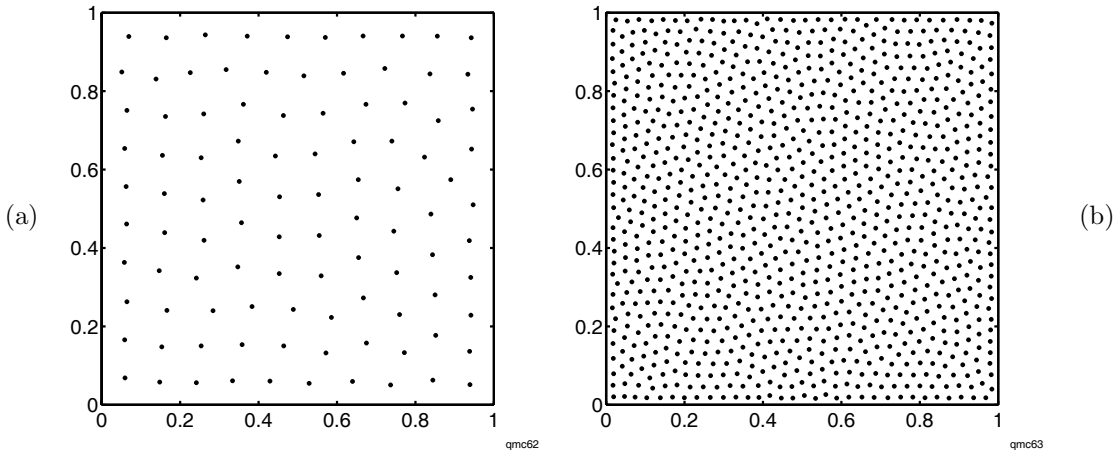


Figure 7. Examples obtained with the potential-field algorithm for (a) 100 points and (b) 1000 points.

implement in high dimensions because it is based on determining the mean-square error, Eq. (6), of an M -dimensional image. This implies the necessity for storing a discretized image in M dimensions, which may be infeasible when M gets larger than four or five, even when coarsely discretized in each dimension. The convolution is not a problem, because with fast Fourier transforms, the cost grows only linearly with M .

Another approach to generating a suitable point set, which is closely related to MVD, is to draw an analogy between the point set and a collection of particles, which interact by means of a potential field. The potential field can be chosen so that the particles repel each other at close distances, but are less repulsive when they are sufficiently far apart. This type of action occurs in the MVD approach, although it is not explicit. Appropriate conditions need to be specified at the boundary of the region. The advantage of this potential-field approach is that an integral over the M -dimensional domain is not required to evaluate the cost function. To calculate the total energy of a specific configuration of the particles, the distances between each point and every other point needs to be calculated, an order- MN^2 calculation for N points in M dimensions, which is quite manageable. On the other hand, to determine the change in energy for optimizing each point, the calculation is only of order MN .

Initial results indicate that this potential-field approach is promising. Figure 7 shows the results obtained with a potential with the same form used in the DBS algorithm, given in Eq. (1)

$$\zeta = \sum_{j=1}^N \sum_{i=j+1}^N (w^2 + |\mathbf{x}_i - \mathbf{x}_j|^2)^{-3/2}, \quad (9)$$

with suitable boundary conditions. The indices i and j label the points in the sequence. These patterns are visually very similar to those produced by MVD, shown in Fig. 4a and Fig. 5b. The rms errors for integrating `func2` with these two point sets are 0.96% for 100 points and 0.17% for 1000 points, which is competitive with MVD. This potential-field approach is somewhat akin to the ‘springs’ idea of Atkins et al.¹⁹

A variation on the problem of creating a point set with a fixed number of points is to create them in a sequential manner. The scheme is to add one point at a time, keeping fixed all previous points. In the context of analyzing computer models, it can occur that after a specified number of parameter sets have been calculated, one wishes to augment that set with further calculation. This requirement to add further points to an existing set is not readily possible for some approaches, for example, Latin hypercube sampling,¹⁵ and stratified sampling,¹ which rely on prior specification of the number of points. It is perfectly feasible, however, for MC and QMC sequences. It may be possible for the MVD algorithm, as well as for the potential field method, described above. With both of these methods, it might be necessary to leave small gaps in the pattern as it develops so that new points can be added later by filling in the gaps. This process might be easier in high dimensions. An investigation is needed to characterize the tradeoff between the degree of randomness in point sets and integration accuracy.

It might be reasonable to optimize the placement of each new point in a less-than-perfect way, thereby leaving some irregularity in the point patterns.

There is clearly a need for a metric that is correlated to the integration accuracy for any specific point set. The discrepancy D_2 is often used in the QMC field to define upper limits to the accuracies of various classes of functions. The results obtained here indicate that D_2 , by itself, does not follow the integration accuracy of a point pattern. Voronoi analysis may be useful for generating QMC point sets, but the metrics derived using Voronoi analysis in this study do not seem to be correlated either. Besides the relative rms deviation in the Voronoi areas listed in Table 1, the first and second moments of the radius, calculated for each Voronoi polygon were considered, and these do not seem to be correlated with integration error.

ACKNOWLEDGMENTS

I appreciatively acknowledge the discussions I have had with Jan Allebach, Charlie Bouman, Tony Warnock, Paul Pederson, Luc Wille, Dave Robinson, Tony Giunta, Dave Higdon, Mike McKay, and Tom Booth.

REFERENCES

1. M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods: Vol. 1, Basics*, John Wiley, New York, 1986.
2. H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, 1992.
3. J. Sullivan, L. Ray, and R. Miller, "Design of minimum visual modulation halftone patterns," *IEEE Trans. Sys., Man, Cyber.* **21**, pp. 33–38, 1991.
4. D. L. Lau and G. R. Arce, "Digital halftoning," in *Nonlinear Image Processing*, S. K. Mitra and G. L. Sicuranza, eds., pp. 375–401, Academic, San Diego, 2001.
5. M. D. McKay, "Nonparameteric variance-based methods of assessing uncertainty importance," *Reliab. Eng. Syst. Safety* **57**, pp. 267–279, 1997.
6. A. Saltelli, K. Chan, and E. M. Scott, *Sensitivity Analysis*, John Wiley, Chichester, 2000.
7. M. Analoui and J. P. Allebach, "Model-based halftoning using direct binary search," in *Human Vision, Visual Processing, and Digital Display III*, B. E. Rogowitz, ed., *Proc. SPIE* **1666**, pp. 96–108, 1992.
8. P. Li and J. P. Allebach, "Look-up-table-based halftoning algorithm," *IEEE Trans. Image Proc.* **9**, pp. 1593–1603, 2000.
9. S. H. Kim and J. P. Allebach, "Impact of HVS models on model-based halftoning," *IEEE Trans. Image Proc.* **11**, pp. 258–269, 2002.
10. H. Niederreiter, *Monte Carlo and Quasi-Monte Carlo Methods 1998*, Springer, Berlin, 1998.
11. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++*, Cambridge University, New York, 2002.
12. T. T. Warnock, "Computational investigations of low-discrepancy point sets II," in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, H. Niederreiter, ed., pp. 354–361, Springer, New York, 1995.
13. M. Junk, "Matlab source code for QMC sequences," 2000. On web: <http://www.mathematik.uni-kl.de/~junk/highdim/sources.html>.
14. R. Ohbuchi and M. Aono, "Quasi-Monte Carlo rendering with adaptive sampling," 1996. On web: http://www.kki.yamanashi.ac.jp/~ohbuchi/online_pubs/eg96_html/eg96.htm.
15. M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods of selecting values of input variables in the analysis of output from a computer code," *Technometrics* **21**, pp. 239–245, 1979.
16. Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: applications and algorithms," *SIAM Rev.* **41**, pp. 637–676, 1999.
17. L. Ju, Q. Du, and M. Gunzburger, "Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations," *Parallel Comp.* **28**, pp. 1477–1500, 2002.
18. A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley, Chichester, 2000.
19. C. B. Atkins, J. P. Allebach, and C. A. Bouman, "Halftone postprocessing for improved rendition of high-lights and shadows," *J. Elec. Imaging* **9**, pp. 151–158, 2000.