Introduction to Monte Carlo and Quasi-Monte Carlo methods

Alvise Sommariva

Padua, autumn, 2024

Doctoral Program in Mathematical Sciences, Padua (I), Autumn 2024

"Numerical cubature and its applications"

In this note we intend to give a very short introduction to Monte Carlo and quasi Monte Carlo, in which we just outline its ideas, problems and the basic theory.

There are many books, primers and videos that one may find in its huge literature and on the web.

Between them, some good resources are surely

- R.E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, Acta Numerica, 1998, pp. 1–48.
- J.Dick, F.Y. Kuo, I.H. Sloan, J. High dimensional integration – the Quasi-Monte Carlo way, Acta Numerica, 2014, pp. 1–157.
- P.J. Davis and P.Rabinowitz, Methods of Numerical Integration, Dover 1984.
- I.M. Sobol, A Primer For The Monte Carlo Method, CRC press 1994.

The task of Monte Carlo type methods is to approximate numerical integrals on the hypercube $[0,1]^s$ of the form

$$I_s(f) = \int_{[0,1]^s} f(\mathbf{x}) d\mathbf{x}, \quad f \in C([0,1]^s)$$

via n-point integration rule of the form

$$Q_{n,s} = \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{t}_i)$$

in which one uses prescribed samples $\mathbf{t}_i \in [0,1]^s$, $i = 1, \ldots, n$.

Following [4],

> The generally accepted birth date of the Monte Carlo method is 1949, when an article entitled 'The Monte Carlo method" by Metropolis and Ulam appeared.
>
> The American mathematicians John von Neumann and Stanislav Ulam are considered its main originators. In the Soviet Union, the first papers on the Monte Carlo method were published in 1955 and 1956 by V. V. Chavchanidze, Yu. A. Shreider and V. S. Vladimirov.
>
> Curiously enough, the theoretical foundation of the method had been known long before the von Neumann-Ulam article was published.

As we will see, these techniques are useful in several instances. Here we will focus our attention on the following topics.

1. Integration domains of $\mathbb{R}^2$ or $\mathbb{R}^3$ with difficult geometry.
2. High dimensional integration. About the latter, following [2], p.4:

> Applications have also played an important role in the development of QMC methods suitable for high dimensional problems. In financial mathematics, the numerical experiments of Paskov and Traub (1995), which used low-discrepancy QMC methods in 360 dimensions to value parcels of mortgage-backed obligations, were successful to a degree that caused universal surprise. ...
>
> Option pricing problems have spurred many developments. ...
>
> The problem of evaluating high dimensional expected values arising from partial differential equations with random coefficients, typified by the flow of a liquid (oil or water) through a porous material, with the permeability treated as a random field, is the newest driver of innovation.

We start with the first problem previously mentioned and intend to numerically approximate

$$\int_{\Omega} f(x)d\Omega \approx \sum_{i=1}^{N} w_i f(P_i).$$

where

- $\Omega \subseteq [0,1]^s$, where $s = 2, 3$ (e.g. a sphere, a polygon, a polyhedron, etc.),
- $f \in C(\Omega)$,
- are available the samplings $f(P_i)$, $i = 1, \ldots, N$ at the scattered data $P_1, \ldots, P_N \in \Omega$.

In the case the set of nodes $\{P_i\}_{i=1,\dots,N}$ is a subset of sequence of points that is uniformly distributed in Ω, the classical approach of Monte Carlo-type methods gives

$$\int_{\Omega} f(x) d\Omega \approx \frac{\mu(\Omega)}{N} \sum_{i=1}^{N} f(P_i).$$

where $\mu(\Omega)$ is the measure of the domain Ω (or an approximation).

Notice that the weights are all equal to $\frac{\mu(\Omega)}{N}$.

A typical case is that Ω is defined by set operations over

$$\Omega_1, \ldots, \Omega_M,$$

e.g. $\cap_{i=1}^M \Omega_i$ or $\cup_{i=1}^M \Omega_i$ (see figures in the next page).

A basic approach is to

- determine a hyper-rectangle $\mathcal{R}$ containing Ω;
- define an uniformly distributed sequence $X_{\mathcal{R}}^*$ on $\mathcal{R}$;
- take the first N^* points $X_{\mathcal{R}}$ of $X_{\mathcal{R}}^*$ (usually N^* is very large);
- determine the sequence of points of $X_{\mathcal{R}}^{(i)} \subseteq X_{\mathcal{R}}$ belonging to Ω_i, $i = 1, \ldots, M$ (*in-domain* functions on each Ω_i must be available, it may not be a trivial task!);
- determine from these $X_{\mathcal{R}}^{(i)}$, $i = 1, \ldots, M$ the required sequence X_Ω on Ω as well as an approximation of $\mu(\Omega)$;
- choose N points $\{P_i\}_{i=1,\ldots,N}$ in X_Ω and from samplings of f compute $\int_\Omega f(x) d\Omega \approx \frac{\mu(\Omega)}{N} \sum_{i=1}^N f(P_i)$.

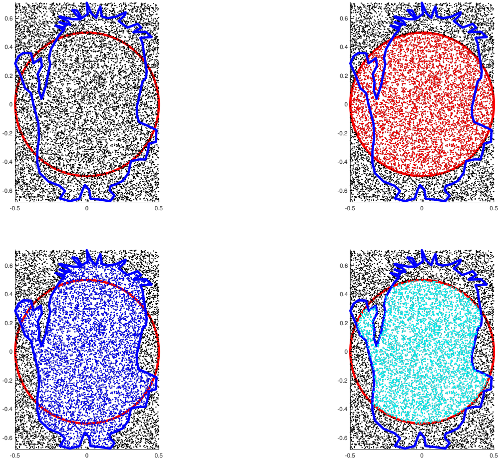

Figure: Intersection Ω of a polygonal minion with a disk. $N^* = 10000$ points in the bounding box $\mathcal{R} \approx [-0.5000, 0.5000] \times [-0.6741, 0.7077]$ of which $N = 4741$ are in Ω. Thus $\mu(\Omega) \approx \mu(\mathcal{R}) \cdot 4741/10000 \approx 0.6551$.

One immediately notices how relevant are indomains routines. If they are available, these methods may provide results even in complicated geometries without tracking the boundaries.

Unfortunately, it is noticed that the convergence may be really slow, depending on the point-set. We will explore this problem in general, providing estimates and some ideas to mitigate this problem.

In particular we will consider the

- convergence of the classical Monte Carlo method based on random points,
- basic results on Quasi-Monte Carlo methods, that in general provide a better convergence rate.

Suppose that $Q_{n,s} = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{t}_i)$ in which $\mathbf{t}_1, \ldots, \mathbf{t}_n$ are i.i.d. (independent and identically distrubuted) uniform random samples from $[0,1]^s$.

Theorem (Monte Carlo probabilistic estimates, [2] p.6)

For all continuous and square integrable functions f, denoting by

- $\mathbb{E}$ *the expectation w.r.t. the uniform random samples* $\mathbf{t}_1, \ldots, \mathbf{t}_n$,
- $\sigma^2(f) := I_s(f^2) - (I_s(f))^2$ *the variance of* f,

we have that

1. $\mathbb{E}[Q_{n,s}(f)] = I_s(f)$;
2. $\sqrt{\mathbb{E}[|I_s(f) - Q_{n,s}(f)|^2]} = \frac{\sigma(f)}{\sqrt{n}}$;

Furthermore, by the central limit theorem, if $0 < \sigma(f) < +\infty$ *then*

$$\lim_n \mathbb{P}\left(|I_s(f) - Q_{n,s}(f)| \leq c\frac{\sigma(f)}{\sqrt{n}}\right) = \frac{1}{\sqrt{2\pi}}\int_{-c}^{c} \exp(-x^2/2)dx.$$

From

$$\lim_n \mathbb{P}\left(|I_s(f) - Q_{n,s}(f)| \leq c\frac{\sigma(f)}{\sqrt{n}}\right) = \frac{1}{\sqrt{2\pi}}\int_{-c}^{c} \exp(-x^2/2)dx.$$

just to make some examples (see [3], p. 387), denoting by

$$E_n = |I_s(f) - Q_{n,s}(f)|,$$

we have that

- Error E_n at 50% level of probability $\leq \frac{.6745\sigma}{\sqrt{n}}$;
- Error E_n at 90% level of probability $\leq \frac{1.645\sigma}{\sqrt{n}}$;
- Error E_n at 95% level of probability $\leq \frac{1.960\sigma}{\sqrt{n}}$;
- Error E_n at 99.99% level of probability $\leq \frac{3.891\sigma}{\sqrt{n}}$.

In view of these results

- we have a probabilistic error bound with a convergence rate of $O(n^{-1/2}$;
- pros: it is independent on s (dimension of the domain);
- cons: the convergence of the method is slow;
- variance reduction techniques as
  - importance sampling,
  - stratified sampling,
  - correlated sampling,

  can be used to improve the efficiency of MC, but in practice MC methods often remain distressingly slow [2].
- Bakhvalov (1959) proved that the $\mathcal{O}(n^{-1/2})$ rate of convergence cannot be improved for general square-integrable or continuous functions f.

Thus for functions with more smoothness, this slow convergence rate is the main motivation for switching to Quasi-Monte Carlo methods where specific pointsets are choosen.

As numerical test, we intend to compute

$$I_1(\exp(-x^2)) = \int_0^1 \exp(-x^2)dx = \frac{\sqrt{2}}{\pi}\texttt{erf}(x) \approx 0.7468241328124270.$$

- Having in mind to see the behaviour of MC approximation, we perform 1000 tests, for $n = 2, 4, 8, 16, 32, \ldots, 262144$.
- For each n, we consider the average error

$$E_n = \mathcal{E}(|I_1(\exp(-x^2)) - Q_{n,1}^k|),$$

  being $Q_{n,1}^k$ the result obtained at the k-th experiment, using n random points.
- Finally, supposing $E_n \approx \frac{C}{\sqrt{n}}$, remembering that for successive n we doubled the cardinality, we check that

$$E_n/E_{2n} \approx \frac{\frac{C}{\sqrt{n}}}{\frac{C}{\sqrt{2n}}} = \sqrt{2} \approx 1.41.$$

To this purpose, we implement the following routine.

```
function demo_montecarlo_1D

NV=2.^(1:18);
f=@(x) exp(-x.^2);
I=integral(f,0,1,"AbsTol",10^(-15));
trials=1000;
AE_mat=[];

for k=1:trials
    AE=[];
    for n=NV
        P=rand(n,1);
        fP=feval(f,P);
        In=sum(fP)/n;
        AE(end+1,1)=abs(I-In);
    end
    AE_mat(:,k)=AE;
end

AE_mat_aver=abs(sum(AE_mat,2)/trials);

for k=1:length(NV)
    fprintf('\n \t n: %7.0f ae: %1.5e',NV(k),AE_mat_aver(k));
end

rat=AE_mat_aver(1:end-1)./AE_mat_aver(2:end)
```

Monte Carlo type methods probabilistic estimates

```
n:       2 ae: 1.13402e-01
n:       4 ae: 8.19724e-02
n:       8 ae: 5.90765e-02
n:      16 ae: 3.93856e-02
n:      32 ae: 2.91128e-02
n:      64 ae: 1.95096e-02
n:     128 ae: 1.47442e-02
n:     256 ae: 9.88877e-03
n:     512 ae: 7.03790e-03
n:    1024 ae: 5.04232e-03
n:    2048 ae: 3.65921e-03
n:    4096 ae: 2.46004e-03
n:    8192 ae: 1.83284e-03
n:   16384 ae: 1.20116e-03
n:   32768 ae: 8.74813e-04
n:   65536 ae: 6.08675e-04
n:  131072 ae: 4.36691e-04
n:  262144 ae: 3.18900e-04
```

```
1.383414640260222e+00
1.387562174586486e+00
1.499954410471419e+00
1.352859677566526e+00
1.492228718527352e+00
1.323204146027808e+00
1.491006713101362e+00
1.405075333291693e+00
1.395765872517404e+00
1.377981147155563e+00
1.487457819785931e+00
1.342202891565608e+00
1.525886629314394e+00
1.373050487924346e+00
1.437241667172072e+00
1.393835577072824e+00
1.369365975075068e+00
```

Figure: Numerical experiments by MC on a certain 1D test case. On the left, MC average errors. On the right: ratio (close to $\sqrt{2}$, indicating an average convergence of the order $\mathcal{O}(n^{-1/2})$)

We implement the following routine, very similar to the 1D version.

```
function demo_montecarlo_2D

NV=2.^(1:14);
f=@(x,y) exp(-x.^2-y.^2);
I=(sqrt(pi)*erf(1)/2)^2;
trials=1000;
AE_mat=[];

for k=1:trials
    AE=[];
    for n=NV
        P=rand(n,2);
        fP=feval(f,P(:,1),P(:,2));
        In=sum(fP)/n;
        AE(end+1,1)=abs(I-In);
    end
    AE_mat(:,k)=AE;
end

AE_mat_aver=abs(sum(AE_mat,2)/trials);

for k=1:length(NV)
    fprintf('\n \t n: %7.0f ae: %1.5e',NV(k),AE_mat_aver(k));
end

AE_mat_aver(1:end-1)./AE_mat_aver(2:end)
```

Monte Carlo type methods probabilistic estimates

```
n:       2 ae: 1.19340e-01
n:       4 ae: 9.01878e-02   1.323234457390696e+00
n:       8 ae: 6.23949e-02   1.445435735698849e+00
n:      16 ae: 4.35506e-02   1.432700286990400e+00
n:      32 ae: 3.13565e-02   1.388885699920435e+00
n:      64 ae: 2.16561e-02   1.447927120904793e+00
n:     128 ae: 1.48008e-02   1.463167859391284e+00
n:     256 ae: 1.08091e-02   1.369298561281823e+00
n:     512 ae: 7.51971e-03   1.437431877336634e+00
n:    1024 ae: 5.35942e-03   1.403082905834149e+00
n:    2048 ae: 3.80363e-03   1.409028470921343e+00
n:    4096 ae: 2.64100e-03   1.440219702191760e+00
n:    8192 ae: 1.95710e-03   1.349445736255882e+00
n:   16384 ae: 1.36272e-03   1.436174079246958e+00
n:   32768 ae: 9.65672e-04   1.411162095897052e+00
n:   65536 ae: 6.51479e-04   1.482276738557081e+00
n:  131072 ae: 4.74735e-04   1.372301647812137e+00
n:  262144 ae: 3.31464e-04   1.432237447795883e+00
```

Figure: Numerical experiments by MC on a certain 2D test case. On the left, MC average errors. On the right: ratio (close to $\sqrt{2}$, indicating an average convergence of the order $\mathcal{O}(n^{-1/2})$)

As second numerical test, we intend to compute

$$I_2(\exp(-x^2-y^2)) = \int_0^1 \exp(-x^2-y^2)dxdy = \frac{2}{\pi^2}\texttt{erf}^2(x) \approx 0.557746285.$$

- Having in mind to see the behaviour of MC approximation, we perform 1000 tests, for $n = 2, 4, 8, 16, 32, \ldots, 262144$.
- For each n, we consider the average error

  $$E_n = \mathcal{E}(|I_1(\exp(-x^2)) - Q^k_{n,2}|),$$

  being $Q^k_{n,2}$ the result obtained at the k-th experiment, using n random points in $[0, 1]^2$.
- Finally, supposing $E_n \approx \frac{C}{\sqrt{n}}$, remembering that for successive n we doubled the cardinality, we check that

  $$E_n/E_{2n} \approx \frac{\frac{C}{\sqrt{n}}}{\frac{C}{\sqrt{2n}}} = \sqrt{2} \approx 1.41.$$

Following Sobol [4], p. 97,

In 1916, H. Weyl found that infinite sequences of non-random points $Q_1, Q_2, \ldots, Q_n \ldots$ exist, which have a property similar to MC: for an arbitrary Riemann-integrable function $f(x_1, \ldots, x_s)$

$$\int_0^1 \ldots \int_0^1 f(x_1, \ldots, x_s) dx_1 \ldots dx_s = \lim_n \frac{1}{n} \sum_{j=1}^{n} f(Q_j)$$

Such sequences are said to be uniformly distributed in the number-theoretical sense.

Next Sobol says

1. The uniformity of distribution should be optimal as $n \to +\infty$,
2. the uniformity of distribution of initial points $Q_1, Q_2, \ldots, Q_n$ should be observed for very small n;
3. formulas for computing these points should be simple.

Referring to [2]:

The key idea is that $\mathbf{t}_1, \ldots, \mathbf{t}_n \in [0,1]^s$ are chosen so that chosen deterministically to be better than random, in the sense that the deterministic nature of QMC leads to guaranteed error bounds, and that the convergence rate may be faster than the MC rate of $\mathcal{O}(n^{-1/2})$ for sufficiently smooth functions.

There are two types of QMC methods:

- The "open" type: this uses the first n points of an infinite sequence. Thus to increase n one only needs to evaluate the integrand at the additional cubature points.
- The "closed" type: this uses a finite point set which depends on n. Thus a new value of n means a completely new set of cubature points.

In literature there are many examples of QMC methods, e.g.

- Van der Corput sequence;
- Halton sequence (1960);
- Hammersley point set;
- Kronecker sequence;
- Sobol sequence;

Lately many efforts have been done to discover the properties of

- Lattice rules;
- Digital nets and sequences (example: Sobol sequence (1967)).

and it is not possible to describe in short all their interesting properties, in view of the massive research on the field.

In the following, being

$$\mathbf{x} = (x_i)_{i=1,\ldots,d} \in \mathbb{R}^d,$$

we define the hyperrectangle

$$[0,\mathbf{x}] := [0,x_1] \times \ldots [0,x_d].$$

Definition (Star discrepancy)

The local discrepancy function Δ_P is defined as

$$\Delta_P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} 1_{[0,\mathbf{x}]}(\mathbf{t}_i) - \int_{[0,\mathbf{x}]} 1_{[0,\mathbf{x}]}(y)dy = \frac{1}{n}\sum_{i=1}^{n} 1_{[0,\mathbf{x}]}(\mathbf{t}_i) - \prod_{i=1}^{d} x_i.$$

The star-discrepancy of a set $P = \{x_1, \ldots, x_N\}$ is defined as

$$D_N^*(P) = \sup_{x\in\mathbb{R}^d} |\Delta_P(x)|$$

Intuitively, in 1D, the star discrepancy is the proportion of points of $P = \{t_i\}_{i=1,\dots,N}$ lying in the interval $[0, x]$ minus the length of the interval, representing somehow as written in [2], p.42:

> .. the departure of this proportion from the “ideal ”proportion, which is the length of the interval.
>
> The star-discrepancy can therefore be understood as a measure for how uniformly the point set P is distributed, i.e., it measures the discrepancy between the empirical distribution of the point set P and the uniform distribution.

Definition (variation in the Hardy-Krause sense)

The variation in the Hardy-Krause sense of $f : [0, 1] \to \mathbb{R}$ is

$$V[f] = \int_0^1 \left| \frac{df}{dt} \right| dt$$

The variation in the Hardy-Krause sense of $f : [0, 1]^d \to \mathbb{R}$ is

$$V[f] = \int_{[0,1]^d} \left| \frac{\partial^d f}{\partial t_1 \dots \partial t_d} \right| dt_1 \dots dt_d + \sum_{i=1}^{d} V[f_1^{(i)}]$$

where $f_1^{(i)}$ is the restriction of the function to the boundary $x_i = 1$.

Observe that in the definition above, since $f_1^{(i)}$ are restrictions involving $d - 1$ variables, the definition is recursive.

Theorem (Koksma-Hlawka)

For

- *any sequence* $\{\mathbf{t}_i\}$,
- *any function* f *with bounded variation in the Hardy-Krause sense,*

the integration error is such that

$$\left|\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{t}_i) - \int_{[0,1]^d} f(y)dy\right| \leq D_N^* V[f].$$

We make some remarks on the previous inequality.

1. The error bound

$$E_N(f) = \left| \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{t}_i) - \int_{[0,1]^d} f(y)dy \right| \leq D_N^* \, V[f].$$

 depends on a factor concerning the sequence and one on the function;
2. one can prove that it is a worst case bound;
3. in general $V[f]$ in Koksma-Hlawka is *usually a gross overestimate, while the discrepancy is indicative of the actual performance* (see [1], p.26).

An effect of Koksma-Hlawka inequality, knowing some inequalities about star-discrepancy, is that we have that

- for Hammersley pointset

$$E_N(f) \leq C' \frac{(\log(N))^{d-1}}{N} V[f],$$

  with C' depending only on the dimension d;

- for Halton sequence

$$E_N(f) \leq C \frac{(\log(N))^{d}}{N} V[f],$$

  with C depending only on the dimension d.

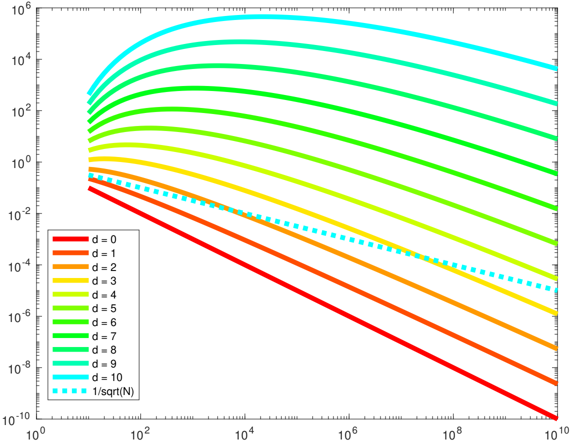

Figure: The graphics of $(\log(N))^d/N$ (straight lines, varying d), compared to $1/\sqrt{N}$ (dotted line). Advantage of QMC w.r.t. MC, in high dimensional problems, is not guaranteed by estimates.

Remark

It is known that for pointsets with N points

- *dimension 1: $D_N^* \geq 0.5/N$;*
- *dimension 2: $D_N^* \geq 0.023 \log(N)/N$;*
- *dimension $d \geq 3$: open problem.*

Conjecture for dimension d for sequences:

$$D_N^* \geq c_d (\log(N))^d / N$$

for some $c_d > 0$.

Those achieving this order are named low-discrepancy sequences.

As numerical comparison in 1D, we run some tests of some QMC methods w.r.t. MC (using Matlab built-in `rand`).

We made our test in the less regular case of the integrand

$$f(x, y) = \exp\left(-x^2\right)$$

over the unit interval $[0, 1]$.

The reference value is $I = \sqrt{\pi}\,\mathrm{erf}(1)/2 \approx 0.7468241328124270$.

The QMC methods are

- Hammersley pointset,
- Halton sequence,
- Sobol sequence.

with the same meaning of the previous tables, that is averaging results over 1000 tests.

In our 1D tests we propose just one table, since for $N = 2, 4, 8, \ldots, 262144$ the pointsets produced by Halton sequence, Sobol sequence and Hammersley pointset are actually the same.

```
>> Pset = haltonset(1);
>> PHL=Pset(1:2^6,:);
>> Pset = sobolset(1);
>> PSB=Pset(1:2^6,:);
>> P = Hammersley(2^6,1); PHM=P';
>> norm(PHL-PSB)
ans =
     0
>> norm(PHL-PHM)
ans =
    3.0272
>> norm(sort(PHL)-sort(PHM))
ans =
     0
>>
```

Notice that in the case of Hammersley pointset, for $N = 2^6$, the pointset is the same but the order of the points is different.

```
function demo_QMC_1D

NV=2.^(1:18);
f=@(x) exp(-x.^2);
I=sqrt(pi)*erf(1)/2;
trials=1000;
AE_mat=[];

QMC_type=2;

for k=1:trials
    AE=[];
    for n=NV
        P=QMC_pointset(QMC_type,n);
        fP=feval(f,P);
        In=sum(fP)/n;
        AE(end+1,1)=abs(I-In);
    end
    AE_mat(:,k)=AE;
end

AE_mat_aver=abs(sum(AE_mat,2)/trials);

fprintf('\n \n \t \t * AVERAGE ABSOLUTE ERRORS \n \n');
for k=1:length(NV)
    fprintf('\n \t n: %7.0f ae: %1.5e',NV(k),AE_mat_aver(k));
end

fprintf('\n \n \t \t \t * AVERAGE RATIOS \n \n');
rat=AE_mat_aver(1:end-1)./AE_mat_aver(2:end);
for k=1:length(NV)-1
    fprintf('\n \t n1: %7.0f n2: %7.0f rat: %1.5e',NV(k),NV(k+1),rat(k));
end
```

The routine `QMC_pointset` is described as follows.

```
function P=QMC_pointset(QMC_type,n)

switch QMC_type

    case 0
        P=rand(n,1);

    case 1

        Pset = sobolset(1);
        P=Pset(1:n,:);

    case 2
        Pset = haltonset(1);
        P=Pset(1:n,:);

    case 3
        P = Hammersley(n,1); P=P';

end
```

The routines `sobolset` and `haltonset` are Matlab built-in, while Hammersley must be downloaded.

```
* AVERAGE ABSOLUTE ERRORS

n:       2 ae: 1.42576e-01
n:       4 ae: 7.51750e-02
n:       8 ae: 3.85490e-02
n:      16 ae: 1.95142e-02
n:      32 ae: 9.81701e-03
n:      64 ae: 4.92347e-03
n:     128 ae: 2.46548e-03
n:     256 ae: 1.23367e-03
n:     512 ae: 6.17071e-04
n:    1024 ae: 3.08594e-04
n:    2048 ae: 1.54312e-04
n:    4096 ae: 7.71595e-05
n:    8192 ae: 3.85807e-05
n:   16384 ae: 1.92906e-05
n:   32768 ae: 9.64534e-06
n:   65536 ae: 4.82268e-06
n:  131072 ae: 2.41134e-06
n:  262144 ae: 1.20567e-06
```

```
* AVERAGE RATIOS

n1:       2 n2:       4 rat: 1.89659e+00
n1:       4 n2:       8 rat: 1.95012e+00
n1:       8 n2:      16 rat: 1.97543e+00
n1:      16 n2:      32 rat: 1.98780e+00
n1:      32 n2:      64 rat: 1.99392e+00
n1:      64 n2:     128 rat: 1.99696e+00
n1:     128 n2:     256 rat: 1.99848e+00
n1:     256 n2:     512 rat: 1.99924e+00
n1:     512 n2:    1024 rat: 1.99962e+00
n1:    1024 n2:    2048 rat: 1.99981e+00
n1:    2048 n2:    4096 rat: 1.99991e+00
n1:    4096 n2:    8192 rat: 1.99995e+00
n1:    8192 n2:   16384 rat: 1.99998e+00
n1:   16384 n2:   32768 rat: 1.99999e+00
n1:   32768 n2:   65536 rat: 1.99999e+00
n1:   65536 n2:  131072 rat: 2.00000e+00
n1:  131072 n2:  262144 rat: 2.00000e+00
```

Figure: Numerical integration of $\int_0^1 \exp\left(-x^2\right) dx$ by means of QMC based on Halton sequence, Sobol sequence and Hammersley pointsets.

As first numerical comparison in 2D, we run some tests of some QMC methods w.r.t. MC (using Matlab built-in `rand`), considering the numerical approximation of

$$\int_{[0,1]^2} \exp(-x^2 - y^2) dxdy = \pi \ \mathrm{erf}^2(1)/4 \approx 0.5577462853510335.$$

The QMC methods are

- Hammersley pointset,
- Halton sequence,
- Sobol sequence.

averaging the results obtained over 1000 tests.

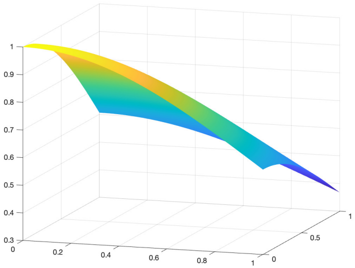

Figure: Plot of the function $f(x, y) = \exp\left(-x^2 - y^2\right)$ over the unit square $[0, 1] \times [0, 1]$.

```
* AVERAGE ABSOLUTE ERRORS

n:      2 ae: 1.57523e-01
n:      4 ae: 1.11644e-01
n:      8 ae: 8.26134e-02
n:     16 ae: 5.74396e-02
n:     32 ae: 4.02072e-02
n:     64 ae: 2.84918e-02
n:    128 ae: 2.07341e-02
n:    256 ae: 1.48894e-02
n:    512 ae: 1.00129e-02
n:   1024 ae: 7.33952e-03
n:   2048 ae: 4.93263e-03
n:   4096 ae: 3.37925e-03
n:   8192 ae: 2.47479e-03
n:  16384 ae: 1.84242e-03
n:  32768 ae: 1.21087e-03
n:  65536 ae: 8.77368e-04
n: 131072 ae: 6.23130e-04
n: 262144 ae: 4.51147e-04
```

```
* AVERAGE RATIOS

n1:      2 n2:      4 rat: 1.41094e+00
n1:      4 n2:      8 rat: 1.35141e+00
n1:      8 n2:     16 rat: 1.43827e+00
n1:     16 n2:     32 rat: 1.42859e+00
n1:     32 n2:     64 rat: 1.41119e+00
n1:     64 n2:    128 rat: 1.37415e+00
n1:    128 n2:    256 rat: 1.39254e+00
n1:    256 n2:    512 rat: 1.48702e+00
n1:    512 n2:   1024 rat: 1.36425e+00
n1:   1024 n2:   2048 rat: 1.48795e+00
n1:   2048 n2:   4096 rat: 1.45968e+00
n1:   4096 n2:   8192 rat: 1.36547e+00
n1:   8192 n2:  16384 rat: 1.34323e+00
n1:  16384 n2:  32768 rat: 1.52157e+00
n1:  32768 n2:  65536 rat: 1.38012e+00
n1:  65536 n2: 131072 rat: 1.40800e+00
n1: 131072 n2: 262144 rat: 1.38121e+00
```

Figure: Numerical integration of $\int_{[0,1]^2} \exp\left(-x^2 - y^2\right) dxdy$ by means of MC.

```
      * AVERAGE ABSOLUTE ERRORS
                                                      * AVERAGE RATIOS

n:       2 ae: 2.45519e-01
n:       4 ae: 1.39225e-01     n1:      2 n2:      4 rat: 1.76347e+00
n:       8 ae: 7.72688e-02     n1:      4 n2:      8 rat: 1.80182e+00
n:      16 ae: 4.22198e-02     n1:      8 n2:     16 rat: 1.83015e+00
n:      32 ae: 2.28156e-02     n1:     16 n2:     32 rat: 1.85048e+00
n:      64 ae: 1.22313e-02     n1:     32 n2:     64 rat: 1.86534e+00
n:     128 ae: 6.51809e-03     n1:     64 n2:    128 rat: 1.87651e+00
n:     256 ae: 3.45747e-03     n1:    128 n2:    256 rat: 1.88522e+00
n:     512 ae: 1.82716e-03     n1:    256 n2:    512 rat: 1.89227e+00
n:    1024 ae: 9.62572e-04     n1:    512 n2:   1024 rat: 1.89820e+00
n:    2048 ae: 5.05726e-04     n1:   1024 n2:   2048 rat: 1.90334e+00
n:    4096 ae: 2.65069e-04     n1:   2048 n2:   4096 rat: 1.90790e+00
n:    8192 ae: 1.38634e-04     n1:   4096 n2:   8192 rat: 1.91200e+00
n:   16384 ae: 7.23662e-05     n1:   8192 n2:  16384 rat: 1.91573e+00
n:   32768 ae: 3.77075e-05     n1:  16384 n2:  32768 rat: 1.91915e+00
n:   65536 ae: 1.96159e-05     n1:  32768 n2:  65536 rat: 1.92229e+00
n:  131072 ae: 1.01890e-05     n1:  65536 n2: 131072 rat: 1.92520e+00
n:  262144 ae: 5.28503e-06     n1: 131072 n2: 262144 rat: 1.92790e+00
```

Figure: Numerical integration of $\int_{[0,1]^2} \exp\left(-x^2 - y^2\right) dxdy$ by means of QMC based on Hammersley pointsets.

∗ AVERAGE ABSOLUTE ERRORS

n	ae
2	2.90704e-01
4	1.57760e-01
8	9.32628e-02
16	5.13634e-02
32	2.83120e-02
64	1.24329e-02
128	8.25970e-03
256	3.91058e-03
512	2.07291e-03
1024	1.21613e-03
2048	6.27785e-04
4096	3.05041e-04
8192	1.72274e-04
16384	1.00214e-04
32768	4.29910e-05
65536	1.99367e-05
131072	1.27472e-05
262144	6.07106e-06

∗ AVERAGE RATIOS

n1	n2	rat
2	4	1.84270e+00
4	8	1.69157e+00
8	16	1.81574e+00
16	32	1.81419e+00
32	64	2.27719e+00
64	128	1.50525e+00
128	256	2.11214e+00
256	512	1.88651e+00
512	1024	1.70452e+00
1024	2048	1.93717e+00
2048	4096	2.05803e+00
4096	8192	1.77067e+00
8192	16384	1.71907e+00
16384	32768	2.33104e+00
32768	65536	2.15637e+00
65536	131072	1.56400e+00
131072	262144	2.09967e+00

Figure: Numerical integration of $\int_{[0,1]^2} \exp\left(-x^2 - y^2\right) dxdy$ by means of QMC based on Halton sequence.

* AVERAGE ABSOLUTE ERRORS

n:	2	ae:	2.45519e-01
n:	4	ae:	1.11517e-01
n:	8	ae:	6.48094e-02
n:	16	ae:	3.01795e-02
n:	32	ae:	1.47587e-02
n:	64	ae:	7.70698e-03
n:	128	ae:	4.10285e-03
n:	256	ae:	1.84811e-03
n:	512	ae:	9.26900e-04
n:	1024	ae:	4.61313e-04
n:	2048	ae:	2.30300e-04
n:	4096	ae:	1.16661e-04
n:	8192	ae:	5.90287e-05
n:	16384	ae:	3.03164e-05
n:	32768	ae:	1.59370e-05
n:	65536	ae:	7.20366e-06
n:	131072	ae:	3.60196e-06
n:	262144	ae:	1.80111e-06

* AVERAGE RATIOS

n1:	2	n2:	4	rat:	2.20163e+00
n1:	4	n2:	8	rat:	1.72069e+00
n1:	8	n2:	16	rat:	2.14746e+00
n1:	16	n2:	32	rat:	2.04486e+00
n1:	32	n2:	64	rat:	1.91498e+00
n1:	64	n2:	128	rat:	1.87844e+00
n1:	128	n2:	256	rat:	2.22002e+00
n1:	256	n2:	512	rat:	1.99386e+00
n1:	512	n2:	1024	rat:	2.00927e+00
n1:	1024	n2:	2048	rat:	2.00309e+00
n1:	2048	n2:	4096	rat:	1.97410e+00
n1:	4096	n2:	8192	rat:	1.97635e+00
n1:	8192	n2:	16384	rat:	1.94708e+00
n1:	16384	n2:	32768	rat:	1.90227e+00
n1:	32768	n2:	65536	rat:	2.21234e+00
n1:	65536	n2:	131072	rat:	1.99993e+00
n1:	131072	n2:	262144	rat:	1.99986e+00

Figure: Numerical integration of $\int_{[0,1]^2} \exp\left(-x^2 - y^2\right) dxdy$ by means of QMC based on Sobol sequence.

As second numerical comparison in 2D, we considered the less regular case of the integrand

$$f(x,y) = \sqrt{x^2 + y^2}$$

over the unit square $[0,1] \times [0,1]$.

The reference value is $I \approx 0.765195713941172$.

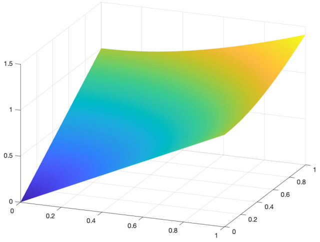

Figure: Plot of the function $f(x,y) = \sqrt{x^2 + y^2}$ over the unit square $[0,1] \times [0,1]$.

```
* AVERAGE ABSOLUTE ERRORS

n:        2 ae: 1.57523e-01
n:        4 ae: 1.11644e-01
n:        8 ae: 8.26134e-02
n:       16 ae: 5.74396e-02
n:       32 ae: 4.02072e-02
n:       64 ae: 2.84918e-02
n:      128 ae: 2.07341e-02
n:      256 ae: 1.48894e-02
n:      512 ae: 1.00129e-02
n:     1024 ae: 7.33952e-03
n:     2048 ae: 4.93263e-03
n:     4096 ae: 3.37925e-03
n:     8192 ae: 2.47479e-03
n:    16384 ae: 1.84242e-03
n:    32768 ae: 1.21087e-03
n:    65536 ae: 8.77368e-04
n:   131072 ae: 6.23130e-04
n:   262144 ae: 4.51147e-04
```

```
* AVERAGE RATIOS

n1:        2 n2:        4 rat: 1.41094e+00
n1:        4 n2:        8 rat: 1.35141e+00
n1:        8 n2:       16 rat: 1.43827e+00
n1:       16 n2:       32 rat: 1.42859e+00
n1:       32 n2:       64 rat: 1.41119e+00
n1:       64 n2:      128 rat: 1.37415e+00
n1:      128 n2:      256 rat: 1.39254e+00
n1:      256 n2:      512 rat: 1.48702e+00
n1:      512 n2:     1024 rat: 1.36425e+00
n1:     1024 n2:     2048 rat: 1.48795e+00
n1:     2048 n2:     4096 rat: 1.45968e+00
n1:     4096 n2:     8192 rat: 1.36547e+00
n1:     8192 n2:    16384 rat: 1.34323e+00
n1:    16384 n2:    32768 rat: 1.52157e+00
n1:    32768 n2:    65536 rat: 1.38012e+00
n1:    65536 n2:   131072 rat: 1.40800e+00
n1:   131072 n2:   262144 rat: 1.38121e+00
```

Figure: Numerical integration of $\int_{[0,1]^2} \sqrt{x^2 + y^2}dxdy$ by means of MC.

```
* AVERAGE ABSOLUTE ERRORS

n:       2 ae: 4.11642e-01
n:       4 ae: 2.20522e-01
n:       8 ae: 1.17358e-01
n:      16 ae: 6.22339e-02
n:      32 ae: 3.29614e-02
n:      64 ae: 1.74428e-02
n:     128 ae: 9.22270e-03
n:     256 ae: 4.87041e-03
n:     512 ae: 2.56823e-03
n:    1024 ae: 1.35195e-03
n:    2048 ae: 7.10398e-04
n:    4096 ae: 3.72594e-04
n:    8192 ae: 1.95063e-04
n:   16384 ae: 1.01938e-04
n:   32768 ae: 5.31807e-05
n:   65536 ae: 2.76987e-05
n:  131072 ae: 1.44040e-05
n:  262144 ae: 7.47908e-06
```

```
* AVERAGE RATIOS

n1:       2 n2:       4 rat: 1.86667e+00
n1:       4 n2:       8 rat: 1.87905e+00
n1:       8 n2:      16 rat: 1.88576e+00
n1:      16 n2:      32 rat: 1.88808e+00
n1:      32 n2:      64 rat: 1.88969e+00
n1:      64 n2:     128 rat: 1.89129e+00
n1:     128 n2:     256 rat: 1.89362e+00
n1:     256 n2:     512 rat: 1.89640e+00
n1:     512 n2:    1024 rat: 1.89966e+00
n1:    1024 n2:    2048 rat: 1.90308e+00
n1:    2048 n2:    4096 rat: 1.90663e+00
n1:    4096 n2:    8192 rat: 1.91012e+00
n1:    8192 n2:   16384 rat: 1.91354e+00
n1:   16384 n2:   32768 rat: 1.91683e+00
n1:   32768 n2:   65536 rat: 1.91997e+00
n1:   65536 n2:  131072 rat: 1.92299e+00
n1:  131072 n2:  262144 rat: 1.92590e+00
```

Figure: Numerical integration of $\int_{[0,1]^2} \sqrt{x^2 + y^2} dxdy$ by means of QMC based on Hammersley pointsets.

* AVERAGE ABSOLUTE ERRORS

n:	2	ae:	4.64733e-01
n:	4	ae:	2.47418e-01
n:	8	ae:	1.39827e-01
n:	16	ae:	7.81307e-02
n:	32	ae:	4.13382e-02
n:	64	ae:	1.84955e-02
n:	128	ae:	1.19832e-02
n:	256	ae:	5.69330e-03
n:	512	ae:	2.97188e-03
n:	1024	ae:	1.76224e-03
n:	2048	ae:	8.85194e-04
n:	4096	ae:	4.41341e-04
n:	8192	ae:	2.47689e-04
n:	16384	ae:	1.41676e-04
n:	32768	ae:	6.08364e-05
n:	65536	ae:	2.89188e-05
n:	131072	ae:	1.78771e-05
n:	262144	ae:	8.60704e-06

* AVERAGE RATIOS

n1:	2	n2:	4	rat:	1.87833e+00
n1:	4	n2:	8	rat:	1.76945e+00
n1:	8	n2:	16	rat:	1.78966e+00
n1:	16	n2:	32	rat:	1.89003e+00
n1:	32	n2:	64	rat:	2.23504e+00
n1:	64	n2:	128	rat:	1.54346e+00
n1:	128	n2:	256	rat:	2.10478e+00
n1:	256	n2:	512	rat:	1.91572e+00
n1:	512	n2:	1024	rat:	1.68642e+00
n1:	1024	n2:	2048	rat:	1.99080e+00
n1:	2048	n2:	4096	rat:	2.00569e+00
n1:	4096	n2:	8192	rat:	1.78183e+00
n1:	8192	n2:	16384	rat:	1.74828e+00
n1:	16384	n2:	32768	rat:	2.32880e+00
n1:	32768	n2:	65536	rat:	2.10369e+00
n1:	65536	n2:	131072	rat:	1.61765e+00
n1:	131072	n2:	262144	rat:	2.07703e+00

Figure: Numerical integration of $\int_{[0,1]^2} \sqrt{x^2 + y^2} dxdy$ by means of QMC based on Halton sequence.

```
* AVERAGE ABSOLUTE ERRORS

n:        2 ae: 4.11642e-01
n:        4 ae: 1.93134e-01
n:        8 ae: 9.88498e-02
n:       16 ae: 4.69025e-02
n:       32 ae: 2.26889e-02
n:       64 ae: 1.11784e-02
n:      128 ae: 5.83017e-03
n:      256 ae: 2.63034e-03
n:      512 ae: 1.35131e-03
n:     1024 ae: 6.54509e-04
n:     2048 ae: 3.24360e-04
n:     4096 ae: 1.62290e-04
n:     8192 ae: 8.17810e-05
n:    16384 ae: 4.19005e-05
n:    32768 ae: 2.20528e-05
n:    65536 ae: 9.90548e-06
n:   131072 ae: 4.95977e-06
n:   262144 ae: 2.48466e-06
```

```
* AVERAGE RATIOS

n1:       2 n2:       4 rat: 2.13138e+00
n1:       4 n2:       8 rat: 1.95382e+00
n1:       8 n2:      16 rat: 2.10756e+00
n1:      16 n2:      32 rat: 2.06720e+00
n1:      32 n2:      64 rat: 2.02971e+00
n1:      64 n2:     128 rat: 1.91734e+00
n1:     128 n2:     256 rat: 2.21651e+00
n1:     256 n2:     512 rat: 1.94651e+00
n1:     512 n2:    1024 rat: 2.06461e+00
n1:    1024 n2:    2048 rat: 2.01785e+00
n1:    2048 n2:    4096 rat: 1.99864e+00
n1:    4096 n2:    8192 rat: 1.98445e+00
n1:    8192 n2:   16384 rat: 1.95179e+00
n1:   16384 n2:   32768 rat: 1.90001e+00
n1:   32768 n2:   65536 rat: 2.22632e+00
n1:   65536 n2:  131072 rat: 1.99717e+00
n1:  131072 n2:  262144 rat: 1.99615e+00
```

Figure: Numerical integration of $\int_{[0,1]^2} \sqrt{x^2 + y^2}dxdy$ by means of QMC based on Sobol sequence.

Finally, we made our test in the case of the integrand

$$f(x,y) = \sqrt{(x-0.5)^2 + (y-0.5)^2}$$

over the unit square $[0,1] \times [0,1]$.

One immediately observes the singularity in $[0.5, \ 0.5]$. The environment `Chebfun 2` is not able to detect an approximation of the integral with many digits (absolute error of $\approx 6.1 \cdot 10^{-8}$), proposing

$$I \approx 3.825979145203977e-01,$$

while Matlab built-in `integral2` gives for absolute tolerance of 10^{-15}

$$I \approx 3.825978534775538e-01.$$

As we shall see, the absolute errors reach orders of the magnitude of 10^{-7}, 10^{-9}, but are much less predictable for QMC.

To this purpose,having installed the `Chebfun` environment, we have:

```
>> f=@(x,y) ((x-0.5).^2+(y-0.5).^2).^(1/2);
>> F=chebfun2(f,[0 1 0 1]);
Warning: Unresolved with maximum CHEBFUN length: 65537.
> In chebfun2/constructor (line 201)
In chebfun2 (line 82)
In compute_integrals (line 5)
>> I=sum2(F);
>> Q = integral2(f,0,1,0,1,'AbsTol',10^(-15));
>> format long e
>> I
I =
     3.825979145203977e-01
>> Q
Q =
     3.825978534775538e-01
>> I-Q
ans =
     6.104284394625736e-08
>>
```

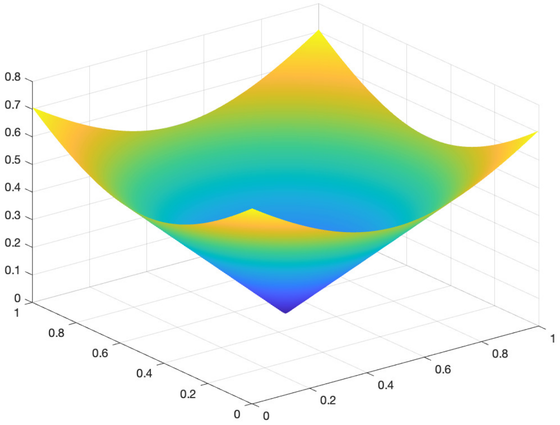

Figure: Plot of the function $f(x,y) = \sqrt{(x-0.5)^2 + (y-0.5)^2}$ over the unit square $[0,1] \times [0,1]$.

```
* AVERAGE ABSOLUTE ERRORS

n:        2 ae: 8.39864e-02
n:        4 ae: 5.71326e-02
n:        8 ae: 4.07585e-02
n:       16 ae: 2.85958e-02
n:       32 ae: 2.03982e-02
n:       64 ae: 1.40227e-02
n:      128 ae: 1.00634e-02
n:      256 ae: 7.25290e-03
n:      512 ae: 5.15978e-03
n:     1024 ae: 3.57244e-03
n:     2048 ae: 2.51180e-03
n:     4096 ae: 1.79318e-03
n:     8192 ae: 1.25828e-03
n:    16384 ae: 8.77745e-04
n:    32768 ae: 6.26451e-04
n:    65536 ae: 4.50126e-04
n:   131072 ae: 3.25956e-04
n:   262144 ae: 2.24430e-04
```

```
* AVERAGE RATIOS

n1:        2 n2:        4 rat: 1.47003e+00
n1:        4 n2:        8 rat: 1.40173e+00
n1:        8 n2:       16 rat: 1.42533e+00
n1:       16 n2:       32 rat: 1.40188e+00
n1:       32 n2:       64 rat: 1.45466e+00
n1:       64 n2:      128 rat: 1.39343e+00
n1:      128 n2:      256 rat: 1.38750e+00
n1:      256 n2:      512 rat: 1.40566e+00
n1:      512 n2:     1024 rat: 1.44433e+00
n1:     1024 n2:     2048 rat: 1.42226e+00
n1:     2048 n2:     4096 rat: 1.40076e+00
n1:     4096 n2:     8192 rat: 1.42510e+00
n1:     8192 n2:    16384 rat: 1.43354e+00
n1:    16384 n2:    32768 rat: 1.40114e+00
n1:    32768 n2:    65536 rat: 1.39172e+00
n1:    65536 n2:   131072 rat: 1.38094e+00
n1:   131072 n2:   262144 rat: 1.45237e+00
```

Figure: Numerical integration of $\int_{[0,1]^2} \sqrt{(x-0.5)^2+(y-0.5)^2}dxdy$ by means of MC.

```
* AVERAGE ABSOLUTE ERRORS

n:       2 ae: 2.90445e-02
n:       4 ae: 7.56719e-03
n:       8 ae: 2.00014e-03
n:      16 ae: 8.68052e-04
n:      32 ae: 9.51130e-04
n:      64 ae: 6.31857e-04
n:     128 ae: 3.11338e-04
n:     256 ae: 1.39404e-04
n:     512 ae: 5.69767e-05
n:    1024 ae: 2.25079e-05
n:    2048 ae: 8.52721e-06
n:    4096 ae: 3.18763e-06
n:    8192 ae: 1.16274e-06
n:   16384 ae: 4.20695e-07
n:   32768 ae: 1.48076e-07
n:   65536 ae: 5.01968e-08
n:  131072 ae: 1.48041e-08
n:  262144 ae: 2.22958e-09
```

```
* AVERAGE RATIOS

n1:       2 n2:       4 rat: 3.83821e+00
n1:       4 n2:       8 rat: 3.78333e+00
n1:       8 n2:      16 rat: 2.30417e+00
n1:      16 n2:      32 rat: 9.12653e-01
n1:      32 n2:      64 rat: 1.50529e+00
n1:      64 n2:     128 rat: 2.02949e+00
n1:     128 n2:     256 rat: 2.23335e+00
n1:     256 n2:     512 rat: 2.44668e+00
n1:     512 n2:    1024 rat: 2.53141e+00
n1:    1024 n2:    2048 rat: 2.63954e+00
n1:    2048 n2:    4096 rat: 2.67509e+00
n1:    4096 n2:    8192 rat: 2.74147e+00
n1:    8192 n2:   16384 rat: 2.76386e+00
n1:   16384 n2:   32768 rat: 2.84108e+00
n1:   32768 n2:   65536 rat: 2.94991e+00
n1:   65536 n2:  131072 rat: 3.39073e+00
n1:  131072 n2:  262144 rat: 6.63986e+00
```

Figure: Numerical integration of $\int_{[0,1]^2} \sqrt{(x-0.5)^2+(y-0.5)^2}dxdy$ by means of QMC based on Hammersley pointsets.

```
* AVERAGE ABSOLUTE ERRORS

n:        2 ae: 5.42889e-02
n:        4 ae: 2.65398e-02
n:        8 ae: 7.10405e-03
n:       16 ae: 9.50287e-04
n:       32 ae: 4.04023e-03
n:       64 ae: 3.48929e-03
n:      128 ae: 4.89643e-05
n:      256 ae: 3.43755e-04
n:      512 ae: 2.98806e-04
n:     1024 ae: 1.51171e-04
n:     2048 ae: 2.50484e-06
n:     4096 ae: 2.73379e-05
n:     8192 ae: 5.99947e-06
n:    16384 ae: 6.19846e-06
n:    32768 ae: 6.90779e-06
n:    65536 ae: 2.41491e-08
n:   131072 ae: 2.55021e-07
n:   262144 ae: 6.17678e-07
```

```
* AVERAGE RATIOS

n1:        2 n2:        4 rat: 2.04557e+00
n1:        4 n2:        8 rat: 3.73586e+00
n1:        8 n2:       16 rat: 7.47569e+00
n1:       16 n2:       32 rat: 2.35206e-01
n1:       32 n2:       64 rat: 1.15789e+00
n1:       64 n2:      128 rat: 7.12621e+01
n1:      128 n2:      256 rat: 1.42439e-01
n1:      256 n2:      512 rat: 1.15043e+00
n1:      512 n2:     1024 rat: 1.97661e+00
n1:     1024 n2:     2048 rat: 6.03515e+01
n1:     2048 n2:     4096 rat: 9.16250e-02
n1:     4096 n2:     8192 rat: 4.55672e+00
n1:     8192 n2:    16384 rat: 9.67896e-01
n1:    16384 n2:    32768 rat: 8.97315e-01
n1:    32768 n2:    65536 rat: 2.86048e+02
n1:    65536 n2:   131072 rat: 9.46943e-02
n1:   131072 n2:   262144 rat: 4.12871e-01
```

Figure: Numerical integration of $\int_{[0,1]^2} \sqrt{(x-0.5)^2+(y-0.5)^2}dxdy$ by means of QMC based on Halton sequence.

```
* AVERAGE ABSOLUTE ERRORS

n:        2 ae: 2.90445e-02
n:        4 ae: 2.90445e-02
n:        8 ae: 1.86116e-02
n:       16 ae: 2.11078e-02
n:       32 ae: 2.33555e-03
n:       64 ae: 2.99863e-04
n:      128 ae: 3.35931e-04
n:      256 ae: 3.48470e-04
n:      512 ae: 3.29063e-04
n:     1024 ae: 6.63485e-05
n:     2048 ae: 1.15129e-05
n:     4096 ae: 4.03502e-06
n:     8192 ae: 5.52585e-07
n:    16384 ae: 8.78295e-08
n:    32768 ae: 8.87589e-08
n:    65536 ae: 8.89632e-08
n:   131072 ae: 7.55938e-08
n:   262144 ae: 5.98070e-08
```

```
* AVERAGE RATIOS

n1:       2 n2:       4 rat: 1.00000e+00
n1:       4 n2:       8 rat: 1.56055e+00
n1:       8 n2:      16 rat: 8.81743e-01
n1:      16 n2:      32 rat: 9.03759e+00
n1:      32 n2:      64 rat: 7.78873e+00
n1:      64 n2:     128 rat: 8.92632e-01
n1:     128 n2:     256 rat: 9.64018e-01
n1:     256 n2:     512 rat: 1.05898e+00
n1:     512 n2:    1024 rat: 4.95961e+00
n1:    1024 n2:    2048 rat: 5.76295e+00
n1:    2048 n2:    4096 rat: 2.85325e+00
n1:    4096 n2:    8192 rat: 7.30207e+00
n1:    8192 n2:   16384 rat: 6.29157e+00
n1:   16384 n2:   32768 rat: 9.89530e-01
n1:   32768 n2:   65536 rat: 9.97703e-01
n1:   65536 n2:  131072 rat: 1.17686e+00
n1:  131072 n2:  262144 rat: 1.26396e+00
```

Figure: Numerical integration of $\int_{[0,1]^2} \sqrt{(x-0.5)^2+(y-0.5)^2}dxdy$ by means of QMC based on Sobol sequence.

- R.E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, Acta Numerica, 1998, pp. 1–48.
- J. Dick, F.Y. Kuo, I.H. Sloan, J. High dimensional integration – the Quasi-Monte Carlo way, Acta Numerica, 2014, pp. 1–157.
- P.J. Davis and P. Rabinowitz, Methods of Numerical Integration, Dover 1984.
- I.M. Sobol, A Primer For The Monte Carlo Method, CRC press 1994.