Low Discrepancy Sequences

- I wish to approximate an integral $\int_0^1 f(u)du$

using a total of n evaluations of the function *f*

Suppose that the function is smooth and I use points $u_i=(2i-1)/(2n), i=1,2,3,...,n$.

Note: $\int_0^{1/n} f(u)du \approx n^{-1} f(u_1)$

For a point in the interval u_1

How good is this approximation? Using a Taylor series expansion around the point $f(u_1)$,

$$f(x) \approx f(u_1) + f'(u_1)(x - u_1)$$

and integrating both sides over the interval $[0, 1/n]$ we obtain

$$\int_0^{1/n} f(u)du \approx \frac{1}{n} f(u_1) + \frac{f'(u_1)}{4n^2}.$$

If the function f has a bounded first derivative this means that the integral over each subinterval

$$\int_{(j-1)/n}^{j/n} f(u)du \approx \frac{1}{n} f(u_j)$$

with an error that is less than

$$\text{constant} \times \frac{1}{n^2}$$

Therefore the error in the sum over n such intervals is less than a constant $\times 1/n$. How does this compare to a crude Monte Carlo integral?

We have seen that if we randomly select n uniform$[0,1]$ points u_i and use the crude estimator

$$\frac{1}{n} \sum_{i=1}^{n} f(u_i)$$

then the estimator has variance

$$\frac{\sigma^2}{n} \text{ where}$$
$$\sigma^2 = var(f(U_i))$$

and standard error

$$\frac{\sigma}{\sqrt{n}}.$$

For large values of n notice that

$$constant \times 1/n < \frac{\sigma}{\sqrt{n}}.$$

i.e. the error in the numerical integral is less than that in the monte Carlo integral. For large values of n, and for smooth functions f in one-dimension, numerical integration is better than Monte Carlo integration.

The story in 2 dimensions changes a little. Suppose we now want to find an integral of the form

$$\iint f(u_1,u_2)du_1du_2$$

where the integral is over the unit square. Again we wish to use n evaluations of the function. Suppose we use equally spaced points on a lattice in the two-dimensional unit square and suppose $n = m^2$. If we define

$$u_i = \frac{2i-1}{2m}, i = 1,2,\ldots m$$

then there are exactly $n = m^2$ points of the form (u_i,u_j) at which we can evaluate the function. The distance between adjacent points of this form is $1/m$ and so if we approximate the integral $\iint f(u_1,u_2)du_1du_2$ by an average

$$\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} f(u_i,u_j)$$

the error in this approximation is less than or equal to

$$\text{constant} \times \frac{1}{m} = \text{constant} \times \frac{1}{\sqrt{n}}.$$

Note that this is now the same order of magnitude as the standard error of a Monte Carlo integral. For dimensions higher than 2, for example for evaluating an integral like

$$\iiint f(u_1, u_2, u_3) du_1 du_2 du_3$$

the Monte Carlo integral as measured by the order of the error term than is a numerical integral based on placement of points on a lattice.

Equally spaced points on the line, or in space have the advantage that they fill holes efficiently (they get reasonably close to all points in the space). One disadvantage is that I need to know in advance how many points (n) are to be selected to that I can space them $1/n$ apart.

Is it possible to construct a sequence so that at least periodically the sequence consists of equally spaced points?

Low Discrepancy Sequences

low-discrepancy: successive numbers are added in a position as far as possible from the other numbers

i.e. avoiding clustering

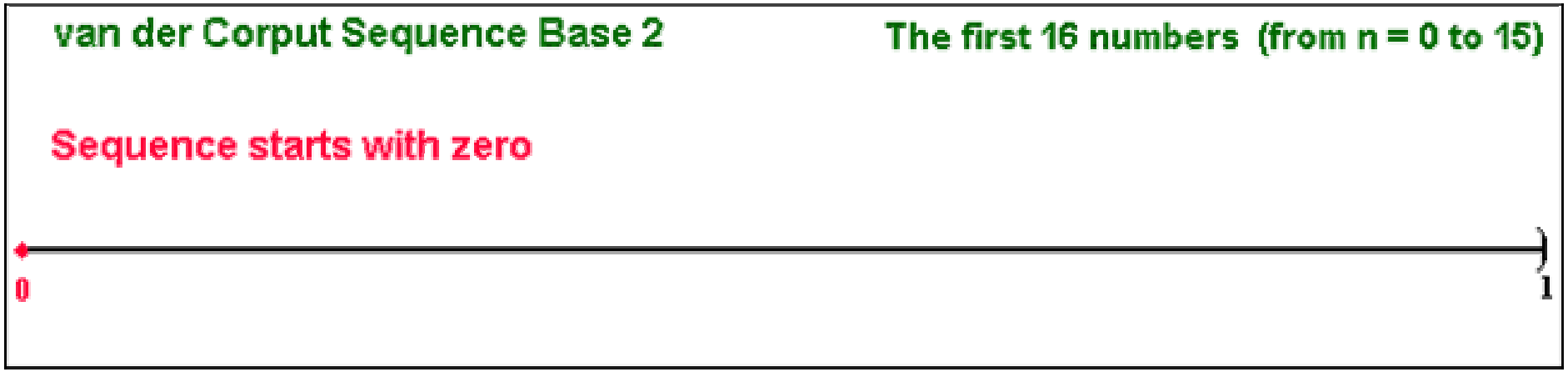
the numbers generated sequentially fill in the larger "gaps" between the previous numbers of the sequence.

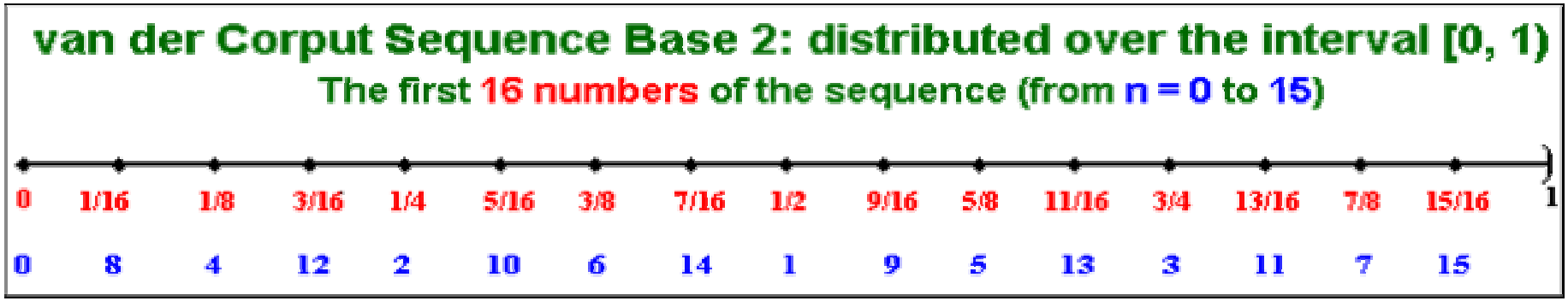
In dimension 1, the ***van der Corput*** **(1935) sequence in base 2**, starts from zero, and is confined in the interval [0, 1)

Van der corput sequence

See: http://www.puc-rio.br/marco.ind/main.html#contents for applets shown here.

with the first 16 Van der Corput numbers (n from 0 to 15) given by:





Van der Corput, base b

- For base 3, has 3 in denominator :
- **0, 1/3, 2/3, 1/9, 4/9, 7/9, 2/9, 5/9, 8/9, 1/27, 10/27, 19/27, ...**
- The n'th term of the van der Corput sequence, for base b, is generated as follows:
  - The decimal-base number n is expanded in the base b. For example, n = 4 in base 2 is **100** (4 = **1** x 4 + **0** x 2 + **0** x 1);
- The number in base b is reflected. In the example, 100 becomes **001**;
- Map into interval [0,1). 001 becomes 0.001 (binary decimal) corresponds to the decimal number 1/8, that is 1/8 (= 0 x (1/2) + 0 x (1/4) + 1 x (1/8)).

General Van der Corput

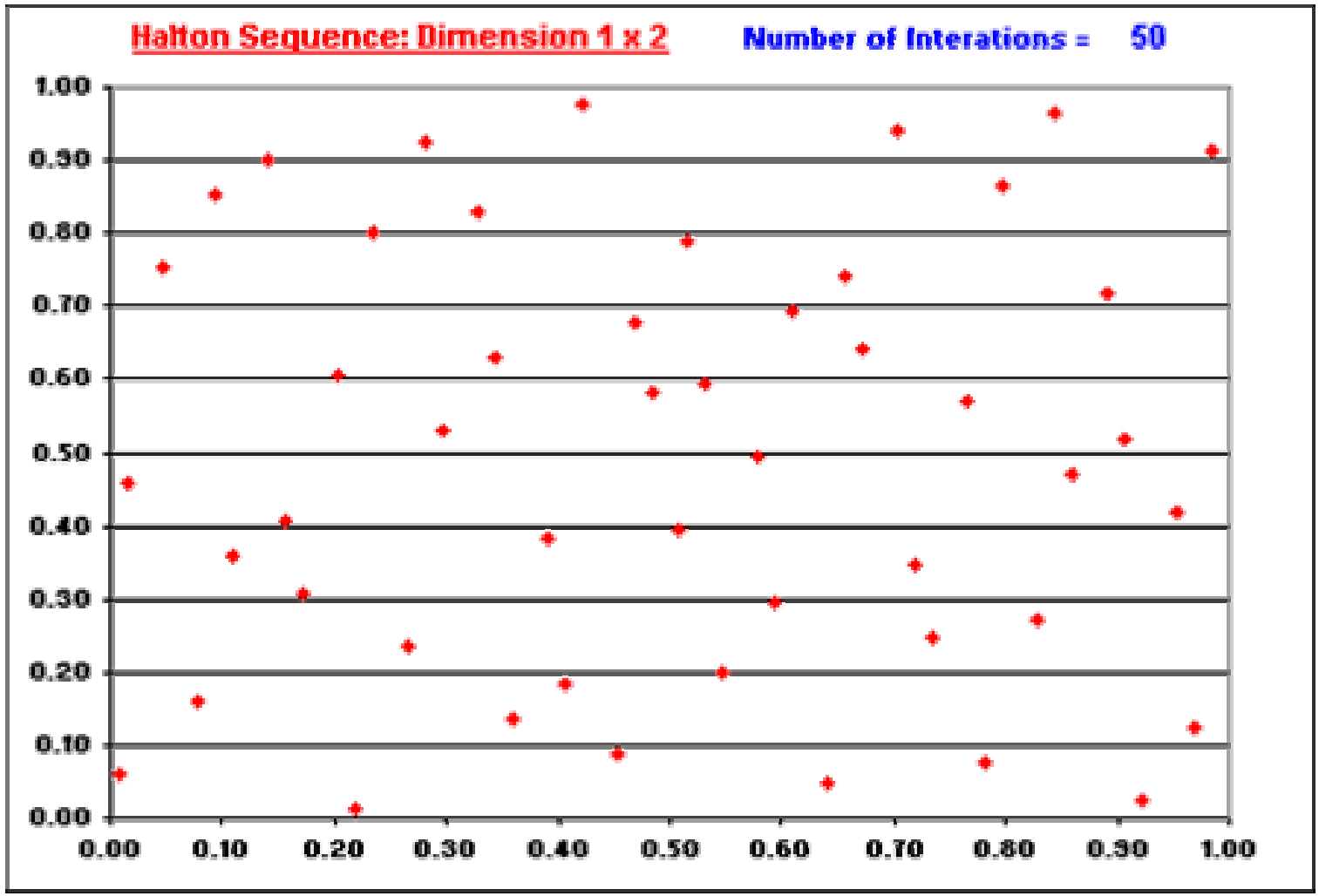
In general, for base b if $$n = \sum_{j=0}^{m} a_j(n)\ b^j$$

$$\text{van der Corput base b } (n) = \Phi_b(n) = \sum_{j=0}^{m} a_j(n)\ b^{-j-1}$$

Halton Sequence of dimension 3: each component is a different Van der Corput sequence (different prime base b)

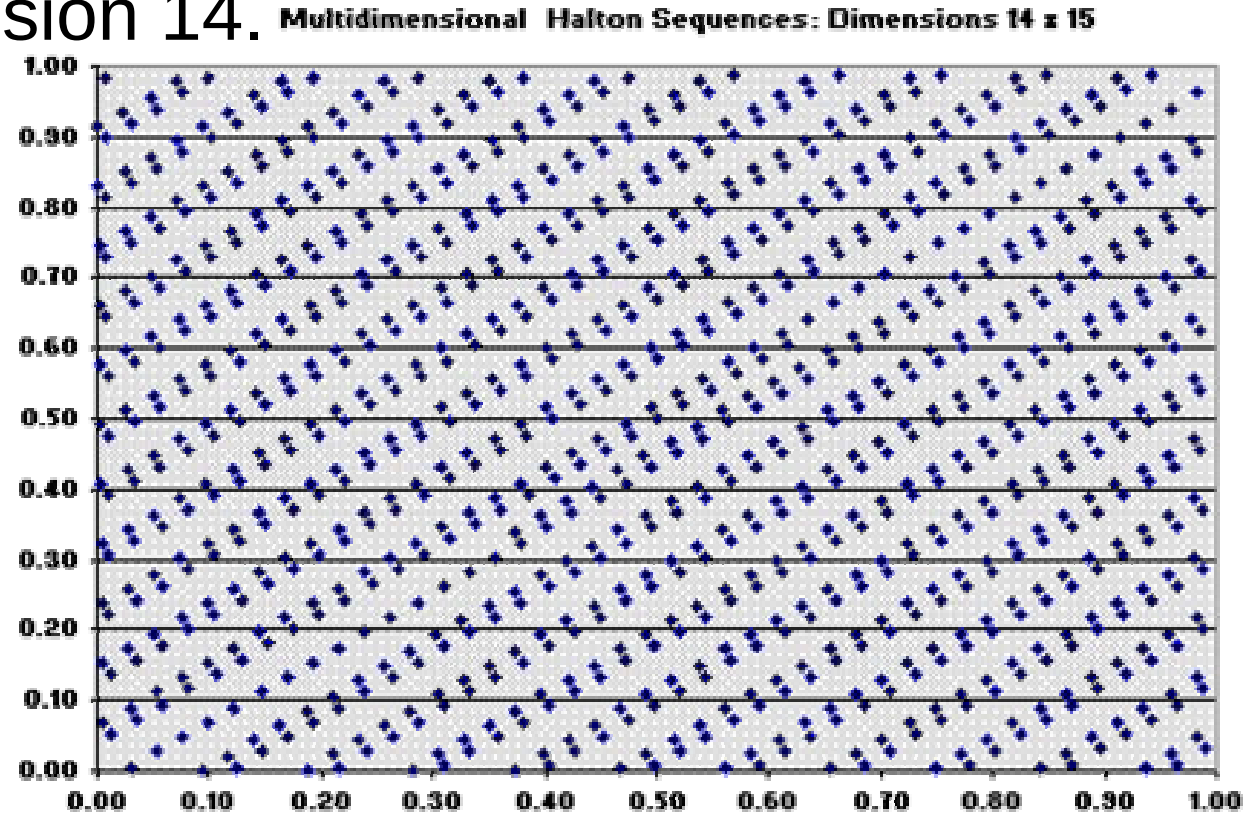
m	repres base 2	first comp	repres base 3	second comp	repres base 5	third comp
1	1	1/2	1	1/3	1	1/5
2	10	1/4	2	2/3	2	2/5
3	11	3/4	10	1/9	3	3/5
4	100	1/8	11	4/9	4	4/5
5	101	5/8	12	7/9	10	1/25
6	110	3/8	20	2/9	11	6/25
7	111	7/8	21	5/9	12	11/25
9	1000	1/16	22	8/9	13	16/25
10	1001	9/16	100	1/27	14	21/25

Halton Sequence



Halton sequences have problems in high dimensions

- Halton sequence exhibits structure in high dimensions, becomes unsatisfactory after ~ dimension 14.



Faure Sequences

- The Faure sequence is like the Halton sequence, but it uses only one base for all dimensions and it uses a permutation of the vector elements for each dimension.
- The base of a Faure sequence is the smallest prime that is larger than or equal to the number of dimensions (or 2 for one dimensional problem).
- If d=50, the last Halton sequence (in dimension 50) uses the 50th prime number that is 229, whereas the Faure sequence uses the first prime number after 50, that is a base 53, which is much smaller than 229, so, the "filling in the gaps" in high-dimensions is faster with Faure sequence than Halton.
- By reordering the sequence within each dimension, Faure sequences prevents some problems of correlation for sequential high-dimensions that occurred with the Halton sequence. Makes a link between the low discrepancy sequences theory and the combinatorial theory for the vector reordering.

Faure

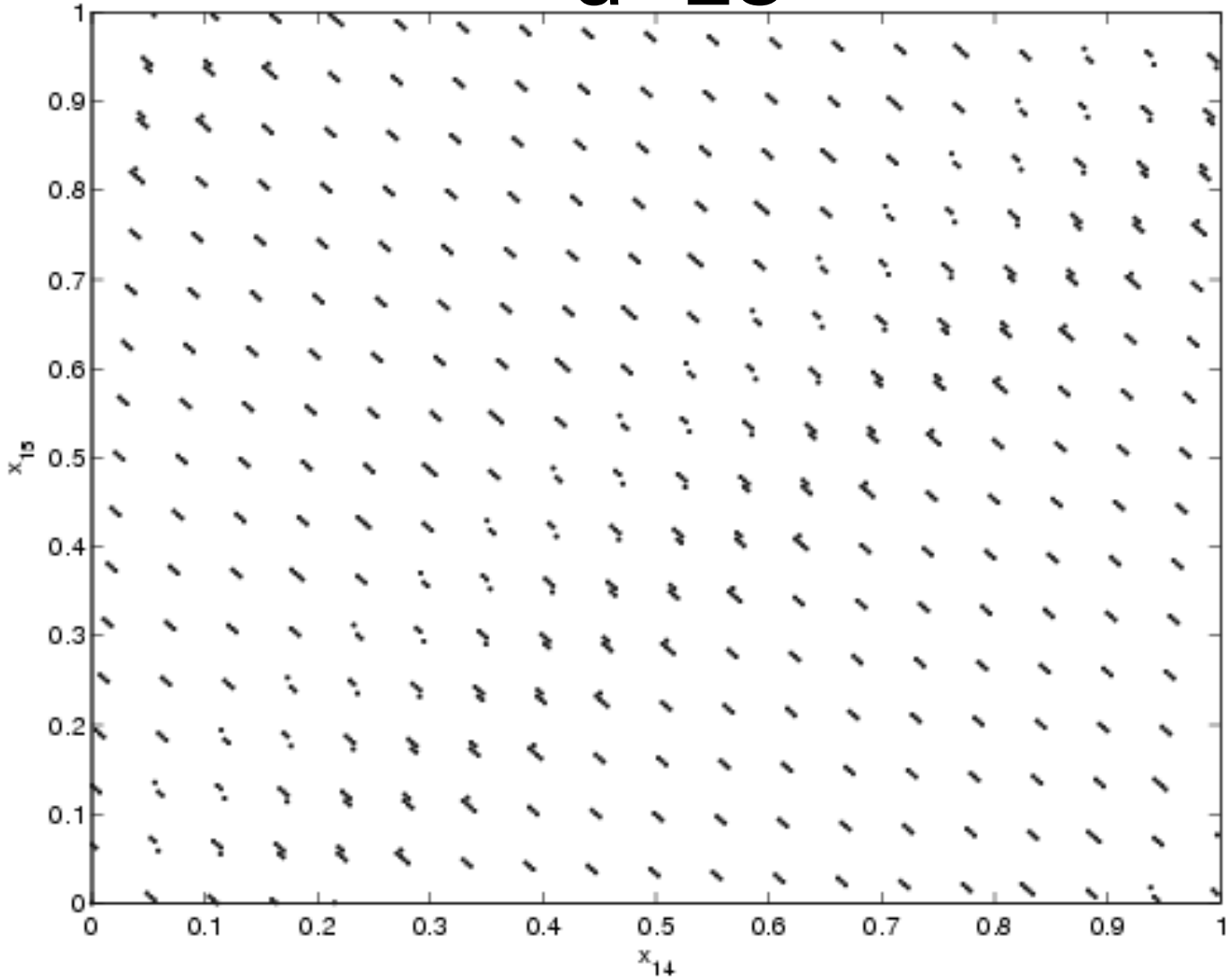
- The algorithm same equation as before but with a combinatorial rearrangment of the **aj**. This is performed using a *recursive* equation, from dimension (d -1) to the new dimension d:
- Start the first dimension using the van der Corput sequence with the specific Faure's base b, reorder the numbers with the equation below for d = 2, and so on:

$$a_i^d(n) = \sum_{j \geq i}^{m} \frac{j!}{i!\,(j-i)!}\, a_j^{d-1}(n) \mod b$$

Sobol Sequences

- The Sobol sequence has the same base for all dimensions and reorders of the vector elements within each dimension.
- Sobol sequence uses base 2 for all dimensions.
- Reordering more complex.
- Sobol reordering is based on a set of "direction numbers", $\{v_i\}$ given by $v_i=m_i/2^i$ "*where the m_i are odd positive integers less than 2^i, and v_j are chosen so that they satisfy a recurrence relation using the coefficients of a primitive polynomial in the Galois field G(2)*" (Gentle, J.E. (1998): "Random Number Generation and Monte Carlo Methods"
Springer- Verlag, New York, Inc., (1998) p.161).
- i.e. Sobol sequence use the coefficients of irreducible primitive polynomials of modulo 2 to reorder.
- C code for Sobol algorithm in Press, W.H. & S.A. Teukolsky & W.H. Vetterling & B.P. Flannery (2002): "Numerical Recipes in C++ – The Art of Scientific Computation"
Cambridge University Press, Second Edition, 2002, 1002 pp.

Last two coordinates of Faure, d=15



Squared error in option price, crude MC, Van der Corput and Sobol