

QUASI-MONTE CARLO IN FINANCE: EXTENDING FOR PROBLEMS OF HIGH EFFECTIVE DIMENSION

Marcos Eugênio da Silva[§]
Thierry Barbe[□]

RESUMO

Neste artigo mostramos que é possível usar métodos de simulação quase-Monte Carlo em problemas de alta dimensão efetiva. Isto é feito por meio de uma combinação de uma cuidadosa construção das seqüências de Sobol e de uma decomposição apropriada da matriz de covariância dos fatores de risco. A eficácia do método é ilustrada por meio da precificação de opções que envolve a solução de problemas com dimensão nominal da ordem de 550 (e dimensão efetiva da ordem de 300). Acreditamos que o método apresentado seja de fácil implementação e de grande interesse para os participantes do mercado financeiro.

Palavras-chave: quase-Monte Carlo, baixa discrepância, Sobol, dimensão efetiva, seqüências determinísticas.

ABSTRACT

In this paper we show that it is possible to extend the use of quasi-Monte Carlo for applications of high effective dimension. This is achieved through a combination of a careful construction of the Sobol sequence and an appropriately chosen decomposition of a covariance matrix. The effectiveness of this procedure is demonstrated as we price average options with nominal dimensions ranging up to 550 (effective dimension around 300). We believe the method we present is easy to implement and should be of great interest to practitioners.

Key words: quasi-Monte Carlo, low-discrepancy, Sobol, effective dimension, deterministic sequences.

JEL classification: G1, C4.

* The authors would like to thank Kevin Dowd and two anonymous referees for their helpful comments. The authors bear full responsibility for all remaining errors.

§ Professor of Economics at the University of São Paulo – USP. Phone-Fax: 55 11 30916067, e-mail: medsilva@usp.br.

□ Banco BNP Paribas. Phone: 55 11 38413429, e-mail: thierry.barbe@br.bnpparibas.com.

Recebido em maio de 2004. Aceito em maio de 2005.

1 INTRODUCTION

Quasi-Monte Carlo simulation has now reached a broader audience. Indeed, it is difficult to find a general presentation of the Monte Carlo method without mention of deterministic techniques. Examples are Hull (2000) and Jorion (2001). We feel however that this is due to the much greater attention quasi-Monte Carlo has received in academic circles than in the markets. Quasi-Monte Carlo is mentioned, but no road map is offered as far as implementation and situations where the technique might or not be efficient. This is obviously due to the problems faced by quasi-Monte Carlo when dealing with problems of high dimensions (50 could be an appropriate subjective threshold). The failure of the method in treating such cases has generated a feeling of unsafety that has so far mitigated market practitioner's response to it.

It is thus not surprising that much research has been devoted to solving this shortcoming. Various solutions have been put forward but none seems to have been adopted by a wide range of users. Although it could be argued that this is due to the state of infancy in which quasi-Monte Carlo methods lie, we believe it is due to a combination of two factors. One of these is the case by case approach, where it is shown that a variant of a deterministic method works for a particular financial instrument, reducing the scope of its applications. The other factor is the implementation issue. In many cases the complexity of the simulation makes it lose some of its attractiveness. Therefore, the aim of this paper is to introduce a technique, at the same time easy to implement and of general use, which is unaffected by the dimensionality of the problem. We hope to catch the attention of some practitioners with the applicability of this procedure and contribute to the popularization of deterministic methods.

The rest of this paper is organized as follows. The next section offers a review of the different approaches that have been adopted for the treatment of the problem of dimensionality. We outline the main trends and comment on the results obtained. We then proceed to introduce our simulation procedure. We show how to construct a modification of the Sobol sequence and how to apply it to the pricing of derivatives. In the third section the results of our experiments, the pricing of Asian options of dimension up to 550,¹ are presented. We will then be able to judge its effectiveness as we compare it to traditional Monte Carlo and to a purely deterministic method. We finally conclude with suggestions as to what could be done to refine this procedure.

2 SOLVING THE PROBLEM OF DIMENSIONALITY

The use of Quasi-Monte Carlo in Finance is relatively recent. The pioneering contributions of Paskov and Traub (1995), Galanti and Jung (1997), Joy, Boyle and Tan (1996) were all published in the latter half of the nineties. However, in other fields, notably in Physics, researchers have long been aware of the advantages (and disadvantages) of quasi-Monte Carlo. See for example Lécot (1991) and Morokoff and Caflisch (1991). Interestingly the authors conclude that the "*error reduction for quasi-Monte Carlo is limited as the spatial dimension increases.*" This is shown in Morokoff and Caflisch (1994, 1995).

However, the results obtained in finance seemed at first more appealing. Indeed, Paskov and Traub (1995) price a CMO with dimension 360 and find that quasi-random methods outperform standard Monte Carlo even with the use of variance reduction methods. The quasi-random se-

¹ The average that enters the payoff function is taken over 250 prices. The dimension in this case is the number of prices that compose the average. In other applications such as basket options or Value at Risk the dimension would be the number of underlying assets or risk factors, respectively. We could define the dimension informally as the number of random points necessary for each realization of the Monte Carlo experiment.

quence used was Sobol's, generated by FINDER, a proprietary software. But, in Galanti and Jung (1997) the quasi-random sequences cease to be effective for certain ranges of dimensionality and appear to converge again for other dimensions. Sometimes the Faure sequence is more efficient, sometimes it is the Sobol sequence. Joy, Boyle and Tan (1996) don't venture in these territories since the higher dimension tested is 52 for an Asian option. They used the Faure sequence and found it more efficient than standard Monte Carlo methods. A later research by Boyle *et al.* (1997), increases the dimensionality to 100. The sequences of Faure and Sobol are used. At that point Faure sequences ceased to be appropriate for the pricing of the Asian option. Sobol was still close to the theoretical result but it was clear that, as expected, the pricing error increased with the dimension.

These somewhat conflicting results may be explained by what has been coined the "effective dimension" of the problem as shown by Caflisch, Morokoff and Owen (1999). In this paper the authors try to explain how such surprising results could be obtained and conclude that the dimension that "matters" for the simulation in the case for example of the CMO used by Paskov and Traub (1995) is much lower than 360. They emphasize that "quasi-Monte Carlo methods provide significant improvements in accuracy and computational speed for problems of small to moderate dimension".

We feel that one of the main reasons that has kept market professionals reluctant to embrace deterministic methods, even for problems of small to moderate dimensions, is the confusion generated by the initial results. Some sequences fare better than others, depending on the problem at hand. Some sequences cease to be effective for some dimensions, and then converge again. Why is it that one can use the sequence for a mortgage-backed security of dimension 360 and not use it for an Asian option of dimension 100? Nonetheless, it seems now that what will be called "crude quasi-Monte Carlo" has been accepted as non-operational for high dimensional problems. Much of the research is now addressing this problem. We will now see what has been done in order to circumvent this problem. It is noteworthy that another interesting line of work resides, following the steps of Cranley and Patterson (1976), in an attempt to define confidence intervals for quasi-Monte Carlo simulations. The absence of these estimates may also be a hindrance for practitioners.

Researchers have tried to extend quasi-Monte Carlo in many ways. We can classify the approaches in two categories, those that try to produce new sequences better adapted to the problems at hand and those that try to modify the problems to render them compatible with quasi-Monte Carlo.

The first trend is best represented by Niederreiter and Xing who have produced a number of sequences which improve substantially on Sobol (for a comparison see for example Niederreiter and Xing, 1996). However these properties are asymptotic and it has been suggested by Jäckel (2002) that for practical purposes the results are inferior to those obtained with Sobol sequences. Another approach for the generation of sequences that could be used for quasi-Monte Carlo simulations is a scrambling of the deterministic sequence. In an attempt to derive the variance of quasi-Monte Carlo simulations Owen (1994a, 1994b) showed that his method of randomization of deterministic sequences, which maintains their main characteristics, improves accuracy over high dimensions. These ideas have been very well adapted, with some modifications for greater ease of use, by Tan and Boyle (2000) to the pricing of financial derivatives. Results are shown for an Asian option with dimensions 50, 250 and 365. The scrambled sequence outperforms crude quasi-Monte Carlo and standard Monte Carlo.

The second trend has focused on the modification of the problem being treated. Indeed, when dealing with the pricing of options for which sample paths of the underlying asset have to be gen-

erated, it is possible to "reduce" the dimensionality of the problem through a carefully chosen discretization. This is shown in a very nice exposition by Moskowitz and Caflisch (1996) where the authors use the Brownian bridge to evaluate high dimensional Feynman Kac path integrals. The results are very promising but the dimensions tested don't exceed 32.

Although this line of work is very useful we understand that the first approach is more promising for it seeks greater generality, through a procedure that can be applied to all problems. Indeed many problems cannot benefit from the treatment suggested. An example would be the calculation of VaR. Incidentally this has been tested by Papageorgiou and Paskov (1999) but with only 34 risk factors.

The method we present has two objectives: the first is to attain greater coverage of financial instruments for which quasi-Monte Carlo remains operational and the second is the ease of implementation, an important factor for use in non-academic environments. We show our construction below.

3 A NEW ALGORITHM FOR GENERATING SOBOL SEQUENCES

3.1 The standard algorithm

We first review the standard algorithm for the generation of uniform numbers in the unit hypercube. This will facilitate the understanding of the innovation proposed. Press *et al.* (1992) and Galanti and Jung (1997) show the algorithm in detail. We follow a sequential approach.

Direction Numbers

Initially we construct a vector of numbers, known as direction numbers, of length w that will serve as a base for the calculation of the Sobol numbers. We need a direction number for each digit, in base 2, of the numbers that will be used in the sequence. Usually, numbers up to $2^{32} = 4294967295$ are used, which means that $w = 32$ direction numbers for each dimension are necessary. The dimensions will be indexed by $k = 1, 2, \dots, D$. The construction of these direction numbers is quite complicated and will be sketched in what follows.

Given a series of integers a_1, a_2, \dots, a_{d-1} , that are zero or one, the primitive polynomial modulo 2 of degree d is defined as

$$P = x^d + a_1 x^{d-1} + a_2 x^{d-2} + \dots + a_{d-1} x + 1 \quad (1)$$

where the coefficients a_i are zero or one.

For each dimension k , an appropriate primitive polynomial is chosen and a series of decimal integers $m_{ik}, d_k < i < w$, is generated, starting from the following recursion with d_k terms, where d_k is the degree of the polynomial associated to the k^{th} dimension:

$$m_{ik} = 2a_{1,k} m_{i-1,k} \oplus 2^2 a_{2,k} m_{i-2,k} \oplus \dots \quad (2)$$

$$\oplus 2^{dk-1} a_{dk-1,k} m_{i-dk+1,k} \oplus (2^{dk} m_{i-dk,k} \oplus m_{i-dk,k}) \tag{3}$$

for $k = 1, 2, \dots, D$, where \oplus represents the bit to bit sum, XOR (exclusive OR), applied on the base 2 representation of the decimal integer m_{ik} . It is necessary, of course, to supply the d_k initial values of m_{ik} in each dimension. These can, in principle, be chosen amongst any of the odd integers inferior to 2^i , with $0 < i \leq d_k$.

The next step is to transform m_{ik} in binary fractions:

$$v_{ik} = \frac{m_{ik}}{2^i} \tag{4}$$

This is tantamount to transforming a decimal integer m_{ik} to its equivalent in base 2 and then shifting the position of the fractional point by i positions to the left. For instance, if $m_{ik} = 7$ and $i = 4$, then $7_{10} = 111_2$, and the v_{ik} would be:

$$v_{ik} = 0.0111 \tag{5}$$

Table 1 displays some examples of direction numbers.

Table 1 – Direction numbers

i	m_{ik}	m_{ik} in base 2	v_{ik}
1	1	1	0.1
2	3	11	0.11
3	5	101	0.101
4	15	1111	0.1111

Each dimension k of the Sobol sequence is created with the use of a different primitive polynomial. There are several tables of these polynomials available in the literature. Press *et al.* (1992) list 150 primitive polynomials, allowing the construction of Sobol sequences with the same dimension. Jäckel (2002) offers a CD containing more than 8 million polynomials! These should suffice for nearly all practical problems in Finance.

Construction of the initial term of the Sobol sequence for each dimension k

The necessary steps are:

1. choose an integer x randomly, for instance, $x = 2$. This number defines the starting point of the sequence (the "seed" of the process).
2. Compute the Gray Code of x , $G(x)$, as explained below (in the example, $G(2) = 3$).
3. Transform $G(x)$ to its binary representation ($G(2) = 11$).
4. Sum bit by bit (XOR) the direction numbers associated with the digits of $G(x)$ (in binary representation) that are different from zero. In the example, counting from right to left, the bits of

$G(x)$ different from zero are the first and second. Therefore XOR has to be done with the first (0.1) and the second (0.11) direction numbers taken from Table 1:

$$y(2, k) = 0.10 \oplus 0.11 = 0.01 \quad (6)$$

5. Transform the resulting number into a decimal number contained in (0,1), by multiplying each digit of that binary representation by 2^{-l} , where l is the position of the digit in the decimal part of number, counting from left to right. Add up the terms. The result is the first Sobol number in dimension k :

$$s(2, k) = 0 \times 2^{-1} + 1 \times 2^{-2} = 0.250 \quad (7)$$

Construction of the following numbers

The most efficient way to construct the following numbers is to use the algorithm suggested by Antonov-Saleev and described in Press *et al.* (1992). Instead of XORing several direction numbers, the $(j-1)^{th}$ Sobol can be obtained from the $(j-1)^{th}$ using just one direction number:

$$y(j, k)_2 = y(j-1, k)_2 \oplus v_{jk} \quad (8)$$

where v_{jk} is the direction number associated with the rightmost zero in the binary representation of $(j-1)$.

In the example, $2_{10} = 10_2$, therefore the rightmost zero is encountered in the first position and one should XOR the first direction number of the k^{th} dimension (taken from Table 1) with $y(2, k)$.

The second number in the sequence will be:

$$y(3, k) = 0.01 \oplus 0.10 = 0.11 \quad (9)$$

$$s(3, k) = 1 \times 2^{-1} + 1 \times 2^{-2} = 0.750 \quad (10)$$

Similarly, $3_{10} = 11_2$ and it is necessary to add a leading zero to put our hands on the rightmost zero in the binary representation of 3, that is, the digit in the third position. Therefore XOR the third direction number. The third number in the sequence is:

$$y(4, k) = 0.110 \oplus 0.101 = 0.011 \quad (11)$$

$$s(4, k) = 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 0.375 \quad (12)$$

The fourth number depends on $y(4, k)$ and on the first direction number, because the rightmost zero in the binary representation of 4 is in the first position: $4_{10} = 100_2$

$$y(5, k) = 0.011 \oplus 0.100 = 0.111 \quad (13)$$

$$s(5, k) = 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 0.875 \tag{14}$$

And so forth. It is worth noticing that the numbers “fill” the gaps in the interval (0,1) looking for empty spaces, as if the procedure knew where the positions of all prior numbers were.

Gray codes

The Gray Code of an integer j is defined as

$$G(j) = j \oplus \text{int} \left[\frac{j}{2} \right] \tag{15}$$

where $\text{int}[x]$ represents the largest integer inferior or equal to x . Table 2 displays some examples of Gray codes. Notice that $G(j-1)$ and $G(j)$ differ, in their binary representations, only in the digit relative to the rightmost zero in the binary representation of $j-1$. Therefore, in the construction of the j th Sobol number, the induction is the same as XORing all the direction numbers associated to the unit bits of $G(j)$.

Table 2 – Gray codes

j in decimal	j in binary	$\left[\frac{j}{2} \right]$ in binary	$G(j)$ in binary	$G(j)$ in binary
0	000	000	000	0
1	001	000	001	1
2	010	001	011	3
3	011	001	010	2
4	100	010	110	6
5	101	010	111	7
6	110	011	101	5
7	111	011	100	4

3.2 The proposed method

It seemed that quasi-random sequences, Sobol’s construction in particular, would revolutionize the use of Monte Carlo simulation in Finance thanks to the economy of time allowed by the method. But, as mentioned before, for problems of high dimension it is necessary to simultaneously use many primitive polynomials and the results become unsatisfactory. Jäckel (2002) explains the reason for this drawback (although Sobol himself already knew the problem). The high degree of dependence among dimensions arises because of the initial values freely chosen for the construction of the m_{ik} in each dimension k . Theoretically, the only restriction on these numbers is that m_{ik} is both odd and inferior to 2^i . Usually a deterministic rule is used for choosing them. For instance, $m_{ik} = 1$, $0 < i \leq d_k$ or m_{ik} is the largest integer less than 2^i , with $0 < i \leq d_k$. These deterministic rules, although valid from a theoretical standpoint, generate surprisingly bad results empirically.

Sobol (1976) presents some mathematical conditions that should be met in order to avoid such distortions. The equations that appear are complicated and Paskov and Traub (1995) have already solved them for problems involving up to 360 dimensions. Jäckel (2002) suggests an alternative method that should work well whatever the dimension of the problem. It amounts to a very simple randomization of the choice process of the initial values m_{ik} . Although not as precise as the method proposed by Sobol, Jäckel sustains that his produces very good results. Unfortunately, the results we achieved using Jäckel's method weren't satisfactory and suffered from the same drawback as the traditional deterministic procedures. We briefly present Jäckel's method and suggest an alternative procedure that performed very well empirically in our quest to extend quasi-Monte Carlo to higher dimensions.

Jäckel's random method of generation of the initial direction numbers

For each dimension k , using a pseudo-random generator, generate d_k uniform numbers u_{ik} such that

$$m_{ik} = \text{int}[u_{ik} \times 2^{i-1}] \text{ for } 0 < i \leq d_k \quad (16)$$

is odd (simply keep drawing u_{ik} until the condition is met)

Although extremely simple, Jäckel reports excellent results for dimensions as high as 90 or 100. Unfortunately, we didn't succeed in reproducing such results. On the contrary, the dimensions can be quite correlated, especially when they are very high. A test done with 2500 dimensions showed that 2449 pairs of consecutive dimensions have correlation greater than 70% (in absolute value)! The process of random generation of the initial direction numbers should have been more efficient for high dimensions; since more initial direction numbers were necessary, it should have been easier to break the regularities.

Alternative random method of generation of the initial direction numbers

Nonetheless, Jäckel's suggestion was quite ingenious and simple and led us to continue experimenting in the same direction. After several attempts, we have come up with a procedure that is as simple as the one proposed by Jäckel, and so far, has worked extraordinarily well for a fair amount of financial applications. The procedure is as follows:

1. For each dimension k , using a pseudo-random generator, draw d_k uniform numbers u_{ik} such that

$$m_{ik} = \max(\text{int}[u_{ik} \times 2^i - 1], 1) \text{ for } 0 < i \leq d_k \quad (17)$$

is odd (simply keep drawing u_{ik} until the condition is met)

2. Generate randomly a seed for each dimension. It is preferable to choose large numbers as seeds; this will force the algorithm to use direction numbers of higher orders in the w -bit long array for each dimension (this enhances the power of randomization of the algorithm)

The method was used by Silva (2002) to generate 2500-dimensional Sobol sequences with 2047 points each.² The uniform numbers in consecutive dimensions covered the unit square very well for nearly all the 2500 dimensions! Table 4, at the end, shows one example of the first 10 direc-

² A text file with the initial direction numbers for dimensions up to 5 thousand and a C routine that generates the Sobol numbers can be downloaded from the author's website at the Universidade de São Paulo, <http://www.econ.fea.usp.br/medsilva>.

tion numbers for the first 50 dimensions using the method proposed here. The analysis of unit square plots of the sequences shows that even in high dimensions the correlation between consecutive dimensions was low and the generated points covered the unit square in a very uniform way. Table 3 shows that there were few consecutive dimensions with correlation (in absolute value) greater than 20% and even in these cases the “filling” of the unit square was more uniform than observed with Jäckel’s method. This result can be improved with a shuffling of the generated points. It has been possible to decrease the absolute value of all correlations between consecutive and non-consecutive dimensions to less than 20 percent.

Table 3 – Correlations above 20% in absolute value for the proposed method with $D = 2500$ and $N = 2047$

$dim(k)$	$dim(k+1)$	correlation
63	64	-0.2428
634	635	0.2737
636	637	0.2412
1219	1220	0.5396
1309	1310	-0.433
1468	1469	0.3177
2341	2342	0.2105

Table 4 – First 10 direction numbers up to dimension 50 (highlighted initial direction numbers are random, the others follow the recursion for m_{ik})

Dimension k	Degree of Polynomial	Repres. Number	i									
			1	2	3	4	5	6	7	8	9	10
1	3	1	1	1	1	13	13	61	81	145	145	157
2	3	2	1	1	1	11	31	55	61	157	181	191
3	4	1	1	1	1	25	25	25	217	321	321	321
4	4	4	1	1	1	13	11	7	31	227	381	653
5	5	2	1	1	3	13	27	57	73	187	101	307
6	5	4	1	1	1	5	1	53	37	245	49	1013
7	5	7	1	1	1	1	5	61	45	237	301	425
8	5	11	1	1	3	7	13	51	79	229	101	47
9	5	13	1	1	5	11	23	51	75	79	425	937
10	5	14	1	1	5	13	25	15	51	55	375	147
11	6	1	1	1	5	11	23	49	97	225	37	43
12	6	13	1	1	1	1	1	3	109	101	461	669
13	6	16	1	1	1	3	9	17	99	135	335	605
14	6	19	1	1	3	1	3	7	194	143	461	971
15	6	22	1	1	1	9	29	59	111	231	391	399
16	6	25	1	1	5	5	11	25	127	123	239	23

The number of dimensions that can be tackled with this method depends only on the amount of available primitive polynomials, currently around 8 million. Thus it seems that the curse of dimensionality has been broken in practice. The mathematical solution of the uniformity conditions proposed by Sobol for problems with dimension superior to 370 is not available yet.³ The next step is to check if these numbers allow us to get better results when they are used in practical problems faced by financial analysts. This is carried out in the next section.

4 RESULTS

We have applied our procedure to the pricing of Asian options. We work initially with an option that depends on the geometric mean of the underlying asset for which there is a closed formula. This enables us to compare our results with a bulletproof benchmark. Then, we use an arithmetic mean option for which the theoretical price is obtained averaging a large number of Monte Carlo simulations. In turn these will provide us with the confidence intervals that will allow a comparison between Monte Carlo and the methods here outlined. Finally, in order to test our procedure with correlated assets, we show the results obtained when pricing a Asian basket option.

4.1 Geometric mean Asian option

We present first the results obtained with the pricing of an option whose price depends on the geometric average of 251 prices, including the spot price. The dimension of this problem is therefore 250. Parameters of interest are: spot price is 100.00, strike price is 100.00, interest rate is 0.20% which must be converted to a continuously compounded rate, volatility is 0.30% and prices that compose the average are registered daily and therefore $t_i - t_{i-1} = 1/252$. In this case an analytic solution for the price of the option is available. Following Kemna and Vorst (1990), we obtain 10.2731. This provides us with a benchmark against which our simulations can be compared. As shown by Jäckel (2002), the full power of Sobol sequences can be unleashed only when combined with an appropriate decomposition of the covariance matrix of a brownian path. Four pricing methods are used: crude Monte Carlo with the Cholesky decomposition, crude Monte Carlo with Schur decomposition, modified Sobol with Cholesky decomposition and modified Sobol with Schur decomposition.⁴ The crude Monte Carlo simulations were conducted with the RAN2 generator taken from Press *et al.* (1992). It is widely accepted in the literature. We varied the number of sample paths (N) generated by our simulations. These range from 1,000 to 50,000 with increments of 1,000. We don't show the results obtained with crude quasi-Monte Carlo because errors are too big for this dimension.

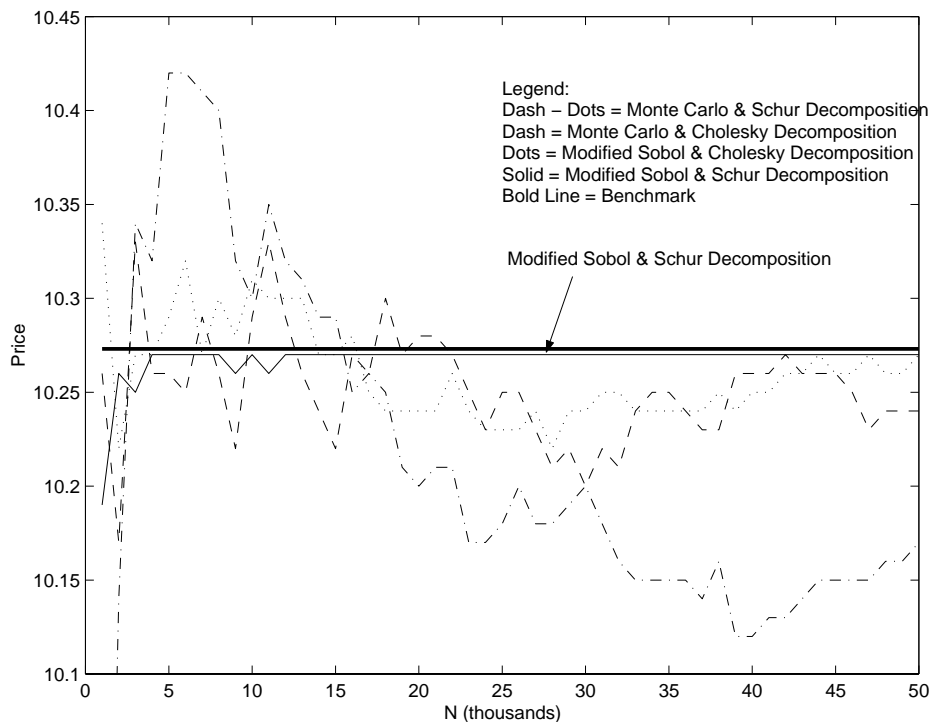
We have chosen two figures to help visualize the magnitude of the errors and the convergence patterns of the methods applied. In Figure 1 we can see the price obtained with each method. Focusing first on the quasi-Monte Carlo method we can see that the "optimized" Sobol sequence with the Cholesky decomposition does not diverge substantially from the theoretical price in a manner similar to a crude quasi-Monte Carlo. In fact it fares quite well when compared to a realization of a crude Monte Carlo simulation. This is already very encouraging. It seems indeed that the dimension has ceased to bear an effect on the results. However, in order to fully exploit the se-

3 Kucherenko and Sobol offer a software that generates uniform sequences up to dimension 370. See <http://www.broda.co.uk>

4 The decompositions generate a matrix A which is used to transform a random variable $z \sim N(0,1)$ into a brownian motion $w = Az$. Matrix A is such that $C = AA'$, where C is the covariance matrix of the brownian path. When using the Cholesky decomposition A' is an upper triangular matrix. In the case of Schur decomposition, $A = QT^{-\frac{1}{2}}$, where Q is a unitary matrix such that $C = QTQ' = AA'$, and T is a diagonal matrix of the eigenvalues of C. For more details see Golub and Van Loan (1983).

quence we apply the Schur decomposition. The results are extremely encouraging. We can see that for a number as modest as 5,000 sample paths we are already very close to the theoretical price and convergence is extremely stable. In fact for $N > 12000$, we obtain an almost straight line, indistinguishable from the bold line representing the benchmark. As far as crude Monte Carlo is concerned, as expected, the Schur decomposition does not improve on the Cholesky decomposition. This is not surprising since Monte Carlo benefits from a dimension increase.⁵

Figure 1 – Price of the geometric average option



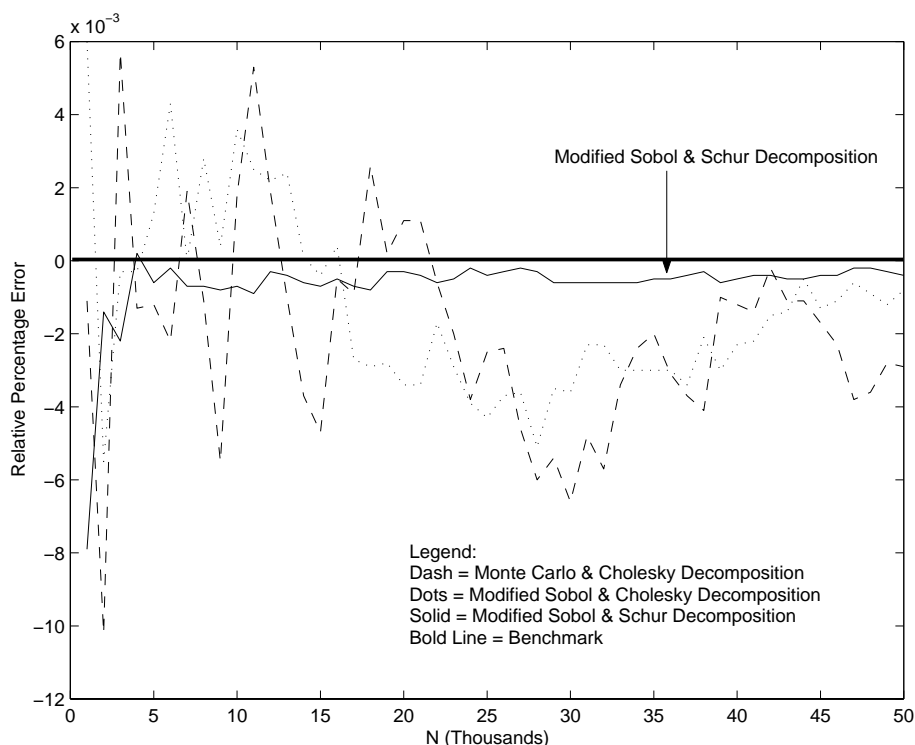
Our observations are confirmed by Figure 2 where we show the relative percentage error. This is simply,

$$error = 100 \times \frac{p_{mc} - \bar{p}}{\bar{p}}$$

where p_{mc} is the price obtained by simulation and \bar{p} is the theoretical price. Here we have chosen not to include crude Monte Carlo with the Schur decomposition for sake of clarity. Again we see that both quasi-Monte Carlo simulations are operational. Errors obtained using the Cholesky decomposition are in the proximity of (generally inferior to) the crude Monte Carlo analysis. Quasi-Monte Carlo with the Schur decomposition offers surprising results. Errors are very small and convergence is extremely stable. This is not the case with either of the other methods which display sharp increases and decreases of the pricing errors. Of course overall the variance seems to decrease as N increases.

⁵ The order of error is of $(ND)^{\frac{1}{2}}$, where N is the number of points used and D is the dimension.

Figure 2 – Error of the geometric average option



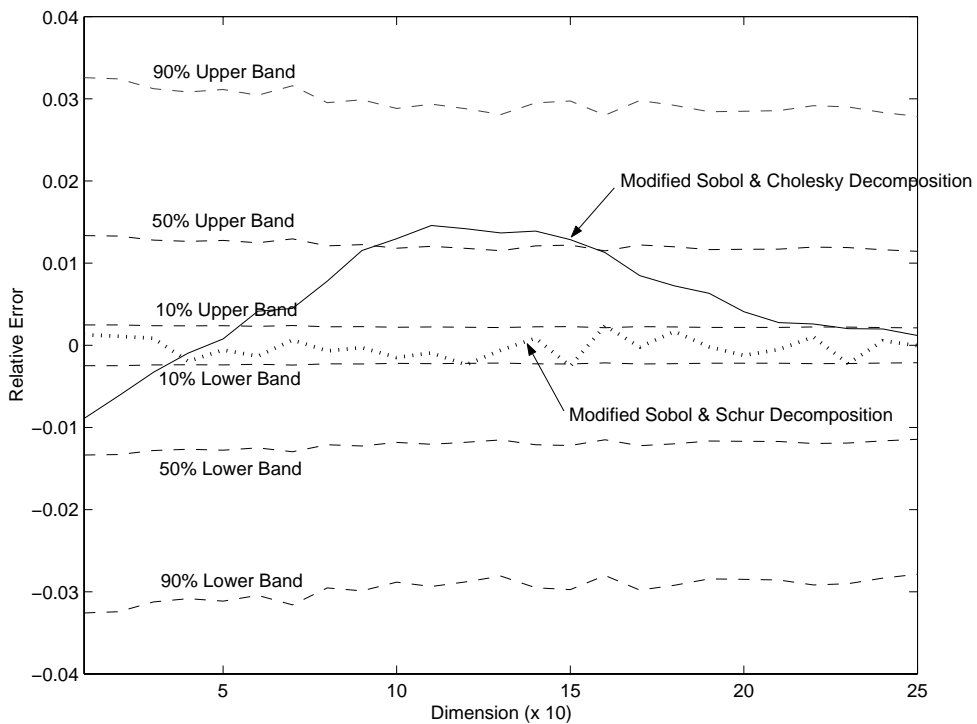
It thus seems that dimension does not affect quasi-Monte Carlo anymore. However this is not sufficient. In order for the method to be effective it has to prove itself superior to standard Monte Carlo. It is not possible to compare the results obtained with quasi-Monte Carlo with only one realisation of crude Monte Carlo. The latter being probabilistic, there is no guarantee that this particular simulation is not a particularly “bad” one. We need the distribution of the Monte Carlo simulations. We now look at the pricing of an arithmetic average Asian option.

4.2 Arithmetic mean Asian option

For this problem we maintain the parameters used for the geometric average case. All sample paths are generated once again with the four methods outlined above. Crude quasi-Monte Carlo is again left out of the visual presentations because the errors are too big to fit the graphs. In fact we start to experience difficulties with crude quasi-Monte Carlo with problems of dimensionality greater than 60. The dimension of the problem is 250. This time no theoretical price is available. Nonetheless, we are confident that an average of 1,000 Monte Carlo simulations of 5,000 sample paths each should get quite close to the theoretical price. In fact this is equivalent to a 5 million sample paths Monte Carlo simulation. The price obtained is 10.9150. The same pattern observed above for quasi-Monte Carlo with the Cholesky decomposition is obtained here. It is close to the benchmark but exhibits a more erratic convergence although it fares remarkably well against realizations of crude Monte Carlo with RAN2. The dimension does not have near the impact it has on crude quasi-Monte Carlo.

Since our objective is to compare the efficiency of the method proposed and crude Monte Carlo we limit ourselves to showing the results obtained for the error of the price obtained with quasi-Monte Carlo and the Schur decomposition along with various confidence intervals inferred from the simulations used to derive the benchmark. We obtain the confidence intervals by applying the central limit theorem to our crude Monte Carlo results, which are averages of realizations of the payoff function. The sample mean and variance allow us to determine confidence intervals around the theoretical price, which can then be translated into confidence intervals for the relative errors. This can be seen in Figure 3. We plot the result obtained with quasi-Monte Carlo and Schur decomposition along with the confidence intervals for crude Monte Carlo simulation. We chose arbitrarily the 10, 50 and 90 percent confidence intervals. We see that for simulations of 3,000 points quasi-Monte Carlo is already within the 10 percent confidence interval and stays comfortably within this range. In fact with simulations of 4,000 points we enter the 5 percent confidence interval which is also never exceeded henceforth. In other words more than 95 percent of the prices obtained via Monte Carlo simulation would constitute a worse approximation than that obtained with the proposed method. We thus see that not only has the dimension been tamed but a very substantial improvement over crude Monte Carlo methods has also been achieved.

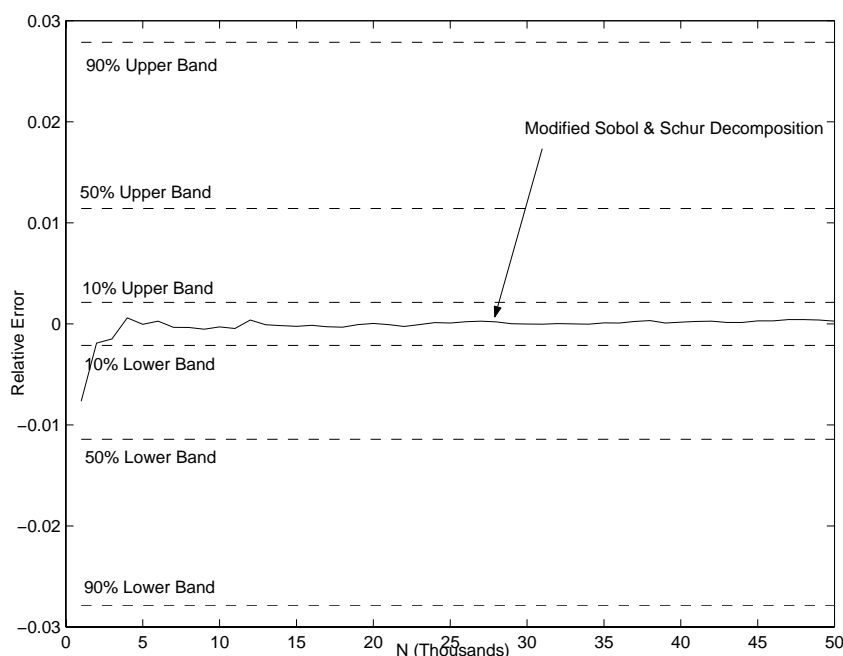
Figure 3 – Varying the dimensions



It can be inferred from, for example Galanti and Jung (1997), that quasi-random sequences may exhibit cyclic behavior. For example, it might be an efficient method for dimensions 1 to 60, then diverge significantly from the theoretical result between dimensions 80 to 200 and at last converge again. Bearing this in mind there is no guarantee that we have not stumbled upon some kind of “optimal” dimension and that another story would have to be told for problems of lesser dimension.

Thus in order to study the behaviour of our sequence we decided to price arithmetic average Asian options with dimensions varying from 10 to 250 with increments of 10. Here we show the results obtained for quasi-Monte Carlo with both the Cholesky and Schur decomposition. Simulations are conducted for $N=5000$. In the case of the geometric average option we saw that quasi-Monte Carlo with Cholesky flirted with crude Monte Carlo. We hope to infer adequately the quality of this quasi-Monte Carlo experiment. In order to do this we included the confidence intervals for 10, 50 and 90 percent for crude Monte Carlo. This is presented in Figure 4. It can be seen that quasi-Monte Carlo with the Schur decomposition is quite stable. It is always within the 10 percent confidence interval. It thus seems that it is a serious alternative for crude Monte Carlo independent of the dimension of the problem. When it comes to quasi-Monte Carlo with Cholesky, results are slightly less clear cut. As hinted from previous works there are ranges of dimension for which the sequences performs less accurately. For the problem under consideration this is indeed the case for dimensions 90 to 160. For these options the price obtained with quasi-Monte Carlo and Cholesky decomposition fluctuates around the 50 percent confidence interval upper band. However, in this case the method is by no means discredited. For the other dimensions crude Monte Carlo is definitely inferior. Also, the dimensionality is clearly no longer a problem. Again, quasi-Monte Carlo with the Schur decomposition improves immensely on crude Monte Carlo.

Figure 4 – Monte Carlo confidence intervals



4.3 Asian basket option

In many applications that involve Monte Carlo simulation it is necessary to generate sample paths for a family of assets whose returns are dependent on one another. An example would be the calculation of Value-at-Risk. A comprehensive and up to date analysis of the several techniques for VaR computation, including Monte Carlo and quasi-Monte Carlo simulations can be found in

Dowd (2002). It is thus interesting to test the proposed procedure for applications of this sort for which one of the dimension components is given by the number of assets.

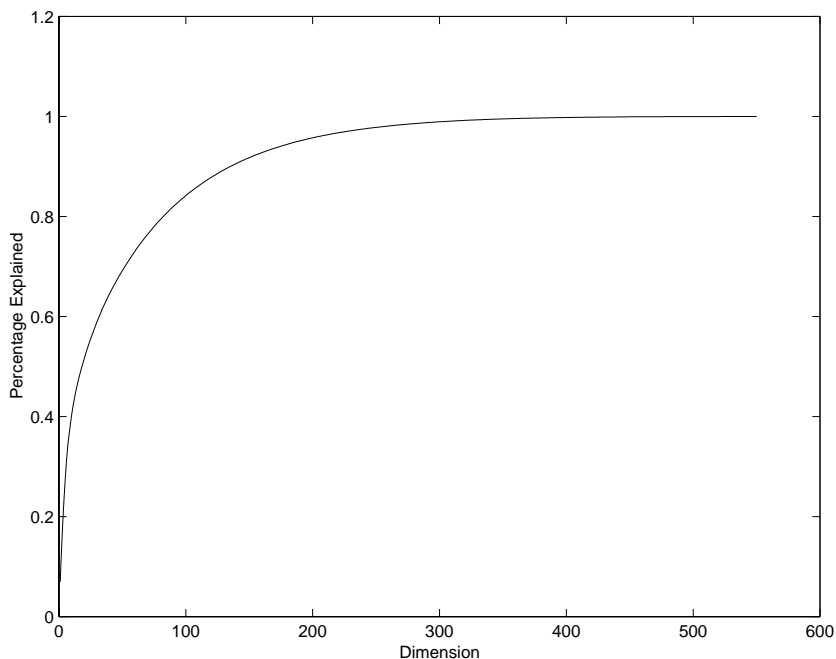
The instrument that is now priced is an Asian basket call option. The number of assets that compose the basket is 55. The call payoff depends on the arithmetic average of 10 prices registered along each of the asset’s sample path. We are therefore in the presence of a problem of dimension $550 = 55 \times 10$. However, its effective dimension is 305, “effective dimension” in the sense that the analysis of the empirical covariance matrix of the multidimensional Brownian motion shows that the first 305 dimensions account for 99% of the total variance. The cumulative percentage explained can be seen on Figure 5. This is obtained by applying the Schur decomposition on the covariance matrix. This number is well beyond the figures that have so far been tested in the financial literature. Since this instrument is rather peculiar we show how it is priced. The first step⁶ consists in generating a matrix of Brownian innovations B of size 55×10 . The payoff of the call is defined as,

$$Call_T = \left(\frac{\sum_{i=1}^{55} q_i \bar{S}_i}{\sum_{i=1}^{55} q_i S_{i0}} - X, 0 \right)^+$$

where,

$$\bar{S}_i = \frac{\sum_{t=1}^{10} S_{it}}{10}$$

Figure 5 – Cumulative explained variance



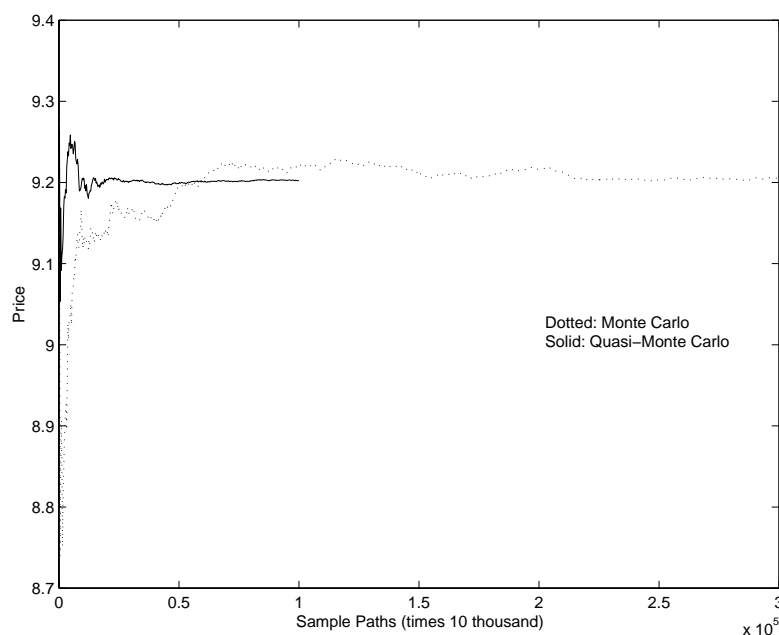
6 The multidimensional Brownian motion construction follows the methodology described by Jäckel (2002).

From the description above $n = 55$ and $N = 10$ in our example. The weights of each asset are given by q_i and X is the strike price; q_i is such that, at time $t = 0$, $q_i \cdot S_{i0}$ is the same for all assets.

The test was implemented with Matlab. We price the Asian basket options with a number of sample paths (this number is obviously the same for each asset) varying from 250,000 to 300,000 in intervals of 250. The pseudo-random number generation is the one offered with the standard version of the software. We assume the price of 9.20% obtained by a Monte Carlo simulation of 300,000 sample paths to be extremely close to the theoretical price. Hardware limitations have not allowed us to conduct quasi-Monte Carlo simulations for N greater than 100,000. We conjecture that, given the results obtained in the previous experiments, that those obtained with the proposed construction of the Sobol sequences should not diverge from the path obtained with simulations of 100 thousand sample paths.

The results of the test can be seen on Figure 6. The dotted line corresponds to the prices obtained via Monte Carlo simulation. The solid line represents the quasi-Monte Carlo simulation. We can see a much faster convergence for the deterministic method. Indeed for N as low as 20,000 the results are already satisfying, as the magnitude of the pricing error is a tenth of a cent. On the other hand traditional Monte Carlo simulation oscillates substantially until N reaches 200,000. At that stage the results obtained are quite stable and very close to the benchmark. We can roughly speak of a tenfold economy! However the most appealing result is the robustness of the proposed construction with regards to the dimension. Indeed it seems that the problems related in literature on quasi-Monte Carlo are no longer an issue. We have dealt here exclusively with financial applications, however tests have been carried out with integrals of much higher effective dimension and the results have been consistent with those presented in this paper.⁷

Figure 6 – Asian Basket call option



⁷ Joe and Kuo (2003) show an example of integration using quasi-Monte Carlo simulation that produces encouraging results. The tested integral has a high effective dimension and the Sobol sequence used performed better than Faure sequences and shifted lattice rules. We have conducted some experiments - which are not shown here for sake of brevity - with the same integral and the results obtained are also quite favourable to our Sobol construction when compared to traditional Monte Carlo and to the method used by Joe and Kuo.

5 CONCLUSION

The aim of this paper was to introduce a method of generation of quasi-random points that performed well independently of the effective dimension of application considered. It is intended as a general enough solution to the problems faced by deterministic sequences as dimensions increase so that it could be applied to all situations in finance where one would require simulation. It was shown that, independently of the decomposition chosen, the simulations performed with these points fared extremely well, in some cases overwhelmingly outperforming crude Monte Carlo. We have conducted other simulations with barrier options and look-back options which have yielded the same results. We plan to test the procedure with applications of a different kind, for example value at risk calculations, where there might be some interference from the covariance matrix. At the same time we hoped to have interested a few practitioners with the ease of implementation of the proposed procedure. It is indeed extremely simple to set up and the extra accuracy should encourage finance professionals to embrace quasi-random simulation with more enthusiasm. Of course, total understanding of the properties of the procedure, such as its asymptotic properties, should enable us to delve deeper into its potential. However this is outside our jurisdiction and we hope it will be of interest to other researchers.

REFERENCES

- Boyle, P. P.; Broadie, M.; Glasserman, P. Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 1997.
- Caflich, R.; Morokoff, W.; Owen, A. Valuation of mortgage-backed securities using brownian bridges to reduce effective dimension. In: Dupire, B. (ed.), *Monte Carlo simulation in finance*. London: Risk Publications, 1999, p. 301–314.
- Cranley, R.; Patterson, T. N. L. Randomizations of number theoretic methods for multiple integration. *Siam Journal on Numerical Analysis*, p. 904-914, 1976.
- Dowd, K. *Measuring market risk*. Wiley, 2002.
- Galanti, S.; Jung, A. Low-discrepancy sequences: Monte Carlo simulation of option prices. *The Journal of Derivatives*, p. 63–83, Fall 1997.
- Golub, G. H.; Van Loan, C. F. *Matrix computations*. Baltimore: John Hopkins University Press, 1983.
- Hull, J. *Options, futures and other derivative securities*. New York: Prentice-Hall, 2000.
- Jäckel, P. *Monte Carlo methods in finance*. New York: Wiley, 2002.
- Joe, S.; Kuo, F. Remark on algorithm 659: implementing Sobol's quasi-random sequence generator. *ACM transactions on mathematical software*. March 2003, p. 49-57.
- Jorion, P. *Value at risk, the new benchmark for managing financial risk*. New York: McGraw-Hill, 2001.
- Joy, C.; Boyle, P. P.; Tan, K. S. Quasi-Monte Carlo methods in numerical finance. *Management Science*, p. 926-938, 1996.
- Kemna, A. G. Z.; Vorst, A. C. F. A pricing method for options based on average asset values. *Journal of Banking and Finance*, p. 113-129, 1990.
- Lécot, C. A quasi-Monte Carlo method for the Boltzman equation. *Mathematics of Computation*, p. 621–644, 1991.
- Morokoff, W.; Caflich, R. A quasi-Monte Carlo approach to particle simulation of the heat equation. *Siam Journal on Numerical Analysis*, p. 1558-1573, 1991.

- _____. Quasi-random sequences and their discrepancies. *Siam J. Sci. Stat. Computing*, p. 1251-1279, 1994.
- _____. Quasi-Monte Carlo integration. *J. Comp. Phys.*, p. 218-230, 1995.
- Moskowitz, B.; Caflisch, R. Smoothness and dimension reduction in quasi-Monte Carlo methods. *Mathl. Comput. Modelling*, p. 37-54, 1996.
- Niederreiter, H.; Xing, C. P. Low-discrepancy sequences and global function fields with many rational places. *Finite Fields Applications*, p. 241-273, 1996.
- Owen, A. Lattice sampling revisited: Monte Carlo variance of means over randomized orthogonal arrays. *The Annals of Statistics*, p. 930-945, 1994a.
- _____. Monte Carlo variance of scrambled net quadrature. *Siam Journal on Numerical Analysis*, p. 1884-1910, 1994b.
- Papageorgiou, A.; Paskov, S. Deterministic simulation for risk management. *The Journal of Portfolio Management*, p. 122-127, May 1999.
- Paskov, S. H.; Traub, J. F. Faster valuation of financial derivatives. *The Journal of Portfolio Management*, Fall 1995.
- Press, W.; Teukolsky, S.; William, T. V.; Brian, P. F. *Numerical recipes in C*. 2nd edition. Cambridge: Cambridge University Press, 1992.
- Silva, M. E. Simulação de quase Monte Carlo em finanças: quebrando a maldição da dimensionalidade. *Resenha da BM&F*, Sep-Oct 2002.
- Sobol, I. M. Uniformly distributed sequences with additional uniform properties. *USSR Computational Mathematics and Mathematical Physics*, 1976.
- Tan, K. S.; Boyle, P. P. Applications of randomized low discrepancy sequences o the valuation of complex securities. *Journal of Economic Dynamics and Control*, p. 1747-1782, 2000.