

**Raport: Wycena opcji metodą Quasi – Monte Carlo**

**Autor: Dominik Winnicki**

**Spis treści**

Opis problemu .....	2
Wstęp teoretyczny .....	2
Liczby Haltona .....	4
Liczby Sobol'a .....	4
Ocena uzyskanych ciągów Haltona i Sobol'a.....	6
Rzuty punktów na płaszczyznę 2-wymiarową .....	6
Wycena opcji europejskiej typu „best of n” w modelu lognormalnym.....	8
Algorytm wyceny .....	8
Analiza wyników .....	9
Analiza zbieżności .....	11
Dokumentacja programów.....	13
Bibliografia.....	15

## Opis problemu

Celem projektu jest napisanie programu do wyceny opcji zależących od wielu instrumentów, wykorzystującego metodę Quasi – Monte Carlo (QMC). W programie wykorzystane zostaną algorytmy Haltona i Sobola generowania ciągów typu *low discrepancy sequence*.

Jako przykłady opcji zależących od wielu instrumentów rozważymy opcje typu „best of n”.

## Wstęp teoretyczny

Standardowym podejściem do wyceny instrumentów pochodnych zależących od wielu instrumentów bazowych jest metoda Monte-Carlo (MC). Bardzo ogólnikowo, metodę tę można streścić w następujących krokach:

- 1) Określenie modelu probabilistycznego opisującego instrumenty bazowe (np. model Blacka – Scholesa),
- 2) Wygenerowanie  $N$  wypłat instrumentu pochodnego
  - a. Wygenerowanie  $N \times d$  liczb pseudo-losowych z rozkładu jednostajnego na odcinku  $[0, 1]$ , gdzie  $d$  oznacza liczbę czynników ryzyka w zadanym modelu probabilistycznym,
  - b. Wyznaczenie wypłat za pomocą przekształcenia pseudo-losowych punktów z przestrzeni  $[0, 1]^d$ , przy czym przekształcenie to zadane jest przez przyjęty model probabilistyczny,
- 3) Obliczenie wartości oczekiwanej odpowiednio zdyskontowanych wypłat instrumentu pochodnego jako średnią arytmetyczną wygenerowanych  $N$  wypłat.

Rząd zbieżności klasycznej metody MC wynosi

$$\frac{1}{\sqrt{N}}.$$

Alternatywą dla metody MC są algorytmy QMC, a motywacją do ich stosowania jest fakt, że oferują one korzystniejszy rząd zbieżności postaci

$$C(d) \frac{(\ln N)^d}{N},$$

gdzie  $C(d)$  jest stałą zależną od wymiaru przestrzeni czynników ryzyka. Wadą znanych nam algorytmów QMC jest fakt, iż a priori nie jest możliwe wyznaczenie  $C(d)$ , co oznacza, że w przypadku problemów o dużym wymiarze, algorytmy QMC mogą nie dawać spodziewanego przyspieszenia zbieżności.

## Wycena opcji metodą Quasi – Monte Carlo

Zarówno w metodzie MC jak i w metodzie QMC klucz do skutecznego działania leży w zdolności algorytmu do generowania punktów w kostce  $[0, 1]^d$  w taki sposób, by rozkład  $N$  pierwszych wygenerowanych punktów w kostce był jak najbardziej równomierny. W idealnym przypadku powinna zachodzić równość

$$\frac{n_{S(y)}}{N} = \prod_{i=1}^d y_i,$$

gdzie  $y \in [0,1]^d$ ,  $S(y) = [0, y_1) \times \dots \times [0, y_d)$ ,  $n_{S(y)} = \sum_{k=1}^N 1_{\{x^{(k)} \in S(y)\}}$ , a ciąg  $x = (x^{(k)})_{k=1}^N$  to ciąg punktów z kostki  $[0,1]^d$  wygenerowany przez dany algorytm. Uznajemy, że algorytm jest prawidłowy, gdy zachodzi

$$\lim_{N \rightarrow \infty} \frac{n_{S(y)}}{N} = \prod_{i=1}^d y_i.$$

W klasycznej metodzie MC ciąg  $x$  otrzymuje się przy użyciu generatora liczb pseudo-losowych. W szczególności, od generatora liczb pseudo-losowych wymaga się, by ciąg spełniał warunek braku seryjnej korelacji. Z kolei, metoda QMC opiera się na pomysłe, że możliwe jest generowanie ciągu punktów z kostki jednostkowej takiego, że równość

$$\frac{n_{S(y)}}{N} \approx \prod_{i=1}^d y_i$$

jest spełniona dla każdego  $N$  w pewnym z góry ustalonym przybliżeniu. Wprowadza się *miarę rozbieżności (ang. discrepancy)*

$$D_N^{(d)} = \sup \left\{ y \in [0,1]^d : \left| \frac{n_{S(y)}}{N} - \prod_{i=1}^d y_i \right| \right\}$$

i definiuje się *ciągi o niskiej rozbieżności (ang. low discrepancy sequence)* jako ciągi spełniające nierówność

$$D_N^{(d)} \leq C(d) \frac{(\ln N)^d}{N}.$$

Do klasy ciągów typu low discrepancy należą tzw. ciągi Haltona i ciągi Sobol'a. Poniżej przedstawiono algorytmy ich generowania.

# Wycena opcji metodą Quasi – Monte Carlo

## Liczby Haltona

Algorytm generowania liczb Haltona

1. Ustalmy ciąg liczb naturalnych  $(\gamma_n)_{n=1}^N$  taki, że  $\gamma_n = n$ .
2. Dla każdego z wymiarów  $k = 1, 2, \dots, d$  wyznaczmy reprezentację liczby  $\gamma_n$  w bazie liczbowej wyznaczonej przez liczbę  $p_i$ , gdzie  $p_i$  to  $i$ -ta co do wielkości liczba pierwsza, tzn.

$$\gamma_n = \sum_{j=1}^m a_{nj} p_k^j, \quad m = \lceil \log_{p_k}(\gamma_n + 1) \rceil.$$

3. Element  $k$  wyrazu  $n$ -tego ciągu Haltona obliczamy ze wzoru

$$x_n = \sum_{j=1}^m a_{nj} p_k^{-j}.$$

## Liczby Sobol'a

Algorytm generowania liczb Sobol'a

1. Ustalmy ciąg liczb naturalnych  $(\gamma_n)_{n=1}^N$  taki, że  $\gamma_n = n$ .
2. Ustalenie wielomianów pierwotnych (ang. *primitive polynomials*). Dla każdego z wymiarów  $k = 1, 2, \dots, d$  ustalmy wielomian pierwotny stopnia  $g_k \leq 31$  z pierścienia GF[2] (wielomiany o współczynnikach 0 lub 1 z dodawaniem i mnożeniem modulo 2), tzn. dla każdego  $i$  dany jest wielomian

$$p_i(z) = \sum_{j=0}^{g_k} a_{kj} z^{g_k - j}, \quad a_{k0} = 1.^1$$

3. Ustalenie początkowych liczb kierunkowych (ang. *direction numbers*). Dla każdego z wymiarów  $k = 1, 2, \dots, d$  ustalmy  $g_k$  początkowe liczby kierunkowe

$$v_{kl}, \quad l = 1, 2, \dots, g_k.$$

Kolejne liczby kierunkowe ustalamy według formuły rekurencyjnej

$$v_{kl} = \frac{v_{k(l-g_k)}}{2^{g_k}} \oplus_2 \left[ \sum_{j=1}^{g_k} a_{kj} v_{k(l-j)} \right]^{\oplus_2}, \quad g_k < l \leq M,$$

przy czym symbol  $\oplus_2$  oznacza sumowanie modulo 2 (czyli operacja bitowa XOR), sumowanie w nawiasie kwadratowym to sumowanie modulo 2, a liczbę  $M$  wyznaczamy ze wzoru

$$M = \lceil \log_2(N + 1) \rceil.$$

4. Wyznaczenie liczb Sobol'a. Wyznaczamy składowe liczby Sobol'a za pomocą formuły

<sup>1</sup> Zakładamy, że każde  $g_k < 32$ , ponieważ 32 to standardowo stosowana we współczesnych komputerach długość pola bitowego używanego do reprezentowania liczb.

## Wycena opcji metodą Quasi – Monte Carlo

---

$$x_{nk} = \left[ \sum_{j=1}^{M_n} \gamma_n^j v_{kj} \right]^{\oplus_2},$$

gdzie  $\gamma_n^j$  oznacza  $j$ -tą (licząc od prawej) cyfrę liczby  $\gamma_n$  w zapisie dwójkowym, a liczbę  $M_n$  wyznaczamy ze wzoru

$$M_n = \lceil \log_2(\gamma_n + 1) \rceil.$$

## Ocena uzyskanych ciągów Haltona i Sobol'a

Na podstawie przedstawionych wyżej algorytmów przygotowano zostały funkcje generujące ciągi Haltona i Sobol'a (o nazwach odpowiednio *halton.m* i *sobol.m*).

W przypadku ciągów Sobol'a konieczne było określenie *wielomianów pierwotnych* i odpowiadających im *początkowych liczb kierunkowych*. W pracy [1] autor podaje wielomiany i początkowe liczby kierunkowe tylko dla 32 wymiarów. W pracy [2] przedstawiono dane dla 20 wymiarów. Na stronie internetowej [4] autorzy udostępniają wielomiany wraz z początkowymi liczbami kierunkowymi dla 1111 wymiarów, przy czym – zgodnie z deklaracją autorów – dane te spełniają tzw. *własność A Sobol'a*.<sup>2</sup>

Wygenerowano 252-wymiarowe ciągi Haltona i Sobol'a po 50000 wyrazów. Następnie przeprowadzono testy mające na celu sprawdzenie otrzymanych ciągów pod kątem przydatności do wykorzystania w wycenie metodą QMC.

## Rzuty punktów na płaszczyznę 2-wymiarową

Zgodnie z [1], jednym ze sposobów sprawdzenia wygenerowanych ciągów jest jakościowa ocena rzutów punktów na płaszczyznę 2-wymiarową dla różnych par wymiarów. Elementarnym warunkiem, jaki powinien spełniać ciąg nadający się do wyceny metodą QMC, jest równomierne pokrycie kwadratu  $[0, 1] \times [0, 1]$  przy rzucie na płaszczyznę dla każdej pary wymiarów dla danej liczby wygenerowanych punktów. Wadą tej metody jest fakt, że liczba możliwych rzutów wynosząca  $\binom{d}{2}$  jest ogromna dla dużych  $d$ . Mimo to, pokażemy przykłady takich rzutów, by zilustrować zarówno sytuacje, w których sformułowany jakościowy warunek jest spełniony, jak i te, w których ów warunek nie jest spełniony.

W Tabeli 1. przedstawiono przykładowe rzuty. Rysunki w pierwszej kolumnie przedstawiają równomiernie rozłożone punkty Haltona i Sobol'a na płaszczyźnie 1-2. W idealnym przypadku oczekuje się równomiernego rozłożenia również dla pozostałych par wymiarów. W przypadku liczb Haltona już dla niewielkich wymiarów pojawiają się poważne problemy. W szczególności oznacza to, że wyniki wyceny metodą QMC przy użyciu tych ciągów mogą dawać nieprawidłowe wyniki.

W przypadku ciągów Sobol'a problemem jest klasteryzacja punktów, którą obserwujemy na rysunku 6. Mimo, że punktu zdają się pokrywać cały kwadrat, to jednak oczywiste jest, że rozkład ten nie jest jednostajny.

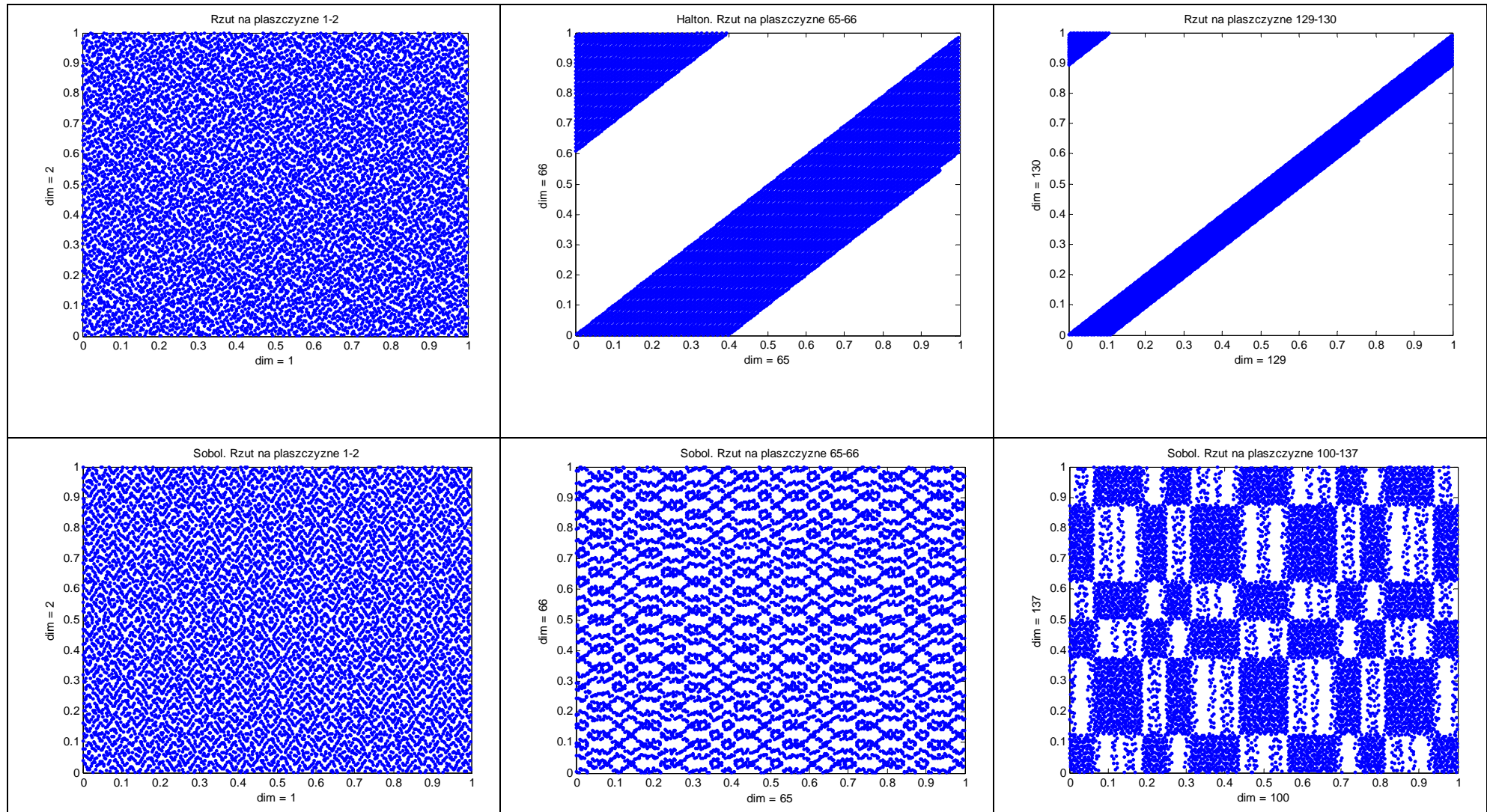
Prezentowany tu test nie jest idealnym narzędziem diagnostycznym. Nawet jeżeli warunek równomierności rozkładu punktów w płaszczyźnie 2-wymiarowej byłby spełniony, to nie gwarantowałyby to jednostajności rozkładu w większych wymiarach. Łatwo się domyśleć, że jeżeli nieprawidłowości pojawiają się przy rzutach 2-wymiarowych, to przy rzutach na hiperpłaszczyzny o wyższych wymiarach mogą kryć kolejne problemy.

---

<sup>2</sup> Ciąg  $d$ -wymiarowy  $x$  spełnia własność A Sobol'a, jeżeli dla każdego  $j = 0, 1, \dots$ , każdy punkt  $x_i$ ,  $j2^d \leq i < (j + 1)2^d$ , należy do innej spośród hiper-kostek  $\left[\frac{a_1}{2}, \frac{a_1+1}{2}\right) \times \dots \times \left[\frac{a_d}{2}, \frac{a_d+1}{2}\right)$ ,  $a_k \in \{0, 1\}$ ,  $k = 1, \dots, d$ .

# Wycena opcji metodą Quasi – Monte Carlo

Tabela 1. Przykładowe rzuty punktów Haltona i Sobol'a na płaszczyznę 2-wymiarową.



## Wycena opcji europejskiej typu „best of n” w modelu lognormalnym

Zgodnie z [5], opcje typu „best of n”, jako opcje zależące od wielu instrumentów bazowych, zaliczane są do klasy opcji *rainbow*. W przypadku europejskiej opcji „best of n” mamy do czynienia z opcją zależącą od cen  $n$  instrumentów bazowych w chwili wygaśnięcia. Wypłata ma postać:

$$\text{Payoff} = \max(\max(S_1(T), \dots, S_n(T)), K),$$

gdzie  $S_i(T)$  to ceny instrumentów bazowych w chwili wygaśnięcia opcji  $T$ , a  $K$  to cena wykonania.

Naszym celem jest wykorzystanie tego przykładu do zademonstrowania możliwości ciągów typu low discrepancy, takich jak liczby Haltona i liczby Sobol’a, w wycenie instrumentów o wielowymiarowej strukturze czynników ryzyka metodą Quasi Monte Carlo. Aby uzyskać wyraźny obraz tych możliwości, będziemy porównywać uzyskiwane wyniki do tych otrzymanych przy zastosowaniu klasycznej metody Monte Carlo.

W programach *best\_of\_nQMC.m* oraz *best\_of\_nMC.m* został zaimplementowany algorytm wyceny opcji „best of n”, którego szczegóły zostały opisane poniżej. Kody programów są niemal takie same. Różnica polega na tym, że w przypadku *best\_of\_nQMC.m* ścieżki cen generowane są za pomocą liczb Haltona i Sobol’a, a w przypadku *best\_of\_nMC.m* – za pomocą wbudowanego w Octave generatora liczb pseudo-losowych.

## Algorytm wyceny

### 1. Dane wejściowe:

$n$	- liczba instrumentów bazowych,
$S_1, \dots, S_n$	- początkowe ceny instrumentów bazowych,
$r$	- stopa procentowa pozbawiona ryzyka,
$\Sigma$	- macierz kowariancji procesu $\ln S$ ,
$T$	- czas do wygaśnięcia,
$M$	- liczba symulacji.

### 2. Wygenerowanie cen końcowych

#### a. Metoda QMC

- Wygenerowanie  $M$  punktów  $n$ -wymiarowych Haltona lub Sobol’a,
- Transformacja za pomocą odwróconej dystrybuanty rozkładu normalnego,
- Wyznaczenie cen końcowych.

#### b. Metoda MC

- Wygenerowanie  $M$  punktów  $n$ -wymiarowych za pomocą generatora liczb pseudo-losowych z rozkładu normalnego (przy pomocy funkcji *randn*),
- Wyznaczenie cen końcowych.

### 3. Obliczenie wypłat opcji „best of n”.

### 4. Obliczenie średniej arytmetycznej zdyskontowanych wypłat.

### 5. Wyznaczenie przedziału ufności dla otrzymanej ceny opcji.



# Wycena opcji metodą Quasi – Monte Carlo

## Analiza wyników

Rozważmy wycenę opcji „best of n” w przypadkach, gdy liczba instrumentów bazowych wynosi kolejno 2, 5, 10, 20, 50, 100 i 150. Pozostałe dane wejściowe ustalamy z wartościami opisanymi w tabeli 2.

Tabela 2. Dane wejściowe.

Typ danych	Wartość
Początkowa cena instrumentów	$S_i = 100, \quad i = 1, \dots, n$
Stopa procentowa wolna od ryzyka	$r = 5\%$
Macierz kowariancji	$\Sigma = \begin{bmatrix} \sigma^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma^2 \end{bmatrix}, \quad \sigma = 30\%$
Czas do wygaśnięcia	$T = 1$
Liczba symulacji	$M = 50000^3$

Następnie przeprowadzamy symulacje z wykorzystaniem programów *best\_of\_nQMC.m* oraz *best\_of\_nMC.m*. Dla każdej z opcji obliczamy jej cenę klasyczną metodą MC oraz metodą QMC dla liczb Haltona i dla liczb Sobol’a. Wyniki w postaci wyestymowanej ceny oraz współczynnika dokładności  $\frac{b}{\sqrt{M}}$  zapisujemy w tabeli 3.<sup>4</sup>

Tabela 3. Wyniki symulacji.

$n$	MC		Halton		Sobol’	
	Cena	$\frac{b}{\sqrt{M}}$	Cena	$\frac{b}{\sqrt{M}}$	Cena	$\frac{b}{\sqrt{M}}$
2	24.8185	0.1118	24.5401	0.1188	24.5452	0.1189
5	43.3117	0.1282	43.3295	0.1278	43.1773	0.1315
10	58.8744	0.1267	58.9063	0.1260	58.4893	0.1294
20	74.3764	0.1254	74.0666	0.1226	73.9444	0.1261
50	94.6980	0.1242	92.2596	0.1322	94.5537	0.1236
100	109.5445	0.1231	104.4023	0.1428	109.4446	0.1219
150	118.2493	0.1235	111.2607	0.1527	118.0307	0.1210

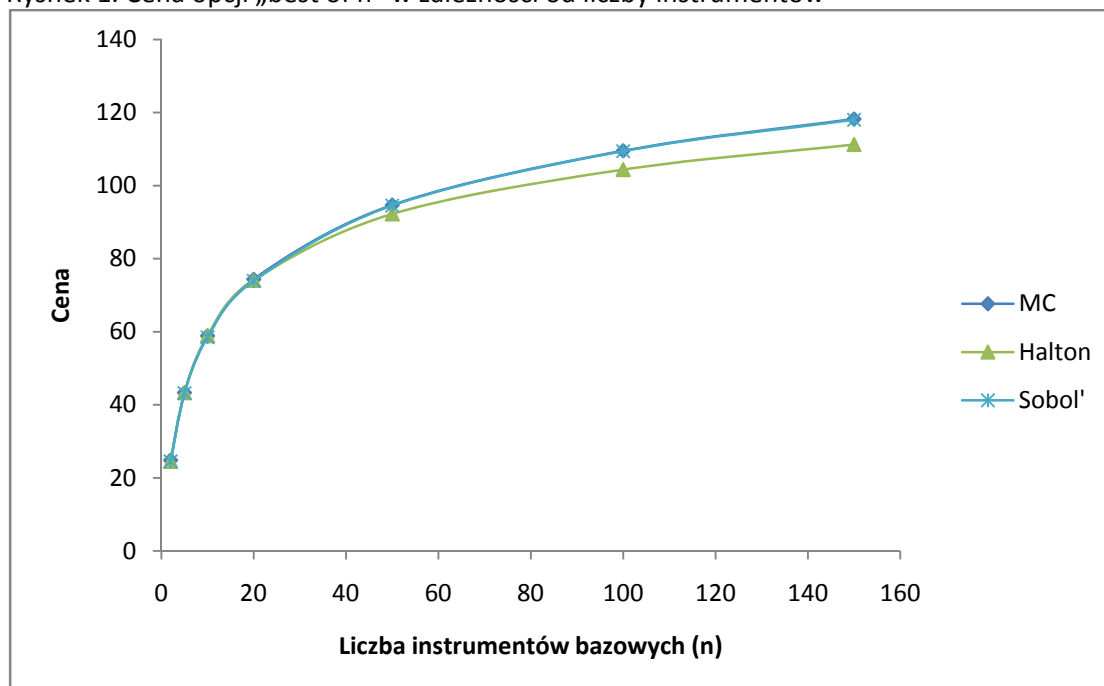
Zauważmy, że ceny uzyskiwane w kolejnych przypadkach dla wszystkich metod są podobne gdy  $n \leq 20$ . Przy  $n = 50$  obserwujemy wyraźne odchylenie ceny wyliczonej metodą QMC z liczbami Haltona od pozostałych. Co więcej, odchylenie to rośnie wraz ze wzrostem wymiaru problemu. Z kolei algorytm QMC z liczbami Sobol’a daje wyniki bliskie tym uzyskiwanym w metodą MC nawet dla dużych  $n$ . Rezultaty te przedstawiono na rysunku 1.

<sup>3</sup> Liczbę symulacji ustalamy na poziomie 50000 ze względu na ograniczenie liczby punktów w wygenerowanym ciągu Sobol’a. Algorytm wyznaczający liczby Sobol’a jest stosunkowo wolny z uwagi na fakt, iż operacje bitowe w Octave nie mają charakteru wektorowego i nie da się obejść konieczności użycia pętli.

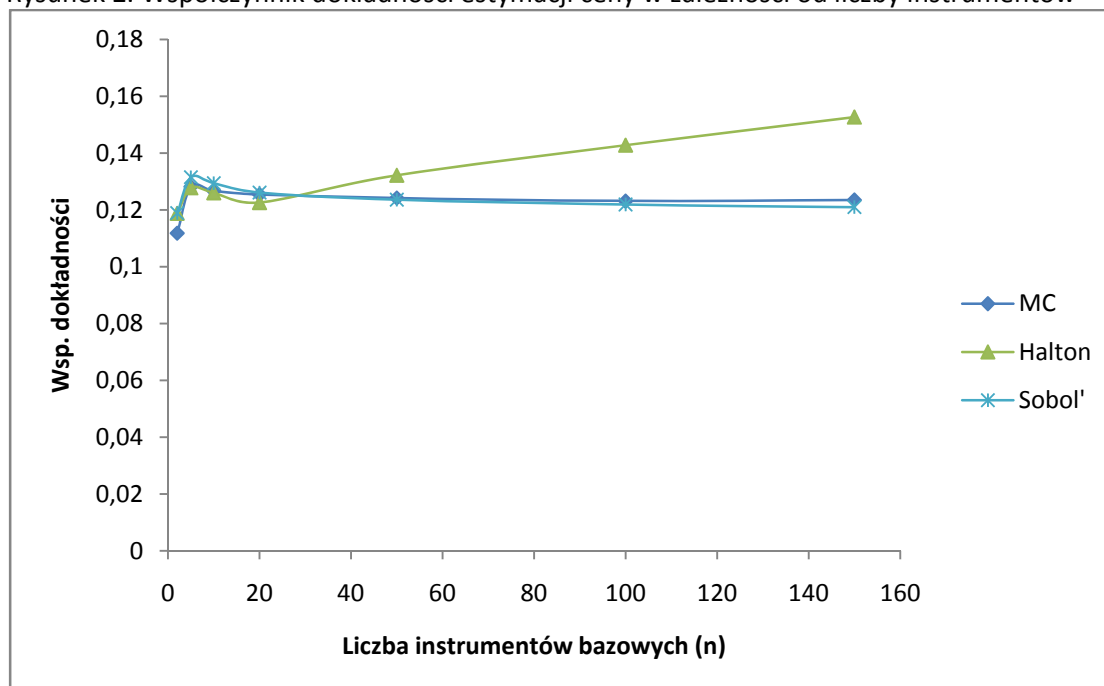
<sup>4</sup> Liczba  $b$  we wzorze na współczynnik dokładności oznacza wariancję wysymulowanych zdyskontowanych wypłat opcji.

# Wycena opcji metodą Quasi – Monte Carlo

Rysunek 1. Cena opcji „best of n” w zależności od liczby instrumentów



Rysunek 2. Współczynnik dokładności estymacji ceny w zależności od liczby instrumentów



# Wycena opcji metodą Quasi – Monte Carlo

## Analiza zbieżności

Na koniec przedstawimy wyniki analizy zbieżności schematów QMC dla liczb Haltona i Sobol'a w zależności od liczby wykonanych symulacji przy różnych liczbach instrumentów bazowych. Wyniki przedstawiono w tabelach 4. i 5.

Tabela 4. Współczynniki dokładności  $\frac{b}{\sqrt{M}}$  dla cen otrzymanych metoda QMC z liczbami Haltona.

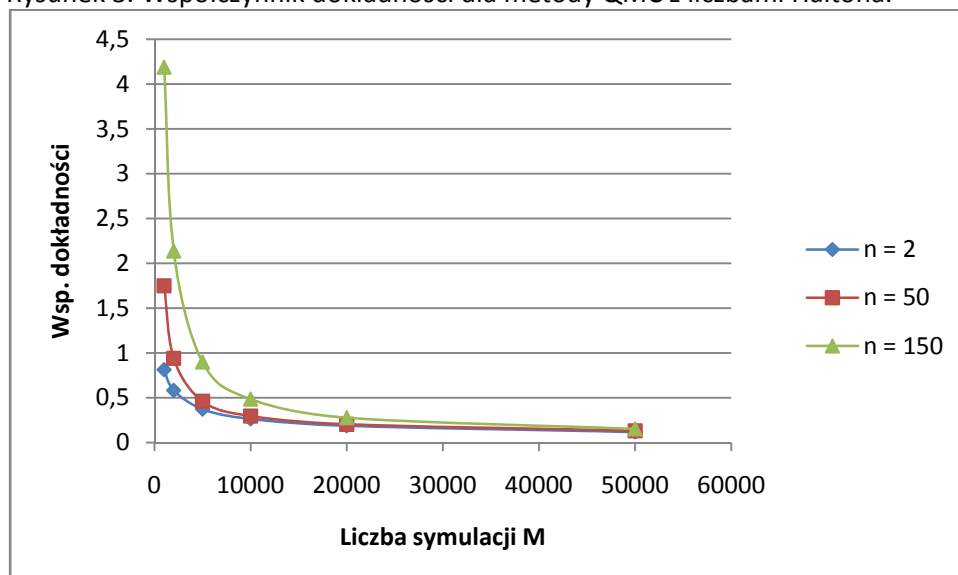
		Liczba symulacji $M$					
		1000	2000	5000	10000	20000	50000
$n$	2	0.8140	0.5832	0.3724	0.2644	0.1877	0.1188
	50	1.7497	0.9409	0.4612	0.2965	0.2041	0.1322
	150	4.1879	2.1369	0.8986	0.4856	0.2783	0.1527

Tabela 5. Współczynniki dokładności  $\frac{b}{\sqrt{M}}$  dla cen otrzymanych metoda QMC z liczbami Sobol'a.

		Liczba symulacji $M$					
		1000	2000	5000	10000	20000	50000
$n$	2	0.8198	0.5853	0.3724	0.2650	0.1878	0.1189
	50	0.7666	0.5675	0.3722	0.2705	0.1934	0.1236
	150	0.7021	0.5275	0.3516	0.2590	0.1881	0.1210

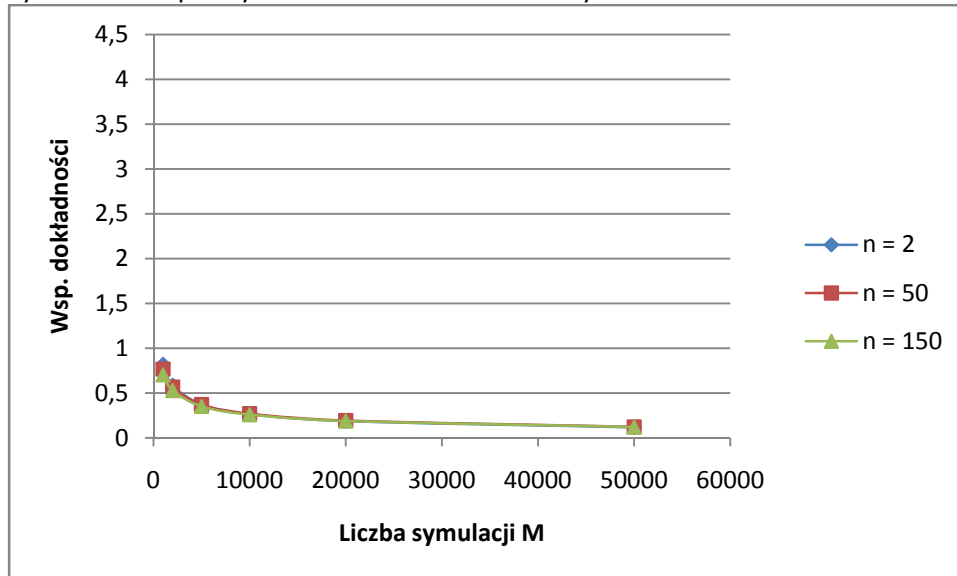
Dane przedstawione w tabelach zilustrowano na rysunkach 3. i 4. Zgodnie z oczekiwaniami, dokładność obu metod, mierzona współczynnikiem  $\frac{b}{\sqrt{M}}$ , rośnie wraz ze wzrostem liczby przeprowadzonych symulacji. Warto zauważyć, że przy metodzie QMC z liczbami Sobol'a rozrzut symulowanych wypłat jest znacząco mniejsza niż w przypadku liczb Haltona.

Rysunek 3. Współczynnik dokładności dla metody QMC z liczbami Haltona.



## Wycena opcji metodą Quasi – Monte Carlo

Rysunek 4. Współczynnik dokładności dla metody QMC z liczbami Sobol'a.



# Wycena opcji metodą Quasi – Monte Carlo

---

## Obsługa programów

Do niniejszego raportu załączono następujące programy i pliki Matlab/Octave:

Katalog `~/Programy/Generowanie ciagow`

1. Plik danych z wielomianami pierwotnymi i początkowymi liczbami kierunkowymi wykorzystywany przez program `sobol.m` do generowania ciągów Sobol'a:
  - a. `init1.mat`
2. Programy do generowania ciągów Haltona i Sobol'a:
  - a. `halton.m`
  - b. `sobol.m`

Katalog `~/Programy/Wycena`

3. Programy realizujące wycenę opcji typu „best of n” metodami QMC i MC:
  - a. `best_of_nQMC.m`
  - b. `best_of_nMC.m`
4. Plik danych wejściowych dla programu głównego:
  - a. `data.mat`
5. Program główny – realizujący wycenę wszystkim trzema metodami naraz
  - a. `best_of_n.m`

Ponieważ algorytmy generujące liczby Haltona i Sobol'a są stosunkowo wolne, procedura ich generowania została wyłączona z programu głównego realizującego wycenę opcji. Zanim możliwe będzie wykorzystanie programu głównego, należy wygenerować macierze H i S liczb Haltona i Sobol'a. Procedura:

1. Komenda `H = halton(50000, 252)` generuje 50000 punktów Haltona w kostce 252-wymiarowej. Oczywiście można podać inne argumenty funkcji.
2. Komenda `S = sobol(50000, 252)` generuje 50000 punktów Sobola w kostce 252-wymiarowej.

Uwaga: program `sobol.m` korzysta z danych w pliku `init1.mat`.

Aby można było przeprowadzić wycenę za pomocą programu `best_of_n.m`, konieczne jest wcześniejsze załadowanie do pamięci Octave otrzymanych powyżej macierzy H i S. Wygodnym rozwiązaniem jest zapisanie otrzymanych macierzy H i S do pliku danych. Nie dołączam tych plików danych wraz z pozostałymi programami, ponieważ zajmują one przeszło 100MB.

Procedura korzystania z programu głównego:

1. Załadować macierze H i S do pamięci.
2. Przygotować plik z pozostałymi danymi wejściowymi `data.mat`.
3. Wywołać program główny komendą:

`>best_of_n(H,S)`

Uwaga: program główny zawiera również kody funkcji `best_of_nQMC.m` i `best_of_nMC.m`.

Do programów dołączono przykładowy plik `data.mat`. Składa się on z następujących zmiennych:

## Wycena opcji metodą Quasi – Monte Carlo

---

- $n$  – liczba instrumentów,
- $S_0$  – wektor początkowych cen instrumentów,
- $K$  – cena wykonania,
- $r$  – stopa procentowa pozbawiona ryzyka,
- $\sigma$  – wektor zmienności cen,
- $\text{CorrM}$  – macierz korelacji,
- $T$  – czas do wygaśnięcia,
- $M$  – liczba symulacji.

W nagłówkach załączonych programów opisano dokładnie format wejścia i wyjścia. Dodatkowo, dokładny opis działania każdego z programów został zawarty w komentarzach w kodzie.

## Bibliografia

- [1] Jaeckel P., *Monte Carlo Methods in Finance*, John Wiley & Sons, 2002
- [2] London J., *Modelling derivatives in C++*, John Wiley & Sons, 2005
- [3] Glasserman P., *Monte Carlo Methods in Financial Engineering*, Springer, 2004
- [4] Joe S., Kuo F., *Sobol Sequence Generator*, <http://web.maths.unsw.edu.au/~fkuo/sobol/>
- [5] *Global Derivatives v3.0*, <http://www.global-derivatives.com/>