PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 1088 — 2005
Inwestycje finansowe i ubezpieczenia – tendencje światowe a polski rynek

Tomasz Oczadły

Akademia Ekonomiczna we Wrocławiu

LICZBY PSEUDOLOSOWE A SYMULACYJNA WYCENA OPCJI

1. Generowanie liczb pseudolosowych

Dziedziny nauki, jakimi są rachunek prawdopodobieństwa i statystyka, są oparte na koncepcjach przestrzeni probabilistycznej oraz zmiennych losowych. Pojęcie zmiennej losowej, mimo że jest podstawą rachunku prawdopodobieństwa, w praktyce nie jest możliwe do zaimplementowania na współczesnych maszynach cyfrowych. Maszyny te wykorzystują koncepcje deterministyczne. Każdy wynik ich działania jest przyczyną zdarzeń mających miejsce w rejestrach pamięci tych maszyn w cyklach wcześniejszych. Stosując metody *Monte Carlo*, musimy dysponować liczbami losowymi o takich właściwościach, które umożliwią nam symulowane interesującego nas procesu. Konsekwencją jest odwołanie się do metod deterministycznych generowania liczb losowych lub dokładniej – liczb pseudolosowych. W procesie generowania liczb losowych możemy wyróżnić dwa podstawowe etapy:

- generowanie liczb z rozkładu jednostajnego na przedziale $(0,1)$,
- transformację tych liczb do interesującego nas rozkładu.

Głównym etapem w tym procesie jest generowanie liczb pseudolosowych z rozkładu jednostajnego na przedziale (0,1) (oznaczmy: $U(0,1)$). Często pojęcie generowania liczb pseudolosowych jest określeniem procesu generowania liczb z rozkładu jednostajnego. W artykule tym pominę etap drugi, którym jest transformacja liczb z rozkładu $U(0,1)$ do interesującego nas rozkładu (zainteresowanych czytelników odsyłam do literatury [Gentle, Haerdle, Mori 2004; Knuth 1981; Zieliński 1970]).

Metody generowania liczb losowych z rozkładu $U(0,1)$ można podzielić na dwie podstawowe klasy. Są to metody wykorzystujące zdarzenia zachodzące w świecie rzeczywistym, takie jak proces rozpadu atomowego (wykorzystanie liczników promieniowania), oraz metody generowania liczb pseudolosowych wykorzystujące odpowiednie algorytmy zaimplementowane na komputerze. W pierwszej grupie metod mówimy o generatorach fizycznych liczb losowych. W przypadku tych generatorów musimy odwołać się do zjawiska fizycznego, które jesteśmy w stanie zmierzyć oraz możemy przyjąć, iż otrzymane pomiary są niezależne. Często generatory fizyczne składają się z dwóch elementów: elementu generującego ciągi „zer" i „jedynek" oraz elementu składającego te „zera" i „jedynki" w liczby binarne o odpowiedniej długości i organizacji, uzależnionej od architektury komputera. Cały problem sprowadza się więc, w tej metodzie, do wygenerowania ciągu bitów, w którym prawdopodobieństwo pojawienia się „0" jest równe prawdopodobieństwu pojawienia się wartości „1", a zdarzenie, jakim jest pojawienie się „0", jest niezależne od zdarzenia, jakim jest pojawienia się „1". Zwykle jako źródła takich ciągów stosuje się bądź liczniki promieniowania wraz ze źródłem promieniowania, bądź urządzenia rejestrujące szumy elektronowe [Zieliński 1970].

Metody te, pomimo iż dają szanse na otrzymanie liczb prawdziwie losowych, mają duże ograniczenia w stosowaniu ze względu na: (1) problemy związane ze stworzeniem oprzyrządowania, (2) powolnością tych metod.

Drugą grupą generatorów liczb losowych są programowe generatory liczb losowych. Ze względu na wykorzystanie w tych generatorach algorytmów deterministycznych, aby podkreślić to, iż liczby otrzymane z tych generatorów nie są liczbami losowymi w takim sensie, jak liczby pochodzące z generatorów fizycznych, generatory te często nazywa się generatorami liczb pseudolosowych, a otrzymane w ten sposób liczby – liczbami pseudolosowymi. Programowe generatory liczb pseudolosowych (liczb o rozkładzie równomiernym) buduje się zwykle według następującego schematu:

- ustala się początkowe wyrazy ciągu $x_1, x_2, \cdots, x_k$,
- jeżeli już wygenerowano $(n-1)$ liczb pseudolosowych, to liczbę x_n ustala się według wzoru:

$$x_n = f(x_{n-1}, x_{n-2}, \cdots, x_{n-k}).$$

Każdy z elementów ciągu x_n przekształca się w ciąg u_n za pomocą funkcji $u_i = g(x_i)$, aby uzyskać ciąg liczb u_n zawierający się w przedziale (0,1].

Ponieważ funkcja f jest deterministyczna, istnieją takie $l > 0$ (parametr aperiodyczności) i $j > 0$, że $x_{l+j} = x_l$, tzn. ciąg x_n jest ciągiem okresowym. Okres generatora możemy wyrazić za pomocą wzoru: $P = min\{i : X_j = X_0, j > 0\}$. Najstarszym generatorem opartym na algorytmie był generator zaproponowany przez J. von Neumanna, nazwany generatorem kwadratowym. Generator ten przybiera postać:

$$x_n = [x_{n-1}^2 * 10^{-m/2}] - [x_{n-1}^2 * 10^{-3m/2}] * 10^m, \tag{1}$$

gdzie: $[a]$ oznacza część całkowitą liczby a. Generator ten nie jest stosowany współcześnie ze względu na stosunkowo krótki okres.

Najszerszą grupą generatorów liczb pseudolosowych wykorzystywanych współcześnie stanowią generatory oparte na liniowej rekursji:

$$x_n = (a_0 x_{n-1} + a_1 x_{n-2} + \cdots + a_k x_{n-k-1} + c) mod M, \qquad (2)$$

gdzie: $a_0, a_1, \cdots, a_k, M$ i c są liczbami całkowitymi, a mod b zaś oznacza resztę dzielenia liczby a przez liczbę b. Generator (2) ma maksymalny okres P rzędu $M^k - 1$ tylko wtedy, gdy M jest liczbą pierwszą, a wielomian charakterystyczny $P(z) = z^k - a_1 z^{k-1} - \cdots - a_k$ jest nie rozkładalny nad ciałem liczb rzeczywistych.

Możemy wyróżnić następujące podgrupy generatorów (2):

- generator liniowy (Fibonacciego)

$$x_n = x_{n-2} + x_{n-1} mod M, \qquad (3)$$

- generator multyplikatywny

$$x_n = (a x_{n-1}) mod M, \qquad (4)$$

- generator mieszany

$$x_n = (a x_{n-1} + c) mod M. \qquad (5)$$

Generatory należące do grupy (2) są generatorami okresowymi. Długość okresu generatora jest uzależniona między innymi od wielkości liczby M. W zależności od architektury komputera, którym dysponujemy, jesteśmy w stanie przechować w nim liczbę o skończonej wielkości. Powoduje to problemy w implementacji generatorów o dużym okresie. W takich przypadkach można zastosować kombinację generatorów, polegającą na łączeniu dwóch lub większej liczby prostych generatorów, otrzymując w ten sposób generator o lepszych właściwościach statystycznych niż generatory wejściowe. Badania w zakresie jakości generatorów kongruentnych, zarówno liniowych (ang. *LCGs*), jak i mieszanych (ang. *MRGs*) pokazują, iż generatory tego typu nie przechodzą testów wykorzystujących $\sqrt{P}$ wygenerowanych liczb pseudolosowych, gdzie P jest okresem testowanego generatora. Coraz częściej przyjmuje się, iż wykorzystywane w symulacjach *Monte Carlo* generatory powinny mieć okres rzędu 2^{100} lub większy [L'Ecuyer 1999].

Przykładem prostego generatora łączonego *(ang. CMRGs)* jest zaproponowany przez Wichman i Hill [Sobol 1994] generator przeznaczony do komputerów o 16--bitowej arytmetyce, mający okres $P = 6{\cdot}9 * 10^{12}$. Generator ten składa się z trzech generatorów multiplykatywnych (4):

$$x_n' = (171 x_{n-1}) mod 30269, x_n'' = (172 x_{n-1}) mod 30307, x_n''' = (170 x_{n-1}) mod 30323. \qquad (6)$$

Liczbę pseudolosową x_n otrzymujemy, dokonując następującego złożenia:

$$x_n = \frac{x_n'}{30269} + \frac{x_n''}{30307} + \frac{x_n'''}{30323}. \qquad (7)$$

Okazuje się, iż kombinacje generatorów mogą produkować ciągi, które są bardziej równomiernie rozłożone oraz bardziej niezależne w porównaniu z pojedynczymi generatorami, cechują się również większym okresem. W pracy L'Ecuyer

[1999] zaproponował oraz przebadał dwa możliwe łączenia generatorów (2) kongruentnych. Weźmy pod uwagę J generatorów typu (2) oraz zdefiniujmy J liczb δ, tj.: $\delta_1, \delta_2, \cdots, \delta_J$, z których każda jest liczbą względnie pierwszą do M_j dla M z (2) oraz $j = 1, \cdots, J$. Wtedy możemy zbudować generator łączony:

$$w_n = (\sum_{j=1}^{J} \delta_j \frac{x_{j,n}}{M_j}) mod 1, \tag{8}$$

$$z_n = (\sum_{j=1}^{J} \delta_j x_{j,n}) mod M_1 u_n = z_n / m_1. \tag{9}$$

Generator (8), (9) jest porównywalny z generatorem (2) dla $M = m_1 * \cdots * m_J$.

2. Symulacyjna metoda wyceny opcji

Podstawową metodą przy wycenie opcji standardowych jest model Blacka-Scholesa. W przypadku bardziej złożonych kontraktów istnieje bardziej bezpośrednie podejście, które często bywa wykorzystane podczas symulacji komputerowych, podejście arbitrażowe [Maksymiuk, Gątarek 1998]. Niech $X(t)$ będzie ceną akcji w chwili t, a r stopą procentową o ciągłej kapitalizacji. Warunek braku arbitrażu oznacza, że proces stochastyczny Y zdefiniowany przez:

$$Y(t) = \exp(-rt) X(t) \tag{10}$$

jest martyngałem. Dla $X(t)$ spełniającego stochastyczne równanie różniczkowe:

$$dX(t) = rX(t)dt + \sigma X(t) dB(t) \tag{11}$$

proces $Y(t)$ jest martyngałem. Zauważmy, iż w przypadku wyceny arbitrażowej stopa zwrotu z instrumentu finansowego jest uzależniona od wolnej od ryzyka stopy procentowej $(rX(t)dt)$ oraz premii za ryzyko $(\sigma X(t) dB(t))$. Fakt ten sprowadza się do wniosku, iż średnia stopa zwrotu z instrumentów finansowych powinna być równa stopie procentowej wolnej od ryzyka. Rozwiązaniem równania (11) jest :

$$X(t) = X_0 \exp[(r - \frac{1}{2}\sigma^2)t + \sigma B(t)]. \tag{12}$$

Obliczenie ceny opcji europejskiej w ujęciu arbitrażowym sprowadza się do obliczenia wartości oczekiwanej z wartości wewnętrznej opcji przy założeniu prawdziwości procesu (12) w znanym rozkładzie (np. Gaussa):

$$c(t) = \exp(-r(T-t)) EH(X(T)) = \exp(-r(T-t)) \int_{-\infty}^{+\infty} H(\exp[(r - \frac{1}{2}\sigma^2)t + \sigma x]) f(x) dx, \tag{13}$$

gdzie: T jest dniem wygaśnięcia opcji, H funkcją jej wartości wewnętrznej, a funkcja f z góry określoną gęstością rozkładu – najczęściej gaussowskiego ze średnią 0 i wariancją 1.

Stosując metodę Monte Carlo, możemy symulacyjnie wyznaczyć cenę europejskiej opcji standardowej na podstawie wzoru (13), stosując następujący algorytm [Glasserman 2002; Jäckel 2003]:

1. Generujemy k niezależnych trajektorii procesu Wienera o długości n. Macierz takich zmiennych losowych oznaczmy jako $W^{n,k}$.
2. Bazując na wygenerowanych trajektoriach procesu Wienera, tworzymy ciąg:

$$X^{n,k}(t) = X(0)exp\{rt + \sigma W^{n,k}(t) - \frac{\sigma^2}{2}t\}. \tag{14}$$

3. Wyznaczamy cenę opcji, korzystając ze wzoru:

$$c(t) = \frac{e^{-r(T-t)}}{N}\sum_{k=1}^{N} H(X^{n,k}(T)), \tag{15}$$

gdzie: T jest dniem wygaśnięcia opcji, H funkcją wartości wewnętrznej opcji, r wolną od ryzyka stopa procentową, t dniem, w którym dokonujemy wyceny opcji, σ zmiennością procesu instrumentu bazowego.

Często w czasie symulacji *Monte Carlo* etapy (1) oraz (2) w schemacie przedstawionym powyżej są łączone. Zamiast generowania macierzy zmiennych losowych $W^{n,k}$ generowane są trajektorie geometrycznego ruchu Browna (ang. *GBM*):

$$X(t) = X_0 + r\int_0^t X(s)ds + \sigma\int_0^t X(s)dB(s). \tag{16}$$

Aby numerycznie rozwiązać równanie (16), należy przekształcić to równanie do postaci dyskretnej. Jedną z najprostszych metod numerycznej aproksymacji równań całkowych postaci (16) jest *schemat Eulera* lub wersja rozszerzona tego schematu – *schemat Milsteina*, który wykorzystałem w przeprowadzonych obliczeniach [Jankowska, Jankowski 1991; Palczewski 1999].

3. Wpływ metod generowania liczb pseudolosowych na wycenę opcji

Stosując metody *Monte Carlo* do rozwiązywania zagadnień matematycznych, musimy odwołać się do ciągu liczb $x_1, x_2, \cdots, x_n$ uzyskanych z pewnego generatora liczb pseudolosowych. Zakładamy, iż ciąg ten jest próbką z populacji o zadanym rozkładzie. W sytuacji gdy ciąg liczb nie pochodzi z populacji o założonym przez nas *a priori* rozkładzie, otrzymane rozwiązanie metodą *Monte Carlo* jest błędne. W celu zweryfikowania hipotezy, czy wykorzystywane przez nas liczby pochodzą z populacji o założonym rozkładzie, stosuje się odpowiednie testy statystyczne. Należy jednak zauważyć, iż nie jesteśmy w stanie stworzyć zbioru testów, który w pełni zweryfikuje tę hipotezę. Jeżeli dany ciąg nie spełnia własności przez nas założonych, to algorytm generujący liczby pseudolosowe jest błędny. Jeżeli jednak dany ciąg spełnia założone przez nas własności (testy), nie wynika z tego, iż ma on również wszystkie inne własności, które moglibyśmy sformułować dla liczb losowych[1]. Testy statystyczne przeznaczone do weryfikacji generatorów liczb pseudolosowych można podzielić na trzy grupy: (1) testy zgodności rozkładu, (2) testy niezależności, (3) zadania kontrolne. W artykule tym korzystam z generatorów liczb pseudolosowych, które poddane były różnorakim testom z grup 1 i 2. Wybrane generatory zostaną więc przetestowane metodą tzw. zadań kontrolnych. Testowanie generatorów ze względu na zadania kontrolne polega na rozwiązywaniu metodami *Monte Carlo* określonych zadań z użyciem liczb losowych. Porównując otrzymane wyniki do wyników otrzymanych droga teoretyczną, możemy wykazać wpływ zastosowania różnych generatorów na błędy rozwiązania otrzymanego metodą *Monte Carlo*. W naszym przypadku zadaniem kontrolnym będzie wycenienie opcji europejskiej z zastosowaniem metodologii Blacka-Scholesa. W celach porównawczych zostaną wykorzystane następujące generatory liczb pseudolosowych:

- standardowy generator *Rand()*, wykorzystywany w Matlabie 6.5,
- generator mieszany zaprezentowany przez Marsaglia (1972) [Zieliński 1997]:

$$x_n = (69069x_{n-1} + 1) mod 2^{32}, \tag{17}$$

- generator łączony zaproponowany przez L'Ecuyera [1999] składający się z dwóch generatorów (2):

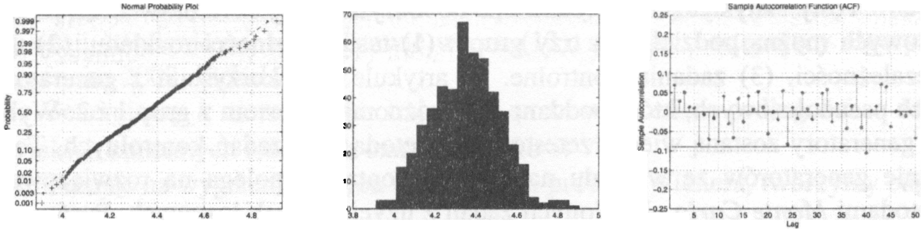
$$\begin{aligned} x_n &= (1403580x_{n-2} - 810728x_{n-3}) \bmod 2^{32} - 209, \\ x_n &= (527612x_{n-2} - 1370589x_{n-3}) \bmod 2^{32} - 22853. \end{aligned} \tag{18}$$

W celu zweryfikowania wpływu generatorów liczb pseudolosowych dokonana zostanie wycena metodą symulacyjną, z zastosowaniem przedstawionych powyżej generatorów liczb pseudolosowych, europejskiej opcji kupna. Otrzymane wyniki porównam zarówno do ceny teoretycznej, otrzymanej z równania Blacka-Scholesa, jak i do cen otrzymanych z symulacji wykorzystujących opisane generatory. Wyceniając opcję metodą symulacyjną, wykonałem 500 niezależnych symulacji. Każdą symulację przeprowadziłem z zastosowaniem metody *Monte Carlo*, generując 1500 trajektorii procesu (16), dyskretyzując proces stochastyczny *schematem Milsteina* z krokiem dyskretyzacji 0,005.

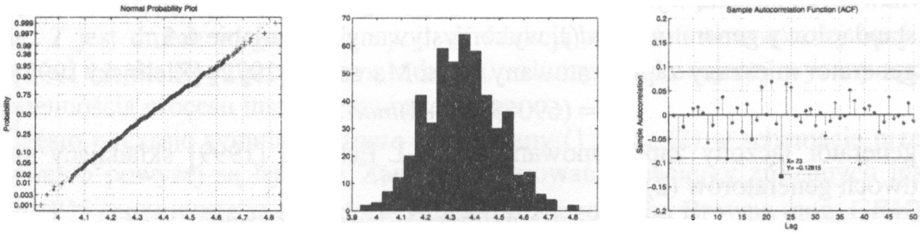
Rozważmy europejską opcję kupna wystawioną na 2 miesiące. W momencie wystawienia opcji, cena instrumentu, na który opcja została wystawiona, wynosi 100 zł, cena wykonania opcji wynosi również 100 zł. Wolna od ryzyka stopa procentowa wynosi 9% w skali roku, a zmienność ceny instrumentu 25% w skali roku. Po wykorzystaniu równania Blacka-Scholesa [Hull 1999] cena takiej europejskiej opcji kupna powinna wynosić 4,4 zł. Poniżej prezentuję: (1) wykresy kwartyl-kwartyl porównujące kwartyle rozkładu empirycznego cen opcji do rozkładu normalnego, dla każdego generatora, (2) histogram cen opcji uzyskanych metodą symulacyjną dla każdego generatora, (3) wykres autokorelacji szeregu cen opcji.

---

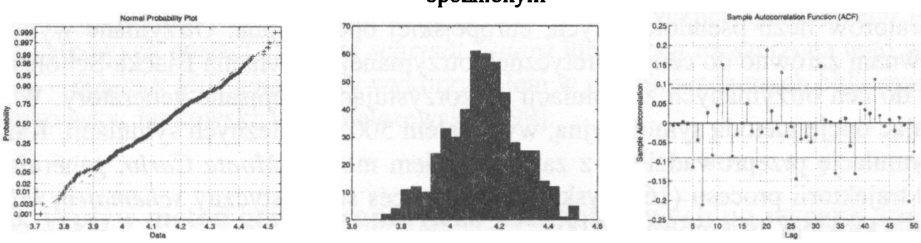
[1]Przez liczby losowe rozumiem ciąg liczb obserwowany w świecie rzeczywistym o własności takiej, iż dysponując liczbami zaobserwowanymi do tej pory, nie jesteśmy w stanie podać liczby, którą dopiero zaobserwujemy.



Rys. 1. (lewy) Wykres kwartyl-kwartyl oszacowanych cen opcji uzyskanych przy wykorzystaniu generatora MATLABa; (środkowy) Histogram cen opcji uzyskanych przy wykorzystaniu generatora MATLABa; (prawy) Wykres współczynników korelacji pomiędzy szeregiem oszacowanych cen opcji a szeregiem opóźnionym



Rys. 2. (lewy) Wykres kwartyl-kwartyl oszacowanych cen opcji uzyskanych przy wykorzystaniu generatora (18); (środkowy) Histogram cen opcji uzyskanych przy wykorzystaniu generatora (18); (prawy) Wykres współczynników korelacji pomiędzy szeregiem oszacowanych cen opcji a szeregiem opóźnionym



Rys. 3. (lewy) Wykres kwartyl-kwartyl oszacowanych cen opcji uzyskanych przy wykorzystaniu generatora (17); (środkowy) Histogram cen opcji uzyskanych przy wykorzystaniu generatora (17); (prawy) Wykres współczynników korelacji pomiędzy szeregiem oszacowanych cen opcji a szeregiem opóźnionym;

Poniżej prezentuję podstawowe statystyki oszacowanych metodą *Monte Carlo* cen opcji:

Generator	Średnia	Odchylenie Std.	Min	Max
Matlab	4,3509	0,1674	3,9131	4,9047
L'Ecuyer (18)	4,3498	0,1478	3,9350	4,8238
Marsaglia (17)	4,1349	0,1486	3,7200	4,5246

Różnica pomiędzy średnią ceną oszacowanej opcji metodą *Monte Carlo*, wykorzystującą generator Matlaba, a metodą wykorzystującą generator łączony (18) nie jest istotna statystycznie. Istotna jest różnica pomiędzy średnią ceną opcji dla generatora (17) a średnią ceną opcji dla generatora (18) oraz różnica pomiędzy ceną opcji dla generatora (17) a generatorem Matlaba. W przypadku cen opcji szereg kolejnych cen uzyskany z wykorzystaniem generatora (18) oraz generatora Matlab nie wykazuje żadnych zależności pomiędzy sobą. W przypadku generatora (17) zależności liniowe są silniejsze niż dla białego szumu. Średnia cena z wszystkich 500 symulacji dla generatora (17) jest znacznie różna od ceny teoretycznej. Generator (17) zastosowany w symulacyjnej wycenie opcji daje gorsze wyniki oszacowania (rozbieżne z ceną teoretyczną) niż generator Matlaba i generator (18). Można wysunąć wniosek, iż stosując generator (18) wykorzystany w symulacji *Monte Carlo*, jesteśmy w stanie otrzymać oszacowanie ceny opcji o niższej wariancji niż z wykorzystaniem generatora Matlaba. Biorąc pod uwagę to, iż generator Matlaba w wersji 6.5 jest bardzo szybki, można przyjąć, iż oba te generatory są porównywalne w przypadku problemu wyceny opcji.

4. Zakończenie

Niedoszacowanie ceny opcji w wyniku zastosowania w symulacji *Monte Carlo* generatora (17) pokazuje, iż nie wszystkie przetestowane generatory o „dobrych parametrach" nadają się do rozwiązywania konkretnych problemów (np. wycena opcji) metodą *Monte Carlo*. Warto przytoczyć tu słowa Donalda E. Knutha [Jäckel 2003]:

Every random number generator will fail in at least one application.

Generator łączony zaproponowany przez L'Ecuyera umożliwia otrzymanie oszacowania ceny opcji na poziomie zbliżonym do ceny otrzymanej metodą analityczną. Dodatkowo w porównaniu z generatorem Matlaba oszacowanie to cechuje się mniejsza zmiennością. Ważny aspektem, który nie jest analizowany w tym artykule, jest czas potrzebny na wykonanie symulacji z wykorzystaniem różnych generatorów liczb pseudolosowych. Ze względu na czas obliczeń generator Matlaba jest o wiele szybszy niż generator mieszany, dlatego też generator ten, będący generatorem liniowym, jest często wykorzystywany w obliczeniach symulacyjnych.

Generatory typu (17) są często wykorzystywane we współczesnych systemach komputerowych, m.in. generator tego typu jest zaimplementowany w Matlabie w wersji 5.0 [Corner 1995]. Analizy przeprowadzone na bazie generatora zaimplementowanego w *Excelu* przez Leo Knüsela wskazują, iż obliczenia symulacyjne w tym pakiecie również mogą być obarczone błędem ze względu na słabe właściwości liczb losowych. Odwołując się do pojęcia liczby losowej, należy zwrócić uwagę, iż nie ma tak naprawdę dobrych procedur arytmetycznych generowania liczb losowych. Coraz częściej naukowcy do symulacji wykorzystują ciągi liczb o pewnych właściwościach. Przykładem takich ciągów mogą być liczby Sobola czy Haltona. Zastosowanie tych liczb w symulacji często daje możliwość szybszego dojścia do prawdziwego rozwiązania niż zastosowanie liczb pseudolosowych.

Literatura

Moler C., *Random Thoughts*, www.mathworks.com, 1995.

Gentle J.E., Haerdle W., Mori Y., *Handbook of Computational Statistics*, Springer--Verlag 2004.

Glasserman P., *Monte Carlo Methods in Financial Engineering*, John Wiley & Sons, Inc. 2002.

Hull J., *Kontrakty terminowe i opcje. Wprowadzenie*, WIG-Press, Warszawa 1999.

Jankowska J., Jankowski M., *Przegląd metod i algorytmów numerycznych*, cz. 2, Wydawnictwo Naukowo-Techniczne, Warszawa 1991.

Jäckel P., *Monte Carlo Methods in Finance*, Springer-Verlag, 2003.

Knüsel L., *On the Reliability of Microsoft Excel Xp for Statistical Purposes*, www.stat.unimuenchen.de/_knuesel/elv/excelxp.pdf.

Knuth D., *The Art of Computer Programming, Volume 2 Seminumerical Algorithms*, Addison-Wesley Publishing Company, Massachusetts 1981.

L'Ecuyer P., *Good Parameter Sets for Combined Multiple Recursive Random Number Generators*, „Operations Research" 1999, nr 47(1).

Maksymiuk R., Gątarek D., *Wycena i zabezpieczenie pochodnych instrumentów finansowych*, Wydawnictwo Naukowo-Techniczne, Warszawa 1998.

Palczewski A., *Równania różniczkowe zwyczajne*, Wydawnictwo Naukowo-Techniczne, Warszawa, 1999.

Sobol I.M., *A Primer for the Monte Carlo Method*, CRC Press, Boca Raton 1994.

Zieliński R., *Metody Monte Carlo*, Wydawnictwo Naukowo-Techniczne, Warszawa 1970.

Zieliński R., *Komputerowe generowanie liczb pseudolosowych*, Wydawnictwo Naukowo-Techniczne, Warszawa 1997.

PSEUDO-RANDOM NUMBERS IN FINANCIAL OPTION VALUATION

Summary

The main purpose of his article is to show the differences between methods of generating random numbers. To test this metods for using them in process of pricing option contracts, I've used idea of control methods test to compare this algorithms of producing random numbers.

Keywords: LCGs, MRGs, CMRGs pseudo numbers generators, Fibonacci pseudo number generator, von Neuman algorithm, Monte Carlo option pricing method.