

Metody redukcji wariancji

Michał Kołodziejczyk

26 maja 2009

Spis treści

1	Przedstawienie problemu	1
2	Metody redukcji - opis teoretyczny	2
2.1	Metoda Antithetic Variates	2
2.2	Metoda Control Variates	3
2.3	Metoda Moment Matching	5
2.4	Metoda Importance Sampling	6
3	Program	7
3.1	Dane wejściowe	7
3.2	Funkcje w programie	8
3.3	Dane wyjściowe	14
4	Testy programu	14
4.1	Wyniki działania algorytmu MC bez metod redukcji wariancji	14
4.2	Wyniki działania algorytmu MC z wykorzystaniem metody antithetic variates	15
4.3	Wyniki działania algorytmu MC z wykorzystaniem metody control variates	15
4.4	Wyniki działania algorytmu MC z wykorzystaniem metody moment matching	15
4.5	Wyniki działania algorytmu MC z wykorzystaniem metody importance sampling	15

1 Przedstawienie problemu

Metody redukcji wariancji są metodami zwiększenia wydajności algorytmów Monte Carlo, poprzez zwiększenie precyzji symulacji. Załóżmy, iż chcemy wyestymować parametr θ , będący na przykład ceną instrumentu pochodnego. Załóżmy następnie, iż jesteśmy w stanie wygenerować, stosując algorytmy Monte Carlo, niezależną próbkę $\{\hat{\theta}_i, i = 1, \dots, M\}$, gdzie $\hat{\theta}_i$ ma średnią θ i wariancję σ^2 . Naturalnym estymatorem parametru θ będzie oczywiście średnia arytmetyczna:

$$\sum_{i=1}^M \hat{\theta}_i / M$$

Na mocy centralnego twierdzenia granicznego dla dużych M ta próbkowa średnia ma w przybliżeniu rozkład normalny ze średnią θ i wariancją σ^2/M . W tej metodzie błąd możemy przedstawić w postaci przedziału ufności na podstawie asymptotycznej zbieżności do rozkładu normalnego. W tej postaci błąd jest proporcjonalny do σ/\sqrt{M} .

Założmy następnie, iż mamy wybór między dwoma schematami Monte Carlo, oznaczmy je przez $\{\hat{\theta}_i^{(j)}, i = 1, \dots, M\}$ dla $j = 1, 2$. Założmy, iż obie metody są nieobciążone, tj.

$$\mathbf{E}(\hat{\theta}_i^{(1)}) = \hat{\theta}_i^{(2)} = \theta$$

natomiast wariancje dla obu metod są różne, $\sigma_1^2 < \sigma_2^2$. Wynika z tego, iż metoda 1 powinna dawać bardziej precyzyjne rezultaty niż metoda 2. Metody zmniejszania błędu w algorytmach Monte Carlo poprzez dobór schematu o jak najniższej wariancji noszą nazwę **metod redukcji wariancji**. Taka analiza w oczywisty sposób nie uwzględnia kosztu przeprowadzenia obliczeń, ten aspekt w poniższej pracy pomijamy.

2 Metody redukcji - opis teoretyczny

Omówimy cztery podstawowe metody redukcji wariancji. Są to:

- antithetic variates,
- control variates,
- moment matching,
- importance sampling.

Spróbujemy dobrać powyższe metody do kalkulacji cen opcji azjatyckich i barierowych w modelu Blacka-Scholesa, tzn. takim, w którym ceny aktywów nie płacących dywidendy spełniają stochastyczne równanie różniczkowe:

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

gdzie W_t jest procesem Wienera. Dla powyższego modelu istnieje i jest wyznaczona jednoznacznie miara neutralna wobec ryzyka, w której ceny aktywów są procesem postaci:

$$S_T = S_0 \exp\left((r - 1/2\sigma^2)T + \sigma W_T\right).$$

Dla ceny będącej procesem w takiej postaci bardzo prosto generuje się niezależne replikacje. Mając dane M niezależnych zmiennych standardowych normalnych Z_i , proces cen możemy przedstawić następująco:

$$S_T^{(i)} = S_0 \exp\left((r - 1/2\sigma^2)T + \sigma\sqrt{T}Z_i\right), \quad i = 1, \dots, M$$

Próbkowanie ze standardowego rozkładu normalnego jest względnie proste, Octave posiada implementację próbkowania przy użyciu funkcji `rnorm()`.

2.1 Metoda Antithetic Variates

Jedną z najprostszych i najszerzej stosowanych metod redukcji jest metoda **antithetic variates**. Metoda ta polega na znalezieniu dwóch ujemnie skorelowanych schematów $\{\hat{\theta}_i^{(j)}, i = 1, \dots, M\}$ dla $j = 1, 2$ o tych samych rozkładach. Wówczas wariancja dla średniej arytmetycznej z tych dwóch metod jest postaci:

$$\text{Var}\left(\frac{\hat{\theta}^{(1)} + \hat{\theta}^{(2)}}{2}\right) = \frac{1}{2}\left(\text{Var}(\hat{\theta}^{(1)}) + \text{Cov}(\hat{\theta}^{(1)}, \hat{\theta}^{(2)})\right)$$

A więc widać, iż wariancja zostaje zredukowana w przypadku, gdy:

$$\text{Var}(\hat{\theta}^{(1)}) > \text{Cov}(\hat{\theta}^{(1)}, \hat{\theta}^{(2)})$$

Tak więc w szczególności, gdy zmienne te są ujemnie skorelowane uzyskujemy poprawę zbieżności. W przypadku procesu w modelu Blacka-Scholesa uzyskanie ujemnie skorelowanych zmiennych jest relatywnie proste. Wiadomo, iż ceny te zależą jedynie od standardowego rozkładu normalnego, który jest rozkładem symetrycznym, czyli zmienna X ma taki sam rozkład, jak $-X$. Dlatego procesy:

$$S_T^{(i)} = S_0 \exp\left((r - 1/2\sigma^2)T + \sigma\sqrt{T}Z_i\right), \quad i = 1, \dots, M$$

$$\tilde{S}_T^{(i)} = S_0 \exp\left((r - 1/2\sigma^2)T + \sigma\sqrt{T}(-Z_i)\right), \quad i = 1, \dots, M$$

mają ten sam rozkład. Wykorzystując powyższą metodę do wyceny instrumentów zależnych od trajektorii algorytm MC wygląda następująco:

```
input: S_0,r,sigma,T,n,M, parametry wycenianego instrumentu
generujemy tablicę Z[M,n-1] próbek Z_i,j z rozkładu N(0,1)
tworzymy tablicę WIENER[M,n], w której zapiszemy M próbek z procesu Wienera
for i=1 to M
  for j=2 to n
    WIENER[i,j]=WIENER[i,j-1]+sqrt(T/n)*Z[i,j-1]
  end
end
generujemy tablicę S_PLUS[M,n] próbek z procesu cen przy użyciu WIENER
generujemy tablicę S_MINUS[M,n] próbek z procesu cen przy użyciu -WIENER
for i=1 to M
  liczymy cenę szukanego instrumentu PRICE_PLUS[i] dla i-tej symulacji S_PLUS
  liczymy cenę szukanego instrumentu PRICE_MINUS[i] dla i-tej symulacji S_MINUS
end
PRICE=exp(-r*T)*(mean(PRICE_PLUS)+mean(PRICE_MINUS))/2
```

Powyższa metoda działa dobrze dla zmiennych zależnych od standardowego rozkładu normalnego. Jak stosować ją dla innych zmiennych? Wiadomym jest, iż dla zmiennych X o dystrybucie F , jeśli zmienna U ma rozkład $\mathcal{U}[0, 1]$ to zmienna $F^{-1}(U)$ ma rozkład taki jak X . W tym przypadku zmienna $1 - U$ ma taki sam rozkład jak U . Daje nam to możliwość tworzenia par

$$\left(F^{-1}(U), F^{-1}(1 - U)\right)$$

zmiennych ujemnie skorelowanych o tych samych dystrybucjach F .

2.2 Metoda Control Variates

Metoda **control variates** jest najszerzej stosowaną metodą redukcji wariancji jako jedna z najprostszych i najbardziej efektywnych. Metoda ta polega na wykorzystaniu wiedzy o jednej zmiennej losowej do policzenia średniej z drugiej. Potrzebujemy zatem dwóch schematów, $\{\hat{\theta}_i^{(j)}, i = 1, \dots, M\}$ dla $j = 1, 2$. Dla pierwszego z nich szukamy średniej, a dla drugiego średnią potrafimy analitycznie policzyć. Następnie liczymy średnią ze schematu Monte Carlo

$$\{\hat{\theta}_i^{(1)} + \beta(\hat{\theta}_i^{(2)} - \theta^{(2)}), i = 1, \dots, M\}.$$

Wartości $\hat{\theta}_i^{(2)} - \theta^{(2)}$ stosowane są do kontroli $\hat{\theta}_i^{(1)}$. Wariancja dla powyższego schematu jest postaci:

$$Var\left(\hat{\theta}^{(1)} + \beta(\hat{\theta}^{(2)} - \theta^{(2)})\right) = Var(\hat{\theta}^{(1)}) + \beta^2 Var(\hat{\theta}^{(2)}) + 2\beta Cov(\hat{\theta}^{(1)}, \hat{\theta}^{(2)})$$

Tak więc widać wprost, iż metoda ta daje dobre rezultaty w przypadku, gdy:

$$\beta^2 Var(\hat{\theta}^{(2)}) + 2\beta Cov(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}) < 0$$

a więc zależy od zmiennej kontrolującej $\hat{\theta}^{(2)}$ i od parametru β . Zależność od parametru β jest zależnością kwadratową, zatem chcąc zredukować wariancję optymalnie, należy dobrać β minimalizującą funkcję kwadratową. Jest to:

$$\beta^* = \frac{-Cov(\hat{\theta}^{(1)}, \hat{\theta}^{(2)})}{Var(\hat{\theta}^{(2)})}.$$

Wstawiając do wzoru na wariancję optymalny parametr β^* otrzymujemy

$$Var\left(\hat{\theta}^{(1)} + \beta(\hat{\theta}^{(2)} - \theta^{(2)})\right) = (1 - \rho_{\hat{\theta}^{(1)}, \hat{\theta}^{(2)}}^2) Var(\hat{\theta}^{(1)})$$

tak więc metoda ta działa najlepiej, gdy jesteśmy w stanie dobrać zmienną kontrolującą dobrze skorelowaną ze zmienną, dla której szukamy średniej.

W przypadku opcji azjatyckich zmienna kontrolująca w modelu Blacka-Scholesa zaproponowana w implementacji, to geometryczna opcja azjatycka. Dla takiej opcji (call) analityczne wzory wyglądają następująco:

$$\theta^{(2)} = S_0 \exp\left((a - r)T\right) N(d_1) - K \exp\left(-rT\right) N(d_2)$$

gdzie

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(a + \frac{\sigma_a^2}{2}\right)T}{\sigma_a \sqrt{T}}, \quad d_2 = d_1 - \sigma_a \sqrt{T}, \quad a = \frac{r - \sigma^2/6}{2}, \quad \sigma_a = \sigma/\sqrt{3}$$

Dla opcji barierowych natomiast w implementacji stosowano odpowiadające im opcje bez barier korzystając ze wzorów Blacka-Scholesa na ceny opcji (call) europejskich:

$$\theta^{(2)} = S_0 N(d_1) - K \exp\left(-rT\right) N(d_2)$$

gdzie

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma \sqrt{T}}, \quad d_2 = d_1 - \sigma \sqrt{T}$$

Ogólnie, dla cen instrumentów w modelu Blacka-Scholesa, korzystając z faktu, iż aktywa S_t są zdyskontowanymi martyngałami, za zmienną kontrolującą możemy brać właśnie cenę na instrument S_t . Wówczas algorytm MC wyceny instrumentów z zastosowaniem zmiennej kontrolującej S_T wygląda następująco:

```
input: S_0,r,sigma,T,n,M, parametry wycenianego instrumentu
generujemy tablicę Z[M,n-1] próbek Z_i,j z rozkładu N(0,1)
tworzymy tablicę WIENER[M,n], w której zapiszemy M próbek z procesu Wienera
for i=1 to M
  for j=2 to n
    WIENER[i,j]=WIENER[i,j-1]+sqrt(T/n)*Z[i,j-1]
  end
end
generujemy tablicę S[M,n] próbek z procesu cen przy użyciu WIENER
for i=1 to M
  liczymy cenę szukanego instrumentu PRICE[i] dla i-tej symulacji S
```

```

end
liczymy parametr BETA z próbkowych estymatorów wariancji i kowariancji
for i=1 to M
    liczymy PRICE_CONTROL[i]=PRICE[i]+BETA*(S[i,n]-exp(r*T)*S_0)
end
PRICE=exp(-r*T)*(mean(PRICE_CONTROL))

```

Metoda ta jak widać jest bardzo prosta do implementacji, wymaga jedynie znajomości średniej dla procesu skorelowanego z szukanym.

2.3 Metoda Moment Matching

Kolejna metoda zwana **moment matching** lub też inaczej **quadratic sampling** polega na dopasowywaniu średnich próbkowych do średnich populacyjnych. Metoda ta jest kolejną po control variates metodą polegającą na wykorzystaniu wiedzy o jednej zmiennej losowej do policzenia średniej z drugiej. Potrzebujemy zatem znowu dwóch schematów Monte Carlo $\{\hat{\theta}_i^{(j)}, i = 1, \dots, M\}$ dla $j = 1, 2$, przy czym zachodzi między nimi zależność funkcyjna:

$$\hat{\theta}^{(1)} = f(\hat{\theta}^{(2)}),$$

gdzie funkcja f jest całkowalna. Dla schematu pierwszego szukamy średniej a w schemacie drugim znamy średnią populacyjną. Wówczas do policzenia średniej próbkowej z $\hat{\theta}^{(1)}$ stosujemy

$$f\left(\hat{\theta}_i^{(2)} - \sum_{k=1}^M \hat{\theta}_k^{(2)}/M + \theta^{(2)}\right),$$

czyli dopasowujemy pierwszy moment próbkowy do momentu populacyjnego. Znając kolejne momenty populacyjne możemy dopasowywać je do próbkowych. Jeśli znana jest wariancja $\hat{\theta}^{(2)}$, to oznaczając przez $s_{\hat{\theta}^{(2)}}$ odchylenie standardowe próbkowe i przez $\sigma_{\hat{\theta}^{(2)}}$ odchylenie standardowe populacyjne, dopasowanie do drugiego momentu otrzymujemy następująco:

$$\left(\hat{\theta}_i^{(2)} - \sum_{k=1}^M \hat{\theta}_k^{(2)}/M\right) \frac{\sigma_{\hat{\theta}^{(2)}}}{s_{\hat{\theta}^{(2)}}} + \theta^{(2)}$$

W przypadku modelu Blacka-Scholesa w ogólnym przypadku można zastosować tą metodę albo do zmiennych standardowych normalnych użytych do wygenerowania trajektorii procesu Wienera, albo do ceny instrumentu bazowego. Drugie podejście zostało zastosowane w implementacji. Wówczas za proces cen instrumentu bazowego do kalkulacji ceny bierzemy:

$$\tilde{S}_T(i) = S_T(i) - \sum_{k=1}^M S_T(k)/M + \exp(rT)S_0$$

co dopasowuje pierwsze momenty. Do dopasowania drugich momentów używamy procesu postaci:

$$\tilde{S}_T(i) = \left(S_T(i) - \sum_{k=1}^M S_T(k)/M\right) \frac{\sigma_{S_T}}{s_{S_T}} + \exp(rT)S_0$$

Opiszemy teraz ogólny algorytm MC do liczenia cen instrumentów zależnych od trajektorii w modelu Blacka-Scholesa z wykorzystaniem metody moment matching:

```

input: S_0,r,sigma,T,n,M, parametry wycenianego instrumentu
generujemy tablicę Z[M,n-1] próbek Z_i,j z rozkładu N(0,1)
tworzymy tablicę WIENER[M,n], w której zapiszemy M próbek z procesu Wienera
for i=1 to M
  for j=2 to n
    WIENER[i,j]=WIENER[i,j-1]+sqrt(T/n)*Z[i,j-1]
  end
end
generujemy tablicę S_MM[M,n] próbek z procesu cen przy użyciu WIENER i metody MM
for i=1 to M
  liczymy cenę szukanego instrumentu PRICE_MM[i] dla i-tej symulacji S
end
PRICE=exp(-r*T)*(mean(PRICE_MM))

```

Opisana powyżej metoda wymaga tak samo jak w przypadku metody control variates znajomości średniej dla innej zmiennej. Jednak w takim przypadku użycie zmiennej jako kontrolnej zamiast do dopasowania momentów daje lepsze rezultaty, dlatego nie jest ona często używana. Kolejny problem, który się pojawia przy okazji tej metody, to utrata niezależności między symulacjami cen. Powoduje to dodatkowe utrudnienie - wariancję próbkową trzeba liczyć z niezależnych przebiegów.

2.4 Metoda Importance Sampling

Ostatnią opisywaną tu metodą jest metoda **importance sampling**. Wykorzystuje ona fakt, iż średnia liczona przy jednej mierze może być wyrażona jako średnia liczona przy innej mierze korzystając z pochodnej Radona-Nikodyma. Czym jest pochodna Radona-Nikodyma? Załóżmy, iż mamy miarę \mathbf{Q} . Wówczas jeśli miara \mathbf{P} jest absolutnie ciągła względem miary \mathbf{Q} , to istnieje zmienna losowa L całkowalna względem miary \mathbf{Q} taka, że:

$$\mathbf{E}_{\mathbf{P}}X = \mathbf{E}_{\mathbf{Q}}LX \quad \forall X \in \mathcal{L}_1(\Omega, \mathbf{Q}, \mathcal{F})$$

Głównym problemem w tej metodzie jest dobranie miary w ten sposób, by ograniczyć wariancję. Metoda ta daje dobre rezultaty w przypadkach, w których potrafimy dobrać miarę w ten sposób, by zdarzenia rzadkie występowały częściej. Tak więc potrzebujemy schematu Monte Carlo $\{L\hat{\theta}_i, i = 1, \dots, M\}$ względem miary \mathbf{Q} absolutnie ciągłej względem \mathbf{P} , gdzie zmienna $\hat{\theta}$ w mierze \mathbf{P} ma szukaną średnią θ , a $L = \frac{d\mathbf{P}}{d\mathbf{Q}}$.

W przypadku modelu Blacka-Scholesa możemy w bardzo prosty sposób manewrować dryfem. Pamiętajmy przecież, iż różnica między miarą fizyczną a martyngałową to różnica w dryfie. W tym przypadku możemy zastosować miarę, względem której proces cen ma dryf μ . Wówczas taka miara jest absolutnie ciągła względem miary martyngałowej, a pochodna Radona-Nikodyma wynosi:

$$L = \left(\frac{S_T}{S_0}\right)^{(r-\mu)/\sigma^2} \exp\left(\frac{(\mu^2 - r^2)T}{2\sigma^2}\right)$$

na mocy twierdzenia Girsanowa. Wówczas algorytm MC do liczenia cen instrumentów w modelu Blacka-Scholesa z wykorzystaniem metody importance sampling wygląda następująco:

```

input: S_0,r,sigma,T,n,M,mu, parametry wycenianego instrumentu
generujemy tablicę Z[M,n-1] próbek Z_i,j z rozkładu N(0,1)
tworzymy tablicę WIENER[M,n], w której zapiszemy M próbek z procesu Wienera
for i=1 to M
  for j=2 to n

```

```

        WIENER[i,j]=WIENER[i,j-1]+sqrt(T/n)*Z[i,j-1]
    end
end
generujemy tablicę S_MU[M,n] próbek z procesu cen przy użyciu WIENER i dryfu MU
for i=1 to M
    liczymy cenę szukanego instrumentu PRICE_MU[i] dla i-tej symulacji S
end
PRICE=exp(-r*T)*(mean(L*PRICE_MU))

```

3 Program

Program przedstawia przykładową implementację powyższych metod redukcji wariancji do liczenia cen opcji azjatyckiej i opcji barierowych. Składa się on z następujących plików:

- dane_gen.m - plik do generowania danych wykorzystywanych przez program
- dane.txt - dane wygenerowane przez plik dane_gen.m
- rw.m - funkcje liczące ceny opcji bez metod redukcji
- rw_antithetic.m - funkcje liczące ceny opcji z wykorzystaniem metody antithetic variates
- rw_control.m - funkcje liczące ceny opcji z wykorzystaniem metody control variates
- rw_mm.m - funkcje liczące ceny opcji z wykorzystaniem metody moment matching
- rw_is.m - funkcje liczące ceny opcji z wykorzystaniem metody importance sampling
- start.m - program uruchomieniowy

W tej części opiszemy metodę generowania i wprowadzania danych do programu, poszczególne funkcje programu, oraz dane wyjściowe programu.

3.1 Dane wejściowe

Do generowania danych do programu służy plik dane_gen.m. W pliku tym znajdują się następujące parametry:

- s - cena instrumentu podstawowego w chwili 0
- r - stopa procentowa ciągła pozbawiona ryzyka
- sigma - zmienność cen instrumentu podstawowego
- T - czas do zapadalności opcji
- K - cena wykonania opcji (w przypadku opcji call należy wprowadzić ją ze znakiem minus)
- M - ilość symulacji w metodzie Monte Carlo
- n - liczba węzłów, w których liczymy wartość instrumentu podstawowego
- alpha - parametr wyboru typu opcji (1 - call, -1 - put)

- `barrier_1` - bariera górna dla opcji barierowej
- `barrier_2` - bariera dolna dla opcji barierowej

Po uruchomieniu powyższego pliku w Octave zostanie wygenerowany plik `dane.txt`. Program właśnie z tego miejsca pobiera dane przy inicjalizacji programu. Plik ten musi znajdować się w tym samym folderze, co uruchamiany program.

3.2 Funkcje w programie

Plik `rw.m` zawiera następujące funkcje:

`asian_priceMC(alpha,s,r,sigma,T,K,M,n)`

Funkcja ta zawiera standardowy algorytm Monte Carlo liczący cenę opcji azjatyckiej dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `asian_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_up_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera standardowy algorytm Monte Carlo liczący cenę opcji barierowej up and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_in_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_up_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera standardowy algorytm Monte Carlo liczący cenę opcji barierowej up and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_out_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_down_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera standardowy algorytm Monte Carlo liczący cenę opcji barierowej down and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_in_option` - wartość opcji na podstawie algorytmu MC

- lewy - lewy kraniec 95% przedziału ufności
- prawy - prawy kraniec 95% przedziału ufności

`barrier_down_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera standardowy algorytm Monte Carlo liczący cenę opcji barierowej down and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_out_option` - wartość opcji na podstawie algorytmu MC
- lewy - lewy kraniec 95% przedziału ufności
- prawy - prawy kraniec 95% przedziału ufności

Plik `rw_antithetic.m` zawiera następujące funkcje:

`asian_priceMC(alpha,s,r,sigma,T,K,M,n)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody antithetic variates liczący cenę opcji azjatyckiej dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `asian_option` - wartość opcji na podstawie algorytmu MC
- lewy - lewy kraniec 95% przedziału ufności
- prawy - prawy kraniec 95% przedziału ufności

`barrier_up_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody antithetic variates liczący cenę opcji barierowej up and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_in_option` - wartość opcji na podstawie algorytmu MC
- lewy - lewy kraniec 95% przedziału ufności
- prawy - prawy kraniec 95% przedziału ufności

`barrier_up_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody antithetic variates liczący cenę opcji barierowej up and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_out_option` - wartość opcji na podstawie algorytmu MC
- lewy - lewy kraniec 95% przedziału ufności
- prawy - prawy kraniec 95% przedziału ufności

`barrier_down_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody antithetic variates liczący cenę opcji barierowej down and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_in_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_down_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody antithetic variates liczący cenę opcji barierowej down and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_out_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

Plik `rw_control.m` zawiera następujące funkcje:

`asian_geom_price_bs(alpha,s,r,sigma,T,K)`

Funkcja ta liczy cenę geometrycznej opcji azjatyckiej ze wzoru analitycznego dla podanych parametrów. Funkcja ta zwraca wartość:

- `asian_geom_bs` - wartość opcji na podstawie wzoru analitycznego

`eur_price_bs(alpha,s,r,sigma,T,K)`

Funkcja ta liczy cenę opcji europejskiej ze wzoru analitycznego dla podanych parametrów. Funkcja ta zwraca wartość:

- `eur_bs` - wartość opcji na podstawie wzoru analitycznego

`asian_priceMC(alpha,s,r,sigma,T,K,M,n)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody control variates liczący cenę opcji azjatyckiej dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `asian_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_up_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody control variates liczący cenę opcji barierowej up and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_in_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_up_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody control variates liczący cenę opcji barierowej up and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_out_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_down_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody control variates liczący cenę opcji barierowej down and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_in_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_down_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody control variates liczący cenę opcji barierowej down and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_out_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

Plik `rw_mm.m` zawiera następujące funkcje:

`asian_priceMC(alpha,s,r,sigma,T,K,M,n)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody moment matching liczący cenę opcji azjatyckiej dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `asian_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_up_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody moment matching liczący cenę opcji barierowej up and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_in_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_up_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody moment matching liczący cenę opcji barierowej up and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_out_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_down_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody moment matching liczący cenę opcji barierowej down and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_in_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

`barrier_down_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)`

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody moment matching liczący cenę opcji barierowej down and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_out_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

Plik `rw_is.m` zawiera następujące funkcje:

```
asian_priceMC(alpha,s,r,sigma,T,K,M,n)
```

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody importance sampling liczący cenę opcji azjatyckiej dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `asian_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

```
barrier_up_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)
```

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody importance sampling liczący cenę opcji barierowej up and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_in_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

```
barrier_up_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)
```

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody importance sampling liczący cenę opcji barierowej up and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_up_out_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

```
barrier_down_in_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)
```

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody importance sampling liczący cenę opcji barierowej down and in dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_in_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

```
barrier_down_out_priceMC(alpha,s,r,sigma,T,K,M,n,barrier)
```

Funkcja ta zawiera algorytm Monte Carlo z wykorzystaniem metody importance sampling liczący cenę opcji barierowej down and out dla podanych parametrów. Funkcja ta zwraca wektor o długości 3 zawierający następujące wartości:

- `barrier_down_out_option` - wartość opcji na podstawie algorytmu MC
- `lewy` - lewy kraniec 95% przedziału ufności
- `prawy` - prawy kraniec 95% przedziału ufności

3.3 Dane wyjściowe

Kod służący do uruchomienia programu znajduje się w pliku `start.m`. Uruchomić go można w konsoli Octave poprzez wprowadzenie polecenia `start`. Aby program został uruchomiony wszystkie pliki muszą znajdować się w katalogu głównym Octave. Po uruchomieniu programu użytkownik zostanie poproszony o podanie typu opcji, której wartość chce policzyć:

Podaj typ opcji, ktora chcesz policzyc:

- 1 - azjatycka
- 2 - barierowa up and in
- 3 - barierowa up and out
- 4 - barierowa down and in
- 5 - barierowa down and out

Następnie użytkownik zostanie poproszony o podanie typu metody redukcji wariancji, którą chce zastosować:

Podaj metode redukcji wariancji, jaka chcesz zastosowac:

- 1 - brak
- 2 - anttithetic variates
- 3 - control variates
- 4 - moment matching
- 5 - importance sampling

Wszystkie parametry opcji należy wygenerować przy użyciu pliku `dane_gen.m`. Program jako wynik działania podaje cenę szukanej opcji, oraz 95% przedział ufności.

4 Testy programu

W tej części zostaną przedstawione przykładowe wyniki uzyskane przy użyciu opisanego powyżej programu.

4.1 Wyniki działania algorytmu MC bez metod redukcji wariancji

Startując z danych: ceny początkowej $S_0 = 50$, ceny wykonania $K = 50$, stopy bezryzykowej $r = 0.05$, zmienności $\sigma = 0.2$, czasu życia $T = 0.5$, liczby węzłów $n = 200$, liczby symulacji $M = 100000$ metoda MC bez redukcji wariancji daje następujące rezultaty:

Typ opcji	Cena opcji	Przedział ufności
Call azjatycka	1.9244	(1.9072, 1.9416)
Call barierowa up and in z barierą 60	2.2705	(2.2389, 2.3020)
Call barierowa up and out z barierą 60	1.1992	(1.1853, 1.2130)
Call barierowa down and in z barierą 45	0.1947	(0.1880, 0.2014)
Call barierowa down and out z barierą 45	3.2410	(3.2100, 3.2720)

4.2 Wyniki działania algorytmu MC z wykorzystaniem metody anti-thetic variates

Startując z danych: ceny początkowej $S_0 = 50$, ceny wykonania $K = 50$, stopy bezryzykowej $r = 0.05$, zmienności $\sigma = 0.2$, czasu życia $T = 0.5$, liczby węzłów $n = 200$, liczby symulacji $M = 100000$ metoda MC z redukcją metodą antithetic variates daje następujące rezultaty:

Typ opcji	Cena opcji	Przedział ufności
Call azjatycka	1.9210	(1.9125, 1.9296)
Call barierowa up and in z barierą 60	2.2386	(2.2189, 2.2584)
Call barierowa up and out z barierą 60	1.1972	(1.1890, 1.2054)
Call barierowa down and in z barierą 45	0.1938	(0.1892, 0.1985)
Call barierowa down and out z barierą 45	3.2572	(3.2406, 3.2737)

4.3 Wyniki działania algorytmu MC z wykorzystaniem metody control variates

Startując z danych: ceny początkowej $S_0 = 50$, ceny wykonania $K = 50$, stopy bezryzykowej $r = 0.05$, zmienności $\sigma = 0.2$, czasu życia $T = 0.5$, liczby węzłów $n = 200$, liczby symulacji $M = 100000$ metoda MC z redukcją metodą control variates daje następujące rezultaty:

Typ opcji	Cena opcji	Przedział ufności
Call azjatycka	1.9270	(1.9266, 1.9273)
Call barierowa up and in z barierą 60	2.2601	(2.2470, 2.2733)
Call barierowa up and out z barierą 60	1.1855	(1.1722, 1.1987)
Call barierowa down and in z barierą 45	0.1886	(0.1822, 0.1951)
Call barierowa down and out z barierą 45	3.2534	(3.2470, 3.2598)

4.4 Wyniki działania algorytmu MC z wykorzystaniem metody moment matching

Startując z danych: ceny początkowej $S_0 = 50$, ceny wykonania $K = 50$, stopy bezryzykowej $r = 0.05$, zmienności $\sigma = 0.2$, czasu życia $T = 0.5$, liczby węzłów $n = 200$, liczby symulacji $M = 100000$ metoda MC z redukcją metodą moment matching daje następujące rezultaty:

Typ opcji	Cena opcji	Przedział ufności
Call azjatycka	1.9178	(1.9007, 1.9349)
Call barierowa up and in z barierą 60	2.2552	(2.2236, 2.2868)
Call barierowa up and out z barierą 60	1.1996	(1.1857, 1.2134)
Call barierowa down and in z barierą 45	0.1883	(0.1817, 0.1949)
Call barierowa down and out z barierą 45	3.2481	(3.2169, 3.2793)

4.5 Wyniki działania algorytmu MC z wykorzystaniem metody importance sampling

Startując z danych: ceny początkowej $S_0 = 50$, ceny wykonania $K = 50$, stopy bezryzykowej $r = 0.05$, zmienności $\sigma = 0.2$, czasu życia $T = 0.5$, liczby węzłów $n = 200$, liczby symulacji $M = 100000$ metoda MC z redukcją metodą importance sampling daje następujące rezultaty:

Typ opcji	Cena opcji	Przedział ufności
Call azjatycka	1.9475	(1.9091, 1.9858)
Call barierowa up and in z barierą 60	2.2608	(2.1769, 2.3448)
Call barierowa up and out z barierą 60	1.1938	(1.1790, 1.2087)
Call barierowa down and in z barierą 45	0.1921	(0.1823, 0.2019)
Call barierowa down and out z barierą 45	3.2734	(3.1945, 3.3523)