

## **Dokumentacja**

### **Wycena opcji amerykańskich metodą Longstaffa-Schwartza**

Dorota Toczydłowska  
Piotr Kosewski  
Katarzyna Mioduszevska  
Piotr Garbuliński  
Bartosz Głowinkowski  
Katarzyna Kańska  
Sonia Stroka  
Kinga Trychta  
Konrad Czechowski  
dodatki AP

## Spis treści

<b>1</b>	<b>Wstęp teoretyczny</b>	<b>3</b>
1.1	Model rynku . . . . .	3
1.2	Wycena opcji metodą Monte Carlo . . . . .	4
1.3	Wycena opcji amerykańskich . . . . .	4
1.4	Metoda Longstaffa-Schwartza . . . . .	5
1.5	Opis algorytmu . . . . .	5
1.6	Total Least Squares . . . . .	8
1.7	Estymator nieobciążony . . . . .	9
<b>2</b>	<b>Wyznaczenie parametrów greckich</b>	<b>9</b>
<b>3</b>	<b>Opcje barierowe (barrier options)</b>	<b>13</b>
3.1	Wycena opcji z barierą obserwowaną w sposób dyskretny, modyfikacja dla bariery okienkowej . . . . .	14
3.2	Wycena opcji z barierą obserwowaną w sposób ciągły . . . . .	15
<b>4</b>	<b>Wyznaczanie parametrów greckich dla barierowych opcji europejskich</b>	<b>17</b>
4.1	Metody wyznaczania parametrów greckich . . . . .	17
4.2	Rachunek Malliavina . . . . .	18
4.3	Wyładzanie barriery . . . . .	20
<b>5</b>	<b>Opis produktu</b>	<b>23</b>
<b>6</b>	<b>Dokumentacja</b>	<b>24</b>
6.1	Opisy funkcji . . . . .	24
6.1.1	brownianBridge . . . . .	24
6.1.2	generateAssetBase_barrier . . . . .	24
6.1.3	generateAssetBase_barrier_eur . . . . .	25
6.1.4	updateStopping_time . . . . .	25
6.1.5	updateStopping_time_eur . . . . .	25
6.1.6	EuropeanOptionPrice . . . . .	26
6.1.7	EuropeanOptionPrice_ . . . . .	26
6.1.8	LSMmodel . . . . .	27
6.1.9	CheckIfActive . . . . .	27
6.1.10	CheckIfHappened . . . . .	27
6.1.11	rozszerzony ExpectedValueCoef . . . . .	27
6.1.12	rozszerzony CalculateExpectedValue . . . . .	28
6.1.13	LaguerrePolyMatrix . . . . .	28
6.1.14	EurGreeks . . . . .	28
6.1.15	dXcalculate . . . . .	28
6.1.16	Psi . . . . .	29
6.1.17	CalculateMC . . . . .	29
6.1.18	CalculateMC_ . . . . .	29
6.1.19	log_two_cosh . . . . .	29

6.1.20	generateSmoothFunc_all . . . . .	30
6.1.21	smoothed_price_and_greeks . . . . .	30
6.1.22	smoothed_price_and_delta . . . . .	30
6.1.23	smoothed_gamma . . . . .	30
6.1.24	smoothed_price_and_greeks_ . . . . .	30
6.1.25	barrier_opt_price_and_greeks_from_asset_double_out	31
6.1.26	barrier_opt_price_and_greeks_from_asset . .	31

**Literatura**

**31**

## Opis projektu

Zadaniem projektu jest opracowanie oprogramowania do wyceny szerokiej klasy obiektów na rynku FX oraz Equity metodą Monte Carlo. Kod został przystosowany do korzystania z danych rynkowych i rzeczywistych dat związanych z życiem opcji (data rozpoczęcia, data wygaśnięcia, daty, w których monitorowana jest bariera). Opracowana została wycena szerokiej klasy opcji, w tym opcji barierowych: zarówno europejskich jak i amerykańskich. Przy wycenie tych drugich wykorzystana została metoda Longstaffa-Schwartza. Projekt jest efektem pracy uczestników seminarium w różnych latach. Początkowo został stworzony kod wyceniający opcje waniliowe europejskie i amerykańskie metodą Monte Carlo oraz opcje z barierą monitorowaną jedynie w momencie wygaśnięcia. Ten projekt został poszerzony o możliwości wyceny o kontrakty, w których bariera jest monitorowana podczas całego życia opcji lub w konkretnych momentach - a nie jedynie w momencie wygaśnięcia. W projekcie zostały zaimplementowane dwie możliwości wyboru współczynników regresji w modelu Longstaffa - Schwartza: standardowa metoda OLS (Ordinary Least Squares), metoda TLS (Total Least Squares) oraz nieobciążony estymator wartości oczekiwanej. Metoda Longstaffa-Schwartza została zmodyfikowana i pozwala na stworzenie efektywnej implementacji wyceny opcji przy ograniczonej złożoności pamięciowej, co umożliwia przeprowadzenie symulacji z dużą liczbą ścieżek i punktów czasowych.

Na końcu pracy zaprezentowane są wyniki porównania różnych metod liczenia ceny i sformułowane wnioski.

## 1 Wstęp teoretyczny

### 1.1 Model rynku

Niech  $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \in [0, T]}, \mathbb{P})$  będzie przestrzenią probabilistyczną, na której mamy zadany proces Wienera  $W$ .  $\sigma$ -ciało  $\mathcal{F}_t$  interpretujemy jako wiedzę uzyskaną do chwili  $t$ . Wiedzę w naszym modelu czerpiemy jedynie z procesu cen akcji, a zatem  $\mathcal{F} = \mathcal{F}^S$ . Powszechnie przyjmuje się, że ceny akcji mogą być modelowane przez następujące stochastyczne równanie różniczkowe (SDE):

$$dS_t = rS_t dt + \sigma dW_t. \quad (1)$$

- $S$  - wartość aktywa
- $r$  - stopa procentowa zależna od czasu
- $\sigma$  - zmienność instrumentu
- $W_t$  - proces Wienera

Stąd też  $\mathcal{F}^S = \mathcal{F}^W$ , zatem zakładamy, że filtracja jest uzupełnioną filtracją procesu Wienera  $W$ .

## 1.2 Wycena opcji metodą Monte Carlo

Metoda Monte Carlo jest jedną z najważniejszych metod (obok drzewa dwumianowego i równań różniczkowych cząstkowych) stosowanych przy algorytmach numerycznych wyceny opcji. Dla prostoty, skupimy się na opcjach europejskich. Model Blacka-Scholesa zakłada, że zwroty z akcji mają rozkład zgodny z przedstawionym wyżej stochastycznym równaniem różniczkowym. Zgodnie z założeniem o braku arbitrażu, muszą być one równoważne średniej wypłacie z opcji, zdyskontowanej do chwili obecnej za pomocą stopy bezryzykowej  $r$ . Przez symulacje Monte Carlo rozumiemy proces, w którym generujemy dużą liczbę ścieżek, które zachowują się zgodnie z dynamiką zadaną przez (1). Następnie obliczamy wypłatę z opcji dla każdej ze ścieżek, po czym bierzemy średnią arytmetyczną owych wypłat, aby otrzymać sprawiedliwą wartość opcji.

W bardziej ścisłym języku: w modelu dyskretnym uczciwa cena opcji to

$$\mathbb{E}_{\mathbb{Q}} X B_T^{-1},$$

gdzie  $\mathbb{Q}$  jest miarą martyngałową na rynku, równoważną wyjściowej mierze  $\mathbb{P}$ ,  $X$  jest wypłatą z opcji, a  $T$  jest momentem zapadalności. Korzystamy z Prawa Wielkich Liczb w następujący sposób: jeśli  $X_1, X_2, \dots, X_M$  są próbkami z danego rozkładu, to dla dostatecznie dużych  $M$ ,  $\frac{1}{M} \sum_{i=1}^M X_i$  jest dobrym estymatorem  $\mathbb{E}X$ . Teraz korzystając z tego, że cena ma rozkład lognormalny, możemy zasymulować  $M$  ścieżek i uśrednić wyniki na tych ścieżkach, otrzymując dobre przybliżenie zdyskontowanej wartości oczekiwanej wypłaty.

Dla opcji amerykańskich sprawa nieco się komplikuje, ponieważ z założenia o racjonalności inwestora, wybiera on moment, gdy wartość oczekiwana jego wypłaty jest największa. Algorytm wyliczenia ceny opcji jest zatem bardziej skomplikowany i opiszemy go później.

## 1.3 Wycena opcji amerykańskich

Wycena ciągłej opcji amerykańskiej może być sformułowana przez zdefiniowanie procesu  $U_t$  dla  $t \in [0, T]$ , będącego procesem całkownym z kwadratem, adaptowanym do filtracji  $(\mathcal{F})_{t \in [0, T]}$ , który określa zdyskontowaną wypłatę z opcji z momentu  $t$  postaci  $U(t) = h(X_t)$ , gdzie  $h$  jest funkcją borelowską.

Zdefiniujemy dodatkowo klasę momentów stopu  $\tau \in \mathcal{T}_{t, T}$  o wartościach z przedziału  $[t, T]$ . Niech  $T$  będzie momentem wygaśnięcia opcji. Przez wartość opcji w chwili 0 oznaczamy

$$V_0 = \sup_{\tau \in \mathcal{T}_{0, T}} \mathbb{E}U_{\tau}$$

Przez wartość opcji w chwili  $t$  oznaczamy

$$V_t = \sup_{\tau \in \mathcal{T}_{t, T}} \mathbb{E}U_{\tau}$$

Aby podjąć się wyceny opcji, w dalszych rozważaniach musimy uciec się do dyskretyzacji rozważanej przestrzeni  $\Omega$ . Zakładamy skończony horyzont czasowy  $[0, T]$ , gdzie przestrzeń stanów  $\Omega$  jest zbiorem wszystkich możliwych stanów rynku pomiędzy 0 i  $T$ . Zakładamy, że  $\omega \in \Omega$  reprezentuje możliwą trajektorię procesu cen. Aby

dostatecznie dobrze przybliżać wartości opcji, musimy ustalić dostatecznie gęstą siatkę momentów  $t_1, t_2, \dots, t_M$ . Korzystając z założenia o braku arbitrażu, zakładamy, że istnieje równoważna z wyjściową miara martyngałowa  $\mathbb{Q}$  dla tego rynku. Można udowodnić, że wartość opcji amerykańskiej może być przedstawiona w postaci obwiedni Snella, jak poniżej.

Formułujemy teraz problem wyceny opcji amerykańskiej jako problem programowania dynamicznego, przyjmując, że:

$$\begin{aligned} V_T &= U_T \\ V_t &= \max(U_t, \mathbb{E}_{\mathbb{Q}}(V_{t+1}|\mathcal{F}_t)) \end{aligned}$$

## 1.4 Metoda Longstaffa-Schwartz

Podstawą metody Longstaffa-Schwartz jest zasada programowania dynamicznego

$$V_T = U_T$$

$$V_t = \max(U_t, \mathbb{E}(V_{t+1}|\mathcal{F}_t)) \quad (2)$$

W języku momentów stopu  $\tau_t = \{k \geq t | U_k = V_k\}$  zasadę tę możemy zapisać jako

$$\begin{aligned} \tau_T &= T, \\ \tau_t &= t \mathbf{1}_{U_t \geq \mathbb{E}(U_{\tau_{t+1}}|\mathcal{F}_t)} + \tau_{t+1} \mathbf{1}_{U_t < \mathbb{E}(U_{\tau_{t+1}}|\mathcal{F}_t)} \quad \text{dla } t \in [0, T-1] \end{aligned} \quad (3)$$

Między innymi z własności procesu Markowa procesu  $X_t$  mamy równość (dowód [11])

$$\mathbb{E}(U_{\tau_{t+1}}|\mathcal{F}_t) = \mathbb{E}(U_{\tau_{t+1}}|X_t)$$

Wówczas

$$\begin{aligned} \tau_T &= T, \\ \tau_t &= t \mathbf{1}_{U_t \geq \mathbb{E}(U_{\tau_{t+1}}|X_t)} + \tau_{t+1} \mathbf{1}_{U_t < \mathbb{E}(U_{\tau_{t+1}}|X_t)} \quad \text{dla } t \in [0, T-1] \end{aligned} \quad (4)$$

W praktyce używa się ścieżek *in - the - money*, co ma swoje uzasadnienie w większej efektywności obliczeniowej algorytmu. W takim wypadku, zdefiniowane wyżej momenty stopu mają postać

$$\begin{aligned} \bar{\tau}_T &= T, \\ \bar{\tau}_t &= t \mathbf{1}_{\{U_t \geq \mathbb{E}(U_{\tau_{t+1}}|X_t)\} \cap \{U_t > 0\}} + \bar{\tau}_{t+1} \mathbf{1}_{\{U_t < \mathbb{E}(U_{\tau_{t+1}}|X_t)\} \cup \{U_t > 0\}} \quad \text{dla } t \in [0, T-1]. \end{aligned} \quad (5)$$

## 1.5 Opis algorytmu

### 1. Krok 1.

Pierwszym krokiem konstrukcji ceny opcji metody Longstaffa-Schwartz jest rzut ortogonalny procesu  $X_t$  na przestrzeń funkcji mierzalnych od  $X_t$ , gdzie  $e_k : E \rightarrow R$  jest ciągiem tych funkcji spełniających warunki dla  $t \in \{0, 1, \dots, T-1\}$

- $e_k(X_t)$  jest zupełny w  $L^2(\sigma(X_t))$ ,
- dla  $m \geq 1$   $\sum_{k=1}^m \beta_{t,k} e_k(X_t) = 0$  p.n.  $\Rightarrow \beta_{t,k} = 0$  dla  $k \in \{1, \dots, m\}$ .

Główną ideą metody Longstaffa - Schwartza jest aproksymowanie wartości oczekiwanej z (2) używając ciągu  $m + 1$  funkcji  $e_k$  w każdym z kroków  $t$

$$\mathbb{E}(V_{t+1}|\mathcal{F}_t) = \mathbb{E}(V_{t+1}|X_t) = \sum_{k=0}^m \beta_{t,k} e_k(X_t) \quad (6)$$

Postulowaną w pracy Longstaffa-Schwartza rodziną funkcji spełniającą powyższe warunki, są ważone wielomiany Laguerre'a. Ich postać to

$$\begin{aligned} e_0(x) &= \exp^{-\frac{x}{2}} \\ e_1(x) &= \exp^{-\frac{x}{2}} (1 - x) \\ &\dots \\ e_k &= \exp^{-\frac{x}{2}} \frac{e^x}{k!} \frac{d^k}{dx^k} (x^k e^{-x}) \end{aligned} \quad (7)$$

W praktyce, używa się nieważonych wielomianów Laguerre'a w związku z szybszą zbieżnością ceny uwarunkowaną stopniem wielomianów, który jest ustalony. Opis zbieżności algorytmu w [11].

## 2. Krok 2.

Krokiem drugim metody Longstaffa - Schwartza jest wyznaczenie parametrów  $\beta_{t,k}$  (6) poprzez minimalizowanie wartości oczekiwanej kwadratu różnicy

$$\mathbb{E} \left( \mathbb{E}(V_{t+1}|X_t) - \sum_{k=0}^m \beta_{t,k} e_k(X_t) \right)^2 \quad (8)$$

Po zminimalizowaniu otrzymywane są następujące macierze

$$\begin{aligned} (\mathbf{M}_{ee})_{k,l} &= \mathbb{E}(e_k(X_t) e_l(X_t)), \\ (\mathbf{M}_{Ve})_k &= \mathbb{E}(\mathbb{E}(V_{t+1}|X_t) e_k(X_t)) \end{aligned} \quad (9)$$

W związku z mierzalnością  $e_k(X_t)$  względem  $X_t$  możemy zapisać

$$(\mathbf{M}_{Ve})_k = \mathbb{E}(\mathbb{E}(V_{t+1} e_k(X_t) | X_t))$$

Wektor  $\beta_t$  równy  $[\beta_{t,0}, \beta_{t,1}, \dots, \beta_{t,m}]$  jest postaci

$$\beta_t = \mathbf{M}_{ee}^{-1} \mathbf{M}_{Ve} \quad (10)$$

W algorytmie będziemy generować  $N$  ścieżek aktywa bazowego. Wówczas estymatorami wyżej zdefiniowanych macierzy będą

$$\begin{aligned} (\hat{\mathbf{M}}_{ee})_{k,l} &= \frac{1}{N} \sum_{n=0}^N e_k(X_t^n) e_l(X_t^n), \\ (\hat{\mathbf{M}}_{Ve})_k &= \frac{1}{N} \sum_{n=0}^N V_{t+1}^n e_k(X_t^n), \end{aligned} \quad (11)$$

a estymatorem wektora  $\beta_t$  w kroku  $t$  jest

$$\hat{\beta}_t = \hat{\mathbf{M}}_{ee}^{-1} \hat{\mathbf{M}}_{Ve}$$

Wówczas algorytm w pseudokodzie wygląda następująco

---

**Algorithm 1** Longstaff-Schwartz Algorithm

---

**Ensure:**  $N, T, X_0, m$  is set

generate  $N$  paths of  $T$  timesteps of underlying asset

set  $V_T := h(X_T)$

**for**  $t = T - 1$  to  $1$  **do**

$$(\hat{\mathbf{M}}_{ee})_{k,l} := \frac{1}{N} \sum_{n=0}^N e_k(X_t^n) e_l(X_t^n)$$

$$(\hat{\mathbf{M}}_{Ve})_k := \frac{1}{N} \sum_{n=0}^N V_{t+1}^n e_k(X_t^n)$$

$$\hat{\beta}_t := \hat{\mathbf{M}}_{ee}^{-1} \hat{\mathbf{M}}_{Ve}$$

$$C_t := \sum_{k=0}^m \beta_{t,k} e_k(X_t)$$

**if**  $C_t > h(X_t)$  **then**

$$\text{set } V_t = e^{-r} V_{t+1}$$

**else**

$$\text{set } V_t = h(X_t)$$

**end if**

**end for**

---

W przypadku obliczeń wykorzystujących liczby losowe, algorytm Longstaffa-Schwartz (jako opierający się o ruch wstecz w czasie) wymaga przechowywania pełnych ścieżek przeprowadzonej symulacji. Prowadzi to do oczywistego problemu z ilością wymaganej pamięci: przeprowadzenie symulacji wykorzystującej np.  $10^6$  ścieżek o  $10^3$  punktów czasowych byłoby niemożliwe na przeciętnym komputerze osobistym (ponad dekadę od publikacji algorytmu).

W symulacjach komputerowych zazwyczaj wykorzystywany jest generator liczb pseudolosowych. Podstawową cechą takiego generatora jest możliwość powtórnego wygenerowania sekwencji liczb losowych, jeśli zainicjujemy go takim samym stanem początkowym. Można zatem odzyskać każdy wcześniejszy stan instrumentu bazowego przechowując jedynie pewną stałą liczbę danych (stan początkowy - *seed*).

Oczywiście przeprowadzanie pełnej symulacji w każdym kroku czasowym ( $T$  razy) nie jest akceptowalne z punktu widzenia złożoności obliczeniowej. Proszym rozwiązaniem tego problemu jest zapisanie nie tylko jednego stanu początkowego, ale całego wektora stanów występujących w trakcie symulacji.

Implementacja została oparta o propozycję opisaną w [2]. Autorzy sugerują następujący schemat symulacji:

1. Krok 1.

Stwórz wektor  $U$  przechowujący stan losowanego procesu stochastycznego (sumę kolejnych liczb losowych w każdej ze ścieżek).

Stwórz wektor  $S$  przechowujący stany generatora. Umieść w nim początkowy *seed*.



2. Krok 2.  
Wylosuj liczby dla wszystkich ścieżek symulacji (cały krok czasowy) i dodaj je odpowiednio do  $U$ .  
Jeśli osiągnąłś koniec symulacji (chwile wygaśnięcia opcji) przejdź do kroku 4.
3. Krok 3.  
Dopisz aktualny stan generatora do  $S$ .  
Przejdź do kroku 2.
4. Krok 4.  
Zakończ generowanie liczb losowych.

Po zakończeniu losowania otrzymujemy wektor  $U$  potrzebny do obliczenia ceny instrumentu bazowego w chwili wygaśnięcia opcji. Wykorzystując zapisane w  $S$  stany generatora liczb pseudolosowych, możemy odtworzyć dowolną wartość w symulacji. W całym programie każde z losowań przeprowadzamy tylko 2-krotnie (optymalnie z punktu widzenia złożoności obliczeniowej<sup>1</sup>). W efekcie unikamy wyjściowej złożoności pamięciowej  $O(TN)$  i otrzymujemy algorytm klasy  $O(T + N)$  (przy czym zazwyczaj  $T \ll N$ ).

## 1.6 Total Least Squares

W klasycznym ujęciu do aproksymacji metodą najmniejszych kwadratów używany jest algorytm *Ordinary Least Squares* (OLS). Wzorując się na [12] zaimplementowano również metodę *Total Least Squares* (TLS) i przeprowadzono porównanie rezultatów numerycznych obu metod.

Na początek założmy, że mamy macierz danych  $A$  wymiaru  $m \times n$  oraz wektor obserwacji  $y$  długości  $m$ . Problem OLS można sformułować następująco:

$$\|y - Ax\|_2 \rightarrow_x \min$$

Jednoznaczne rozwiązanie problemu OLS dane jest wzorem:

$$x_{OLS} = (A^T A)^{-1} A^T y.$$

W roku 1980 Golub i VanLoan opisali metodę TLS, formułując następujący problem:

$$\|A | r\|_F \rightarrow_{A,r} \min$$

gdzie  $\|\cdot\|$  oznacza normę Frobeniusa czyli  $\|\cdot\|_F^2 = \sum_i \sum_j |b_{ij}|^2$ . Gdy otrzymamy rozwiązanie minimalizujące  $[A | r]$ , wówczas  $\forall x : (A + \tilde{E})x = y + \tilde{r}$ ,  $x$  jest również rozwiązaniem problemu TLS. Niech  $\tilde{s}_n$  będzie najmniejszą wartością własną macierzy  $[A]$  i  $s_{n+1}$  będzie najmniejszą wartością własną macierzy  $[A | r]$ . Wówczas

<sup>1</sup>Rzeczywistą szybkość algorytmu można poprawić przy wykorzystaniu obliczeń równoległych. Jeśli podczas pierwszego losowania  $N$  ścieżek w pewnym kroku czasowym  $t$  zapiszemy więcej stanów generatora, w drugim losowaniu będziemy mogli skorzystać z kilku procesorów.

z twierdzenia o rozdzielaniu  $\tilde{s}_n > s_{n+1}$ . Gdy  $\tilde{s}_n$  jest ściśle większa od  $s_{n+1}$ , istnieje jednoznaczne rozwiązanie problemu TLS i dane jest ono wzorem:

$$x_{TLS} = (A^T A - s_{n+1}^2 I)^{-1} A^T y.$$

Zatem problem TLS jest równoważny z problemem rozwiązania problemu minimalizacji  $\|(A + \tilde{E})x - (b + \tilde{r})\|_2$ .

## 1.7 Estymator nieobciążony

W pracy [9] zwrócono uwagę, że wyznaczany za pomocą powyższych metod estymator jest obciążony. Wobec tego rozważano wprowadzenie kolejnej modyfikacji zaproponowanej przez autora wspomnianego artykułu. Mianowicie zamiast jednej, należy wykonać dwie serie symulacji ścieżek. Pierwszą z nich wykorzystuje się wyłącznie do obliczenia współczynników regresji liniowej (OLS oraz TLS). Następnie, zamiast obliczać wartości dopasowane na hiperpłaszczyźnie regresji, do otrzymanych funkcji wstawia się ceny instrumentu bazowego pochodzące z drugiej serii symulacji. Uzyskuje się w ten sposób nieobciążony estymator wartości oczekiwanej. ten pomysł nie został w efekcie uwzględniony w ostatecznej wersji kodu, ponieważ nie można go uzgodnić z pomysłem używanie ścieżek *in-the-money*.

## 2 Wyznaczenie parametrów greckich

W przypadku opcji waniliowych, portfeli takich opcji oraz opcji z barierą europejską (takie opcje rozważano tylko w wersji opcji europejskich) poza ceną opcji, zwracane są także wartości parametrów greckich, które obliczamy wraz z ceną opcji. Korzystamy przy tym z wniosków pochodzących z [9]. W pracy tej opisano dwie metody wyznaczania parametrów greckich dla opcji amerykańskich: metoda różniczkowania po trajektoriach oraz metoda współczynnika wiarygodności.

W metodzie różniczkowania po trajektoriach zauważamy, że dla ceny aktywa bazowego w chwili  $t$  danego funkcją  $S_t$ , dla opcji  $V_t$  z funkcją wypłaty  $f$  oraz parametru  $\eta$ , od którego zależy cena opcji, mamy:

$$\frac{\partial V_0(S_0)}{\partial \eta} = \mathbb{E} \left( \exp(-r_d \tau) \frac{\partial f(S_\tau(\eta))}{\partial S_\tau} \frac{\partial S_\tau(\eta)}{\partial \eta} \right), \quad (12)$$

gdzie  $\tau$  nazywamy czasem zatrzymania (długością czasu do momentu realizacji opcji) a  $r_d$  jest krajową stopą procentową. Możliwe jest zatem obliczenie parametru greckiego poprzez uśrednienie pewnej funkcji (iloczynu 2 pochodnych cząstkowych) po realizacjach na ścieżkach symulacji Monte Carlo. Wystarczy wycenić tę funkcję tak, jak wyceniamy wartości realizacji opcji w kolejnych ścieżkach symulacji.

Pierwsza pochodna w tym iloczynie jest zależna nie od tego, który parametr grecki obliczamy, a jedynie od funkcji wypłaty. Może być zatem zaimplementowana niezależnie od algorytmu przeprowadzającego symulację. Jedyne warunki to istnienie pochodnej funkcji wypłaty  $f(S)$ . Dla wielu opcji waniliowych (np. *call* lub *put*) funkcja wypłaty nie jest różniczkowalna, ale brak różniczkowalności w jednym punkcie można

zignorować (prawdopodobieństwo trafienia na ten punkt w symulacji jest bardzo małe). Dla pewnych opcji (np. opcji binarnych) brak różniczkowalności w jednym punkcie uniemożliwia bezpośrednie zastosowanie przedstawionego podejścia (sposobem na poradzenie sobie z takim problemem może być zastąpienie prawdziwej funkcji wypłaty pewnym jej wyładzeniem).

Druą pochodną w tym iloczynie nie zależy od funkcji wypłaty i dla modelu Blacka-Scholes może być w wielu wypadkach obliczona analitycznie

$$\frac{\partial S_\tau(\eta)}{\partial \eta} = \frac{\partial}{\partial \eta} \left( S_0 \exp\left((r_d - r_f - \sigma^2/2)\tau + \sigma W_\tau\right) \right). \quad (13)$$

We wzorze tym  $r_d$  jest krajową stopą procentową a  $r_f$  stopą procentową zagraniczną (stopą dywidendy dla rynku Equity).

Dobrym przykładem poprawnego obliczenia analitycznej pochodnej jest policzenie parametru  $\Delta$ , czyli pochodnej po początkowej wartości akcji  $S_0$ . Mamy wtedy

$$\frac{\partial S_\tau}{\partial S_0} = \exp\left((r_d - r_f - \sigma^2/2)\tau + \sigma W_\tau\right) = \frac{S_\tau}{S_0}.$$

Także parametr  $\rho$ , czyli pochodną po stopie procentowej  $r_d$  można policzyć ze wzoru (12). Mamy wtedy

$$\frac{\partial V_0(S_0)}{\partial r_d} = \mathbb{E} \left( -\tau \exp(-r_d \tau) f(S_\tau) + \exp(-r_d \tau) \frac{\partial f(S_\tau)}{\partial S_\tau} \frac{\partial S_\tau}{\partial r_d} \right),$$

gdzie

$$\frac{\partial S_\tau}{\partial r_d} = \tau S_0 \exp\left((r_d - r_f - \sigma^2/2)\tau + \sigma W_\tau\right).$$

Aby znaleźć parametr *Vega*, czyli pochodną po zmienności  $\sigma$  różniczkujemy formalnie

$$\frac{\partial S_\tau}{\partial \sigma} = (W_\tau - \sigma \tau) S_0 \exp\left((r_d - r_f - \sigma^2/2)\tau + \sigma W_\tau\right).$$

Takie formalne postępowanie jest dopuszczalne, ponieważ w symulacjach nie mamy procesu stochastycznego  $W_t$ , ale realizację zmiennej losowej  $\xi\sqrt{t}$ , gdzie  $\xi$  jest wzięte z rozkładu  $\mathcal{N}(0, 1)$ .

Podobnie znajdujemy parametr  $\Theta$ , czyli pochodną po czasie  $t$  (wykorzystujemy fakt, że  $\tau = T - t$ )

$$\frac{\partial V_0(S_t)}{\partial t} = -\mathbb{E} \left( -r_d \exp(-r_d \tau) f(S_\tau) + \exp(-r_d \tau) \frac{\partial f(S_\tau)}{\partial S_\tau} \frac{\partial S_\tau}{\partial \tau} \right),$$

gdzie

$$\frac{\partial S_\tau}{\partial \tau} = \left( r_d - r_f - \frac{1}{2}\sigma^2 + \frac{1}{2}\sigma \frac{W_\tau}{\tau} \right) S_0 \exp\left((r_d - r_f - \sigma^2/2)\tau + \sigma W_\tau\right).$$

W tym ostatnim wzorze wyrażenie  $\frac{1}{2}W_\tau/\tau$  jest zapisem różniczkowania wyrażenia  $\xi\sqrt{\tau}$  po  $\tau$ .

Metoda różniczkowanie po trajektoriach jest właściwa także dla parametrów greckich będących pochodnymi ceny opcji rzędu wyższego niż 1, ale wymagałoby to obliczenia odpowiedniej pochodnej funkcji wypłaty. Jednak ze względu na zazwyczaj nieciągłą pochodną funkcji wypłaty po cenie aktywa, nie da się w ten sposób obliczyć np. parametru gamma.

W tym celu skorzystamy z drugiego pomysłu opisanego w [9] - metody współczynnika wiarygodności (ang. "likelihood ratio"). W tej metodzie obliczenie parametru greckiego sprowadza się do wyceny opcji w kroku czasowym  $S_1$  oraz obliczenia ważonej średniej tych wycen. Cenę opcji można zapisać wzorem

$$V_0(S_0) = \mathbb{E}(\exp(-r_d\Delta T)V_1(S_1)|S_0) = \int \exp(-r_d\Delta T)V_1(S_1)g_1(S_0, S_1)dS_1,$$

gdzie  $g_1(S_0, S_1)$  jest gęstością  $S_1$  pod warunkiem  $S_0$ , czyli gęstością przejścia ze stanu  $S_0$  do  $S_1$  a  $\Delta T$  jest czasem przejścia ze stanu  $S_0$  do  $S_1$ .

Jeśli  $F(S_1) = \exp(-r_d\Delta T)V_1(S_1)$  jest zdyskontowaną na chwilę 0 wartością opcji w chwili 1, to greckie parametry dane są wzorem

$$\begin{aligned} \frac{\partial V_0(S_0)}{\partial \eta} &= \frac{\partial}{\partial \eta} \int F(S_1)g_1(S_0, S_1)dS_1 \\ &= \int \left( \frac{\partial F(S_1)}{\partial \eta} g_1(S_0, S_1) + F(S_1) \frac{\partial g_1}{\partial \eta} \right) dS_1 \\ &= \int \left( \frac{\partial F(S_1)}{\partial \eta} + F(S_1) \frac{\partial \log g_1}{\partial \eta} \right) g_1(S_0, S_1)dS_1 \\ &= \mathbb{E} \left( \frac{\partial F(S_1)}{\partial \eta} + F(S_1) \frac{\partial \log g_1}{\partial \eta} \right). \end{aligned} \quad (14)$$

W obliczeniu parametru  $\Gamma$  potrzebujemy policzenia drugiej pochodnej

$$\Gamma(S_0) = \frac{\partial \Delta(S_0)}{\partial S_0} = \mathbb{E} \left( F(S_1) \left( \frac{\partial^2 \log g_1}{\partial S_0^2} + \left( \frac{\partial \log g_1}{\partial S_0} \right)^2 \right) \right). \quad (15)$$

Ponieważ funkcja  $g_1(S_0, S_1)$  dla rynku Blacka-Scholes dana jest wzorem analitycznym, ta metoda może być wykorzystana do liczenia wszystkich greckich parametrów. Należy jedynie policzyć odpowiednie pochodne:  $\frac{\partial \log g_1}{\partial S_0}$  do obliczenia parametru *delta*  $\frac{\partial \log g_1}{\partial \sigma}$  do obliczenia parametru *vega* i  $\frac{\partial \log g_1}{\partial t}$  do obliczenia parametru *theta* oraz pochodną  $\frac{\partial F(S_1)}{\partial \eta}$ .

Wykorzystując wzór na funkcję gęstości

$$g_1(S_0, S_1) = \frac{1}{\sqrt{2\pi\sigma^2\Delta T}S_1} \exp\left(-\left(\log(S_1/S_0) - (r_d - r_f - \sigma^2/2)\Delta T\right)^2 / 2\sigma^2\Delta T\right),$$

otrzymujemy odpowiednie wzory na pochodne:

$$\begin{aligned}\frac{\partial \log g_1}{\partial S_0} &= \frac{d(S_1)}{S_0 \sigma \sqrt{\Delta T}}, \\ \frac{\partial^2 \log g_1}{\partial S_0^2} &= -\frac{d(S_1) \sigma \sqrt{\Delta T} + 1}{S_0^2 \sigma^2 \Delta T}, \\ \frac{\partial \log g_1}{\partial \sigma} &= \frac{d^2(S_1) - d(S_1) \sigma \sqrt{\Delta T} - 1}{\sigma}, \\ \frac{\partial \log g_1}{\partial t} &= -\frac{\partial \log g_1}{\partial \Delta T} = -\frac{1}{2\Delta T} \left( d^2(S_1) + 2d(S_1) \frac{(r_d - r_f - \sigma^2/2) \sqrt{\Delta T}}{\sigma} - 1 \right),\end{aligned}$$

gdzie

$$d(S) = \frac{\log(S/S_0) - (r_d - r_f - \sigma^2/2)\Delta T}{\sigma \sqrt{\Delta T}}.$$

Korzystając z funkcji dyskontowych można pochodną  $\frac{\partial \log g_1}{\partial t}$  zapisać następująco

$$\frac{\partial \log g_1}{\partial t} = -\frac{1}{2\Delta T} \left( d^2(S_1) - d(S_1) \sigma \sqrt{\Delta T} - 2d(S_1) \log \frac{disFac}{disFactF} / \sigma \sqrt{\Delta T} - 1 \right),$$

gdzie  $disFac$  i  $disFactF$  oznaczają czynniki dyskontowe na okres  $\Delta T$ , odpowiednio krajowy i zagraniczny (czynnik implikowany przez stopę dywidendy dla rynku Equity).

Dla opcji europejskich metoda różniczkowania po trajektoriach daje dobre wyniki jedynie dla opcji *call* i *put*. Dla opcji binarnych oraz opcji z barierą europejską wyniki są niepoprawne bo funkcja wypłaty jest nieciągła. W tych przypadkach dla opcji europejskich stosować można metodę współczynnika wiarygodności. Obliczanie tą metodą parametrów greckich daje dobre wyniki, bo stan  $S_1$  to moment realizacji opcji a  $\Delta T$  to czas do momentu zapadalności opcji. Wynika z tego, że  $F(S_1) = F(S_T) = \exp(-r_d T) f(S_T)$ , gdzie  $f$  jest funkcja wypłaty z opcji. Wynika stąd, że pochodna  $\frac{\partial F}{\partial \eta}$  jest równa zero z wyjątkiem przypadku obliczania parametrów *rho* oraz *theta*. Dla parametru *theta* mamy

$$\frac{\partial F}{\partial t} = -\frac{\partial F}{\partial T} = r_d F = -\log(disFact)/TF.$$

Jednak dla opcji amerykańskiej policzone metodą współczynnika wiarygodności wartość parametrów greckich obarczona jest dużym błędem. Wynika on z faktu, że dla opcji amerykańskiej stan  $S_1$  to krok czasowy symulacji, w którym tylko na nielicznych trajektoriach następuje realizacja opcji. W efekcie, aby otrzymać wynik z małym błędem należałoby wykonać symulacje dla bardzo dużej liczby trajektorii. Dlatego wszystkie parametry, dla których jest to możliwe zostały policzone metodą różniczkowania po trajektoriach. Ponieważ parametru  $\Gamma$  nie można policzyć różniczkując po trajektoriach obliczanie tego współczynnika dla opcji amerykańskich wykonano inną metodą. Metodą różniczkowania po trajektorii obliczono wartości współczynnika  $\Delta(S_0 + ds)$  oraz  $\Delta(S_0 - ds)$ . Współczynnik  $\Gamma$  obliczono z tych wartości metodą różnic skończonych

$$\Gamma = \frac{\Delta(S_0 + ds) - \Delta(S_0 - ds)}{2ds}.$$

### 3 Opcje barierowe (barrier options)

Opcje barierowe są szczególnym typem instrumentów zależnych od trajektorii. Wpłata z opcji barierowej zależy od tego, czy w ustalonych momentach czasu lub podczas całego życia opcji, cena instrumentu podstawowego spadnie poniżej lub przekroczy pewną ustaloną wielkość zwaną barierą. Opcje barierowe dzielimy na opcje wyjścia (knock-out) i wejścia (knock-in). Te pierwsze dezaktywują się, gdy cena instrumentu podstawowego przekroczy barierę, zaś drugie w chwili przekroczenia bariery aktywują się. Opcje te możemy łączyć z dowolnym typem wypłaty zadanej przez funkcję  $S_t$  i dowolnym typem wykonania.

W zależności od sposobu monitorowania bariery rozróżniamy następujące opcje:

- **z barierą monitorowaną w sposób ciągły** (tzw. bariera amerykańska) - sprawdza się, czy w żadnej chwili życia opcji, cena instrumentu podstawowego nie osiągnęła bariery;
- **z barierą monitorowaną w sposób dyskretny** - cena instrumentu podstawowego jest monitorowana w wybranych momentach wyszczególnionych w kontrakcie opcyjnym (np. codziennie lub w każdy wtorek miesiąca)
- **z barierą europejską** - cena instrumentu podstawowego monitorowana jest tylko w momencie wygaśnięcia
- **window barrier option** - cena monitorowana jest pomiędzy ustalonymi z góry datami

Ze względu na to, kiedy opcja zaczyna istnieć lub wygasa pod wpływem zdarzenia barierowego możemy wyróżnić następujące opcje:

- **up-and-in** - opcja zaczyna istnieć, gdy cena przekroczy barierę, wówczas wypłata jest równa:

$$X = f(S_\tau) \mathbb{1}_{\{\max_{t \leq \tau} S_t \geq B\}}$$

- **up-and-out** - opcja wygasa, gdy cena przekroczy barierę, wówczas wypłata jest równa:

$$X = f(S_\tau) \mathbb{1}_{\{\max_{t \leq \tau} S_t \leq B\}}$$

- **down-and-out** - opcja wygasa, gdy cena spadnie poniżej bariery, wówczas wypłata jest równa:

$$X = f(S_\tau) \mathbb{1}_{\{\min_{t \leq \tau} S_t \geq B\}}$$

- **down-and-in** - opcja zaczyna istnieć, gdy cena spadnie poniżej bariery, wówczas wypłata jest równa:

$$X = f(S_\tau) \mathbb{1}_{\{\min_{t \leq \tau} S_t \leq B\}}$$

Ponadto, możemy rozważać opcje dwubarierowe:

- **double-knock-out** - opcja wygasa, gdy cena spadnie poniżej  $B_1$  lub przekroczy barierę  $B_2$ , wówczas wypłata jest równa

$$X = f(S_\tau) \mathbb{1}_{\{\forall t \in [0, \tau] B_1 \geq S_t \geq B_2\}}$$

- **up-in-up-out** - opcja zaczyna istnieć, gdy cena przekroczy  $B_1$  i wygasa, gdy spadnie poniżej  $B_2$ , wówczas wypłata jest równa

$$X = f(S_\tau) \mathbb{1}_{\{\max_{t \leq \tau_0} S_t \geq B_1\} \cap \{\min_{\tau_0 \leq t \leq \tau} S_t \geq B_2\}}$$

- **down-in-down-out** - opcja zaczyna istnieć, gdy cena spadnie poniżej  $B_1$  i wygasa, gdy spadnie poniżej  $B_2$ , wówczas wypłata jest równa

$$X = f(S_\tau) \mathbb{1}_{\{\min_{t \leq \tau_0} S_t \leq B_1\} \cap \{\min_{\tau_0 \leq t \leq \tau} S_t \geq B_2\}}$$

- **double-no-touch** - opcja wygasa, jeśli jej cena dotknie jednej z barier, wówczas wypłata jest równa

$$X = f(S_\tau) \mathbb{1}_{\{\forall t \in [0, \tau] B_1 > S_t > B_2\}},$$

gdzie  $\tau$  jest momentem wykonania opcji - dla opcji europejskiej jest to  $T$ , zaś dla opcji amerykańskich to dowolny moment stopu z przedziału  $[0, T]$ , a  $\tau_0 \in [0, \tau]$  jest pewnym momentem stopu. W sposób analogiczny można zapisać wypłatę z opcji z barierą monitorowaną w sposób dyskretny, z tym, że zamiast rozpatrywać powyższe indykatory, maksima i minima  $\forall t \in [0, T]$  rozpatrujemy je dla  $t \in \{t_1, t_2, \dots, t_n\}$ , gdzie  $t_1, t_2, \dots, t_n$  są ustalonymi w kontrakcie momentami czasu. Podobnie zapisujemy wypłatę z opcji z barierą monitorowaną w oknie czasowym, zastępując znów indykatorem lub ekstremum po całym zbiorze, indykatozem po zbiorze  $t \in [\tau_1, \tau_2]$ , gdzie  $\tau_1, \tau_2$  są ustalonymi w kontrakcie opcyjnym momentami rozpoczęcia i zakończenia monitorowania bariery.

### 3.1 Wycena opcji z barierą obserwowaną w sposób dyskretny, modyfikacja dla bariery okienkowej

Metody wyceny opcji europejskich z barierą obserwowaną w sposób dyskretny są, z dokładnością do postaci funkcji wypłaty, identyczne dla wszystkich typów barier. Wobec tego w niniejszym podrozdziale przedstawiony zostanie wzór jedynie dla przykładowo wybranej opcji: up-and-out. Cena, czyli estymator wartości zdyskontowanej wypłaty wyraża się w tym przypadku przez wartość oczekiwaną następującej zmiennej:

$$V_0 = f(S(T)) \mathbb{1}_{\{\max_{t \in t_1, \dots, t_n} S(t) < B\}}, \quad (16)$$

gdzie  $f(S(t))$  jest potencjalną zdyskontowaną wypłatą z odpowiedniej opcji waniliowej, gdyby była wykonana w chwili  $t$ ,  $T$  jest momentem zapadalności opcji,  $t_1, \dots, t_n$  są momentami obserwacji bariery, zaś  $B$  - jej wartością. Wzory na ceny opcji o pozostałych typach bariery powstają w analogiczny sposób, przez wstawienie odpowiednich dla nich funkcji indykatorowych. Ważny jest jedynie fakt, że wartości tych funkcji zależą jedynie od zachowania się ceny aktywa bazowego w stosunku do bariery w pewnej skończonej liczbie momentów czasu.

Metoda Monte Carlo posłużyła autorkom do symulowania ścieżek cen instrumentu bazowego. Dokonana została ocena, na których ścieżkach instrument bazowy przekroczył barierę oraz ostatecznie obliczona została średnia wartość zdyskontowanej wypłaty przy uwzględnieniu wszystkich ścieżek oraz po sprowadzeniu do zera wartości wypłaty na tych ścieżkach, gdzie opcja okazała się nie być aktywna.

W prezentowanym algorytmie w celu wyboru momentów czasu, w których obserwowana jest bariera, posłużono się wektorem *monitoringGrid* będącym jednym z wyników zwracanych przez funkcję *discountCalculate* zaimplementowaną w pliku *calculate\_functions*. Jest to wyznaczony za pomocą funkcji kalendarzowych wektor o liczbie współrzędnych równej liczbie rozważanych punktów na siatce czasu. Wartość 1 na współrzędnej tego wektora oznacza, że w danym punkcie czasowym (węźle) obserwowana jest bariera. W przeciwnym przypadku odnotowywana jest wartość 0. Stosowany w programie algorytm dla monitorowania dyskretnego, zanim porówna wartość aktywa z wielkością bariery w danym momencie czasu, ma za zadanie sprawdzić, czy odpowiednia współrzędna wektora *monitoringGrid* ma wartość 1 (bariera jest obserwowana). Za pomocą jedynie drobnej modyfikacji autorki przeszły do wyceny opcji z barierą okienkową. Skorzystano mianowicie ze zmiennych *nodeWindow\_start* i *nodeWindow\_end* (również tworzonych przez funkcję *discountCalculate*), które wyznaczają odpowiednio pierwszą i ostatnią współrzędną na siatce czasu, dla których obserwowana jest bariera. Przekształcono zmienną *monitoringGrid* tak, by przyjmowała 0 na wszystkich współrzędnych o numerach mniejszych niż *nodeWindow\_start* i większych niż *nodeWindow\_end*.

### 3.2 Wycena opcji z barierą obserwowaną w sposób ciągły

Przypadek opcji z barierą obserwowaną w sposób ciągły jest nieco trudniejszy. Problem stanowi fakt, że przeprowadzając symulacje Monte Carlo, możliwe jest wygenerowanie ścieżek cen instrumentu bazowego jedynie dla skończenie wielu węzłów na każdej trajektorii czasu. W ten sposób, biorąc pod uwagę ścieżkę, na której cena aktywa nie przekracza bariery w żadnym z powyższych węzłów, nie można mieć pewności, że takie zdarzenie nie nastąpiło pomiędzy węzłami. Stosując zatem wzory takie jak 19, czy 16, można uzyskać jedynie estymatory obciążone cen opcji. Eliminacja obciążenia w prezentowanym przez autorki algorytmie odbywa się za pomocą techniki tzw. mostu Browna (*Brownian bridge*). Polega on na estymacji prawdopodobieństw, że bariera nie zostanie przekroczona na żadnym z odcinków czasu pomiędzy węzłami. Dla opcji jednobarieryowych analityczne, proste w implementacji wzory są znane (por. [3] i [8]) i wyrażają się w następujący sposób:

$$\mathbb{P}[M^{(m)} < B | S^{(m)}, S^{(m+1)}] = 1 - \xi^{(m)}(B), \quad (17)$$

$$\mathbb{P}[L^{(m)} > b | S^{(m)}, S^{(m+1)}] = 1 - \xi^{(m)}(b), \quad (18)$$

gdzie  $B$  jest barierą górną,  $b$  dolną,  $M^{(m)} = \max\{S(t) : t \in [t_m, t_{m+1}]\}$ ,  $L^{(m)} = \min\{S(t) : t \in [t_m, t_{m+1}]\}$ .

Jeśli przez  $P^{(m)}$  oznaczymy  $\mathbb{P}[M^{(m)} < B | S^{(m)}, S^{(m+1)}]$  (dla bariery górnej) lub  $\mathbb{P}[L^{(m)} > b | S^{(m)}, S^{(m+1)}]$  dla bariery dolnej, to nieobciążonym estymatorem ceny opcji typu knock-out będzie wartość oczekiwana zmiennej:

$$V_0 = f(S(T)) \mathbb{1}_{\{\max_{t \in t_1, \dots, t_n} S(t) < B\}} \prod_{m=1}^M P^{(m)}, \quad (19)$$



gdzie  $\min[S^{(m)}, S^{(m+1)}] > b$ ,  $\max[S^{(m)}, S^{(m+1)}] < B$ ,

$$\xi^{(m)}(x) = \exp\left(-2 \ln\left(\frac{x}{S^{(m)}}\right) \ln\left(\frac{x}{S^{(m+1)}}\right) / \sigma^{(m)} \delta t_m\right).$$

Skoro zawarty we wzorach iloczyn  $\prod_{m=1}^M P^{(m)}$  oznacza prawdopodobieństwo, że na żadnym z odcinków czasów pomiędzy obserwowanymi węzłami nie zostanie przekroczona bariera, prawdopodobieństwem przekroczenia bariery na co najmniej jednym z takich odcinków jest liczba  $1 - \prod_{m=1}^M P^{(m)}$ , wobec czego można łatwo wyznaczyć cenę opcji typu knock-in jako wartość średnią zmiennej:

$$V_0 = f(S(T)) \mathbb{1}_{\{\max_{t \in t_1, \dots, t_n} S(t) < B\}} \left(1 - \prod_{m=1}^M P^{(m)}\right) + f(S(T)) \mathbb{1}_{\{\max_{t \in t_1, \dots, t_n} S(t) > B\}}.$$

Dla opcji dwubarierowych nieobciążone estymatory odpowiednich prawdopodobieństw wyrażają się przez sumy nieskończonych szeregów ([3]). W ([8]) zaproponowano jednak dobre oszacowania tych liczb, wykorzystujące tzw. ograniczenia Frechet'a. Dla opcji typu double-knock-out wyrażają się one następująco:

$$P^{(m)} \geq P_L^{(m)} = \max[1 - (\xi^{(m)}(B) + \xi^{(m)}(b)), 0],$$

$$P^{(m)} \leq P_U^{(m)} = \min[1 - \xi^{(m)}(B), 1 - \xi^{(m)}(b)].$$

W ([8]) powyższe ograniczenia przedstawione są dla opcji opiewającej na wiele instrumentów bazowych. Wówczas rozważane są różne typy barier dla różnych instrumentów. Stwierdza się, że oszacowanie górne jest bliższe szukanemu prawdopodobieństwu, gdy zdarzenia mówiące o przekroczeniu rozważanych barier są dodatnio skorelowane, zaś oszacowanie dolne jest bliższe prawdziwej wartości w przypadku ujemnej korelacji. W sytuacji ograniczenia się do opcji typu double-knock-out lub double-no-touch dla jednego instrumentu, można natychmiast zauważyć, że zdarzenia przekroczenia górnej i dolnej bariery są skorelowane ujemnie, przez co właściwsza jest dolna wartość oszacowania. Górne oszacowanie, bliższe niż  $P_U$  prawdziwej wartości może być wyznaczone przez liczbę:

$$P_I^{(m)} = \prod_{m=1}^M [1 - \xi^{(m)}(B)][1 - \xi^{(m)}(b)],$$

odpowiadającą prawdopodobieństwu nieprzekroczenia bariery górnej ani dolnej, w przypadku gdyby były to zdarzenia niezależne. W tej sytuacji ostatecznie przybliżamy liczbę  $P^{(m)}$  przez średnią oszacowania górnego i dolnego:  $(P_L + P_I)/2$ .

W prezentowanym programie opcje z pojedynczą barierą oraz barierami typu double-knock-out i double-no-touch wyceniane są przez wyestymowanie wartości średnich omawianych w niniejszym rozdziale zmiennych i przemnożenie ich przez odpowiednie mosty Browna. Trudniejszy okazał się przypadek opcji typu knock-in-knock-out.

Posługując się techniką analogiczną do wyżej opisanej, wyznaczono przybliżone prawdopodobieństwa, że w danym odcinku czasu pomiędzy węzłami cena instrumentu bazowego przekroczy barierę aktywującą opcję (o ile nie stało się to wcześniej) i jednocześnie nie przekroczy bariery dezaktywującej, a także prawdopodobieństwo, że pomiędzy danymi węzłami przekroczona zostanie bariera dezaktywująca, jeśli na węźle początkowym opcja jest aktywna. Z powodu katastrofalnie narastającej liczby możliwych przypadków, autorki nie podjęły się jednak rozważania wszelkich możliwych zdarzeń za pomocą funkcji indykatorowych i przemnażania ich przez odpowiednie prawdopodobieństwa mówiące o aktywnym stanie opcji. Zamiast tego została zastosowana technika opisana w ([7]). Mianowicie otrzymane prawdopodobieństwa porównano z próbką wylosowaną z rozkładu jednostajnego. Jeśli prawdopodobieństwo przekroczenia bariery na danym odcinku było większe niż odpowiednia liczba z próbki, uznawano, że bariera została przekroczona. W tej sytuacji, posiadając już dane nie tylko o zachowaniu ceny instrumentu bazowego na węzłach siatki czasu, lecz również pomiędzy nimi, można było zastosować standardowy wzór na wypłatę dany w rozdziale 1., dokonując dyskontowania i obliczenia wartości średniej z symulowanych ścieżek.

## 4 Wyznaczanie parametrów greckich dla barierowych opcji europejskich

### 4.1 Metody wyznaczania parametrów greckich

Kolejną istotną kwestią przy wycenie opcji jest badanie jej wrażliwości na małe zmiany parametrów. Zauważmy, że funkcja wyceny opcji jest zależna od jej aktualnej ceny, czasu do zapadalności, zmienności oraz stopy procentowej. Różniczki funkcji wyceny po tych parametrach nazywamy współczynnikami greckimi. W sposób szczególnie interesować nas tu będzie Delta  $\delta = \frac{\partial V}{\partial S}$  oraz Gamma  $\gamma = \frac{\partial^2 V}{\partial S^2}$ . Przypomnijmy, że będziemy rozważać europejskie opcje barierowe, gdzie wypłata  $\Phi$  to

$$\Phi(\max_{t \leq T} S_t, \min_{t \leq T} S_t, S_T)$$

Rozróżniamy trzy podstawowe metody obliczania Greków. Są to:

1. *Metoda różnic skończonych* - opiera się na obliczaniu wyceny dla parametru o nieznacznie zmienionej wartości - w przypadku opcji barierowych jest mało dokładna, generuje obciążone estymatory i jest kosztowna, jeśli symulujemy dużą liczbę ścieżek w wielu momentach czasu.
2. *Różniczkowanie po trajektoriach* - polega na przejściu z różniczkowaniem pod znak wartości oczekiwanej. Niestety w przypadku opcji barierowych jest bezużyteczna ze względu na brak regularności funkcji wypłaty (nieróżniczkowalność indykatora).
3. *Likelihood method / Score method* - opiera się na różniczkowaniu względem rozkładu wypłaty, a nie samym różniczkowaniu funkcji wypłaty. Będziemy z niej korzystać, zatem poświęcimy jej więcej uwagi.

## 4.2 Rachunek Malliavina

Jak już zostało wspomniane, w naszych obliczeniach będziemy korzystać z trzeciej metody. Opiera się ona na rachunku Malliavina. Malliavin odkrył, że zmienne losowe, będące rozwiązaniami równań stochastycznych, można w odpowiednim sensie (sensie Malliavina) różniczkować. Chcących zagłębić się bardziej w tą tematykę odsyłamy do [1]. Jak wiadomo, cena w chwili zero opcji jest równa  $\mathbb{E}(\exp(-rT)\Phi)$ . Korzystając z metody całkowania przez części zaproponowanej przez Malliavina, można otrzymać wagi  $H$  takie, że  $\Delta$  i  $\Gamma$  mogą być przedstawione jako  $\mathbb{E}(\exp(-rT)\Phi H)$ . Autorki opierały się tu na pracy [5], korzystając z przedstawionych tam wyprowadzeń. Po pierwsze, autorki rozważyły transformację procesu  $S_t$  taką, że  $X_t = A(S_t)$ , gdzie  $A$  jest ściśle rosnącą funkcją i  $A(y) = \int_1^y \frac{du}{u\sigma(u)}$ . Wówczas

$$X_t = x + \int_0^t h(X_s)ds + W_t,$$

gdzie  $x = A(S_0)$  a  $h(u) = (r/\sigma(y) - (y\sigma(y))'/2)|_{y=A^{-1}(u)}$ . Odpowiada to logarymicznej zmianie w modelu Blacka - Scholesa. Przyjmujemy również następujące oznaczenia:  $M_t = \max_{s \leq t, s \in I} X_s$  oraz  $m_t = \min_{s \leq t, s \in I} X_s$ . Teraz korzystając z rachunku Malliavina możemy zapisać

$$\Delta = \partial_x \mathbb{E}(\Phi((M_t, m_t, X_T))) = \mathbb{E}(\Phi(M_t, m_t, X_T)H_1)$$

$$\Gamma = \partial_{x,x}^2 \mathbb{E}(\Phi((M_t, m_t, X_T))) = \mathbb{E}(\Phi(M_t, m_t, X_T)H_2)$$

dla pewnych zmiennych losowych  $H_1, H_2$ . Przy pewnych założeniach możemy uzyskać dane *explicité* wyrażenia powyższych zmiennych. Zakładamy teraz, że istnieje  $a > 0$  takie że funkcja  $\Phi(M, m, z)$  nie zależy od zmiennych  $(M, m)$  dla każdego  $(M, m, z)$  takich, że  $0 \leq M - x < a$  lub  $0 \leq x - m < a$ . Jeśli funkcja wypłaty zależy tylko od maksimum lub minimum, powyższe założenie może być uproszczone:

- dla pewnego  $a > 0$ , funkcja  $\Phi(M, z)$  nie zależy od  $M$  jeśli  $0 \leq M - x < a$  lub
- dla pewnego  $a > 0$ , funkcja  $\Phi(m, z)$  nie zależy od  $m$  jeśli  $0 \leq x - m < a$

Założenie to nie jest tak restrykcyjne dla większości rozważanych opcji barierowych. Podamy teraz wyrażenia na  $a$  dla poszczególnych opcji barierowych.

### 1. Opcje z jedną barierą

- up-and-out:  $a = B - x$ ,
- down-and-in:  $a = x - D$ ,

### 2. Opcje dwubarierowe

- przyjmujemy:  $a = \min(B_1 - x, x - B_2)$ ,

Aby wypisać wspomniane formuły, konieczne jest jeszcze wprowadzenie *procesu dominującego*  $X$ , przez który rozumiemy niemalejący, adaptowany, prawostronnie ciągły

proces  $(Y_t)_{0 \leq t \leq T}$ , taki, że  $\forall t \in I \quad |X_t - x| \leq Y_t$ . Do konstrukcji procesu  $Y$  wrócimy pod koniec tego paragrafu. Proces  $Y$  musi być gładki w sensie Malliavina, zatem poddajemy go transformacji funkcją  $\Psi \in C_b^\infty$ . Funkcja ta powinna spełniać:  $\Psi : [0, \infty) \rightarrow [0, 1]$  oraz  $\Psi(x) = 1$ , dla  $x \leq a/2$  i  $\Psi(x) = 0$ , dla  $x \geq 2$ . Autorki skorzystały tu z pewnej transformacji funkcji transportowej przyjmując

$$\Psi(x) = \begin{cases} 1 & \text{dla } x < \frac{a}{2} \\ \exp\left(-\frac{1}{1 - \frac{4}{a^2}(x - \frac{a}{2})^2} + 1\right) & \text{dla } x \in [\frac{a}{2}, a] \\ 0 & \text{dla } x > a. \end{cases}$$

Poniżej przytoczymy wzory na  $H_1$  oraz  $H_2$  przytoczone przez autorów ([5]).

$$H_1 = \delta\left(\frac{Z_T}{\int_0^T \Psi(Y_t) dt} \Psi(Y)\right) + \partial_x Z_T,$$

$$H_2 = \delta\left(\frac{H_1}{\int_0^T \Psi(Y_t) dt} \Psi(Y)\right) + \partial_x H_1,$$

gdzie  $\partial_x Z_T = Z_T \int_0^T h'(x + V_s)(dV_s - h(x + V_s)ds)$ ,  $\delta(u) = \int_0^1 u_t dW_t$ , zaś  $Z_T = \frac{d\mathbb{P}}{d\mathbb{Q}}|_{\mathcal{F}_T} = \exp\left(\int_0^T h(X_s) dV_s - \frac{1}{2} \int_0^T h^2(X_s) ds\right)$ , co definiuje miarę  $\mathbb{Q}$ , w której proces  $V_t = X_t - x$  jest standardowym ruchem Browna. Wzory te są dość skomplikowane, ale jeśli funkcja dryfu  $h(x) \equiv \mu$  (tak jak to jest w modelu Blacka - Scholesa), wzory się upraszczają:

$$H_1 = \delta\left(\frac{1}{\int_0^T \Psi(Y_t) dt} \Psi(Y)\right),$$

$$H_2 = \delta\left(\frac{H_1}{\int_0^T \Psi(Y_t) dt} \Psi(Y)\right) + \partial_x H_1.$$

Ostatnią kwestią jest konstrukcja procesu dominującego. Rozważamy dwa przypadki:

1.  $I$  jest całym przedziałem  $[0, T]$ .
2.  $S$  składa się ze skończonej liczby momentów.

Na początek ograniczymy się do przypadku (1). Najprostszym procesem dominującym jest wówczas proces

$$Y_t = \max_{s \leq t} (X_s - x) - \min_{s \leq t} (X_s - x).$$

Spełnia on konieczne założenia, jednak jego wadą jest fakt, że  $Y_t \notin \mathbf{D}^{2,p}$ , zatem nie może być używany do obliczania  $\Gamma$  i wyższych pochodnych. Innym pomysłem na konstrukcję procesu dominującego jest:

$$Y_t := 8 \left(4 \int_0^t \int_0^t \frac{|X_s - X_u|^\gamma}{|s - u|^{m+2}} ds du\right)^{\frac{1}{\gamma}} \frac{m+2}{m} t^{\frac{m}{\gamma}}$$

dla pewnej parzystej liczby  $\gamma$  oraz  $0 < m < \frac{\gamma}{2} - 2$ . Proces ten spełnia wymagane warunki i jest wystarczająco gładki do obliczania wyższych pochodnych.

W drugim przypadku znalezienie procesu dominującego o odpowiednich własnościach jest łatwiejsze. Znowu możemy skorzystać z 'extrema proces'

$$Y_t = \max_{0 \leq i \leq N: t_i \leq t} (X_{t_i} - x) - \min_{0 \leq i \leq N: t_i \leq t} (X_{t_i} - x),$$

jednak znów posłużyć on może jedynie do obliczania  $\Delta$ . Dla wyższych różniczek używamy tutaj procesu

$$Y_t = \sqrt{N \sum_{1 \leq i \leq N: t_i \leq t} (X_{t_i} - x_{t_{i-1}})^2}.$$

### 4.3 Wygładzanie bariery

Obliczanie greckich parametrów metodą różniczkowania po trajektoriach jest nie możliwe gdy funkcja wypłaty nie jest wystarczająco regularna. Aby wykorzystać tę metodę dla opcji barierowych potrzebne jest wygładzenie bariery. Weźmy dla przykładu opcję call z barierą down-out. Wypłatę z takiej opcji można opisać następująco:

$$P(S_T, S_{min}) = R(S_T - K)H(S_{min} - B)$$

Gdzie:

$$R(x) = \min(x, 0)$$

$$H(x) = 1_{\{x > 0\}}$$

Zastąpmy powyższe funkcje przez:

$$H_\epsilon = \frac{1}{2} [\tanh(\frac{x}{\epsilon}) + 1]$$

$$R_\epsilon = \int_{-\infty}^x H_\epsilon(x) dx$$

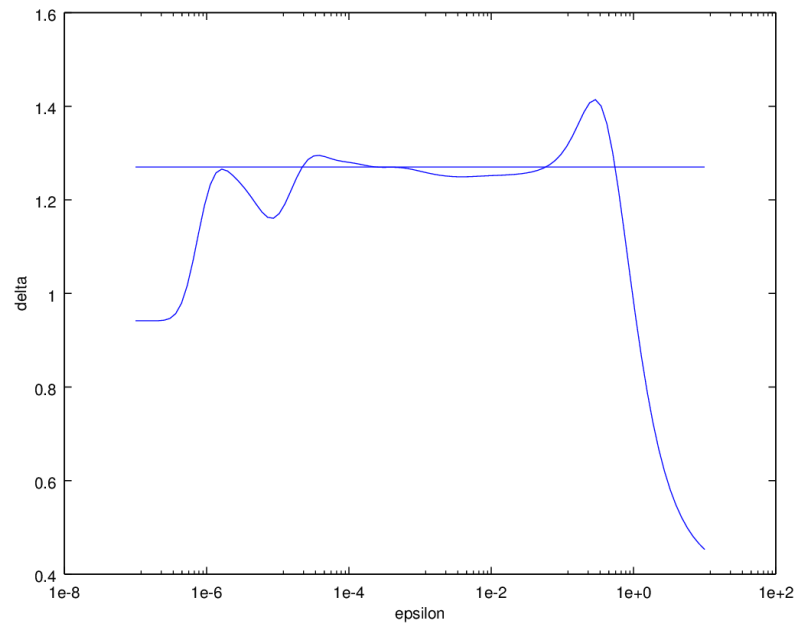
dla pewnego parametru epsilon. Dla tak zdefiniowanej wypłaty możemy już łatwo liczyć greki. (szczegóły w [13])

Problemem jest wybór parametru epsilon. Kiedy jest on zbyt duży wypłata jest przybliżana zbyt niedokładnie, kiedy zbiega do zera wariancja estymatora dąży do nieskończoności. W związku z tym narzucającym się rozwiązaniem jest wybranie epsilon na podstawie wariancji estymatora, niestety estymacja tej wariancji dla bardzo niskich epsilonów zawiedzie. Rozpatrzmy to na uproszczonym przykładzie - założmy że wypłata z opcji jest równa:

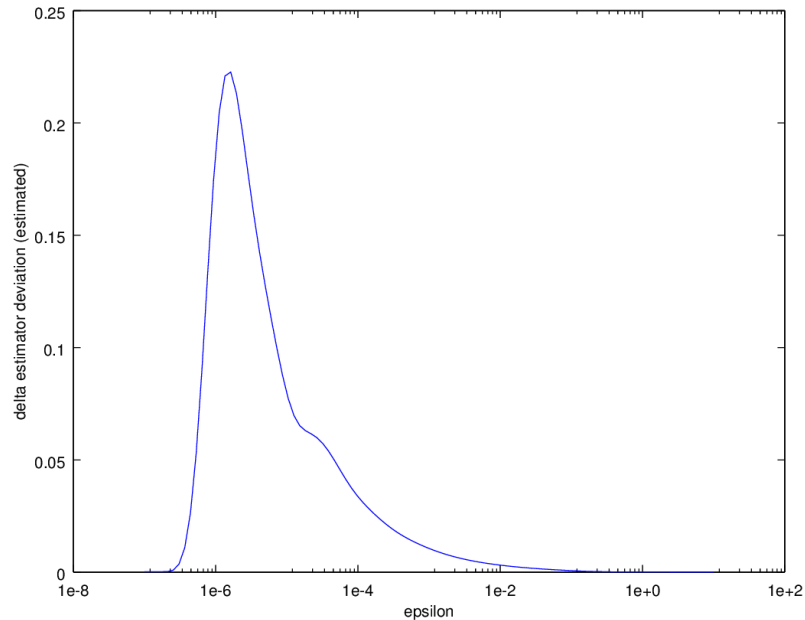
$$P(S_T, S_{min}) = H_\epsilon(1 - S_{min})$$

dla bardzo małego epsilon. Pochodna tej funkcji jest bliska zeru prawie wszędzie i ekstremalnie wysoka w wąskim obszarze wokół 1. W związku z tym jeśli nie wylosujemy żadnej próbki dostatecznie blisko 1 estymacja delty będzie bliska zeru - tak jakbyśmy próbowali zastosować metodę różniczkowania po trajektoriach bez wygładzania. Jednocześnie wariancja z próbki również będzie bliska zeru, podczas gdy prawdziwa wariancja jest bardzo wysoka.

Na poniższym wykresie jest pokazana wartości estymatora delty w zależności od epsilon: (prawdziwa wartość oznaczona poziomą kreską)



Jak widać, dla bardzo małych epsilonów wartość estymatora stabilizuje się na błędnej wartości. Poniżej wartość dokładności estymatora (a dokładniej jego odchylenie standardowe szacowane z próbki). Przy wycenie opcji waniliowej za pomocą Monte Carlo powinno być bliskie zeru, tutaj jednak wygładzanie barriery prowadzi do wysokiej wariancji - która rośnie wraz z malejącym epsilonem, niestety dla tych bardzo małych nie jesteśmy w stanie wywnioskować na podstawie próbki.



W zaimplementowanym kodzie została przyjęta następująca strategia:

- 1) Trajektorie są losowane tylko raz, aby zachować realny czas obliczeń.
- 2) Dla tych trajektorii są liczone estymatory dla różnych epsilon'ów. Startowym parametrem jest  $\epsilon = 1$ , jeśli (próbkowa) wariancja estymatora jest odpowiednio mała, sprawdzany jest kolejny, mniejszy parametr epsilon. Procedura jest powtarzana aż wariancja próbkowa zrobi się zbyt duża.

Powyższa strategia ma swoje uzasadnienie w fakcie że estymowana wariancja najpierw znacząco rośnie, by dopiero potem zbiec do fałszywej wartości. Jeśli nie przekroczy pewnego progu niezależnie od epsilon to oznacza to że bariera nie ma dużego znaczenia dla wyceny opcji - prawie nigdy nie jest przekraczana. W takim wypadku wycena oparta na  $\epsilon = 0$  nie zaburza znacząco wyniku. Cała metoda sprowadza się do kontrolowania błędu wariancji estymatora przy jednoczesnym minimalizowaniu błędu wygładzenia (zaburzenia funkcji wypłaty). Metoda ta ma więc z góry ustaloną optymistyczną dokładność, dla delty jest to  $\max(0.01, \text{delta}/100)$ , dla gammy  $\max(0.1, \text{gamma}/10)$ . Użycie jedynie względnego błędu prowadzi często do przedwczesnego zastopowania powyższego algorytmu (zbyt wysokiego parametru epsilon) - gdy estymacja (dla pewnego epsilon) jest bliska zeru. Różne stałe dla gammy i delty wynikają z realnych możliwości dokładności estymacji.

W kodzie są zaimplementowane metody liczenia greckich parametrów opcji jedno barrierowych, double-knock-out, i innych dwu barrierowych. W obecnym stanie metoda liczenia tych ostatnich nie jest używana, zamiast tego są one liczone przez złożenie opcji jedno barrierowych.

## 5 Opis produktu

Wynikiem projektu jest stworzenie silnika do symulacji Monte Carlo według modelu Longstaffa - Schwarza, mającego na celu wycenę opcji europejskich i amerykańskich oraz ich greckich parametrów. Wycena ta ma używać danych rynkowych takich jak czynniki dyskontowe, kalendarz różnych giełd i lat oraz status dnia - roboczy bądź nie.

Zaimplementowane typy opcji są wymienione poniżej. Dokładny opis funkcji wypłat poniższych opcji jest opisany w dokumentacji [10].

Wyceniane są następujące opcje europejskie i amerykańskie oraz obliczane ich greckie parametry:

1. Opcje waniliowe *call* oraz *put*,
2. Strategia *cash-or-nothing*,
3. Strategia *asset-or-nothing*,
4. Strategia *Straddel*,
5. Strategia *Risk Reversal*,
6. Strategia *Butterfly*,
7. Strategia *Strangle*.

Prócz tego można wycenić barierowe opcje europejskie z barierą typu europejskiego oraz wyliczyć ich greckie parametry:

1. Opcje z barierą europejską *down-and-out*,
2. Opcje z barierą europejską *up-and-in*,
3. Opcje z barierą europejską *up-and-out*,
4. Opcje z barierą europejską *down-and-in*.

Wyceniane są także opcje europejskie i amerykańskie z barierą amerykańską monitorowaną w następujące sposoby: ciągle monitorowanie przez czas życia opcji, ciągle monitorowanie w oknie czasowym, dyskretne monitorowanie przez czas życia opcji, dyskretne monitorowanie w oknie czasowym. Dla wszystkich tych opcji obliczana jest tylko cena opcji. Greckie parametry nie są obliczane. Wycena jest zrobiona dla następujących opcji europejskich i amerykańskich:

1. Opcje *down-and-out*,
2. Opcje *up-and-in*,
3. Opcje *up-and-out*,
4. Opcje *down-and-in*,
5. Opcje *double-knock-out*,
6. Opcje *up-in-up-out*,
7. Opcje *down-in-down-out*.



## 6 Dokumentacja

W tej części przedstawimy opisy zaimplementowanych funkcji wyliczających ceny poszczególnych opcji amerykańskich i europejskich w modelu Longstaffa-Schwartz. Dokumentacja opisuje program główny zawarty w pliku `SO_MC_barrier.m` składający się z funkcji głównej `calculate_MC`, która oblicza ceny opcji amerykańskich i europejskich przez wywołanie potrzebnych funkcji pomocniczych oraz funkcji pomocniczych. Obliczenia dla opcji europejskich są realizowane przez funkcję `EuropeanOptionPrice` a dla opcji amerykańskich, przez funkcję `LSModel`.

### 6.1 Opisy funkcji

#### 6.1.1 `brownianBridge`

1. **Opis:** Funkcja pomocnicza obliczająca wartości odpowiednich mostów Browna dla danego odcinka czasu.
2. **Input:** `prev_assetPrice` - cena aktywa bazowego w poprzednim węźle; `assetPrice`; `B`; `sigma`; `dt_i` - długość odcinka czasu pomiędzy poprzednim a kolejnym węzłem; `barrier_type`
3. **Output:** `bB` - dla opcji jednobarierowych jest to przybliżona wartość prawdopodobieństwa nieprzekroczenia bariery na odcinku czasu pomiędzy dwoma węzłami. Dla opcji typu `double-knock-out` i `double-no-touch` jest to analogiczne przybliżone prawdopodobieństwo nieprzekroczenia żadnej z barier. W przypadku opcji typu `knock-in-knock-out` zwracany jest wektor o dwóch współrzędnych - pierwsza mówi, z jakim prawdopodobieństwem w danym odcinku czasu przekroczone zostanie bariera aktywująca i jednocześnie nie zostanie przekroczone bariera dezaktywująca (dla opcji, które nie były jeszcze aktywne). Druga współrzędna wykorzystywana jest dla opcji aktywnej i oznacza prawdopodobieństwo, że na danym odcinku czasu nie zostanie ona zdezaktywowana.

#### 6.1.2 `generateAssetBase_barrier`

1. **Opis:** Funkcja pomocnicza, która w celu optymalizacji kodu oblicza jednocześnie kilka istotnych, wykorzystywanych w późniejszych etapach pracy programu zmiennych.
2. **Input:** `S0`; `B`; `discountDelta`; `discountDeltaF`; `sigma`; `number_of_paths`; `grid`; `dt`; `monitoring_type`; `barrier_type`; `monitoringGrid`; `nodeWindow_start`; `nodeWindow_end`
3. **Output:** `RNGSeeds` - macierz ziarna pozwalająca podczas powtórnego losowania otrzymać identyczne próbki z rozkładu normalnego, wykorzystywane do symulacji ścieżek cen aktywa bazowego; `stopping_time` - wektor momentów zatrzymania, tj. numerów węzłów na siatce czasu, dla których na poszczególnych

ścieżkach po raz pierwszy przekroczone bariere, zmodyfikowany przez porównanie obliczonych mostów Browna z wartościami próbki rozkładu jednostajnego;  $ST$  - wektor uzyskanych w wyniku symulacji cen aktywa bazowego w chwili zapadnięcia opcji,  $z$  - suma wylosowanych wektorów z rozkładu normalnego (obecna we wzorze na cenę instrumentu bazowego w chwili  $T$ ).

### 6.1.3 generateAssetBase\_barrier\_eur

1. **Opis:** Funkcja pomocnicza, która w celu optymalizacji kodu oblicza jednocześnie kilka istotnych zmiennych, wykorzystywanych w późniejszych etapach obliczeń dla opcji europejskich.
2. **Input:**  $S0$ ;  $B$ ;  $discountDelta$ ;  $discountDeltaF$ ;  $sigma$ ;  $number\_of\_paths$ ;  $grid$ ;  $dt$ ;  $monitoring\_type$ ;  $barrier\_type$ ;  $monitoringGrid$ ;  $nodeWindow\_start$ ;  $nodeWindow\_end$
3. **Output:**  $stopping\_time\_eur$  - wektor momentów zatrzymania, tj. numerów węzłów na siatce czasu, dla których na poszczególnych ścieżkach po raz pierwszy przekroczone bariere;  $ST$  - wektor uzyskanych w wyniku symulacji cen aktywa bazowego w chwili zapadnięcia opcji;  $bridgewhole$  - iloczyn prawdopodobieństw nieprzekroczenia bariery na odcinkach czasu pomiędzy wszystkimi węzłami (potrzebny do odliczenia ceny opcji europejskiej).

### 6.1.4 updateStoping\_time

1. **Opis:** Funkcja pomocnicza, która w  $i$ -tym węźle na siatce czasu sprawdza, na których ścieżkach została przekroczone bariere. Wówczas wybiera te ścieżki  $i$  jeśli do tej pory nie został dla nich wyznaczony moment przekroczenia bariery, przypisuje mu wartość  $i$ . W przypadku ciągłego monitorowania bariery zostaje za każdym razem uwzględniony most Browna - prawdopodobieństwo nieprzekroczenia bariery na odcinku  $[i - 1, i]$  - jego wartość porównywana jest z próbka z rozkładu jednostajnego. Jeśli jest większa - uznaje się, że bariere nie została przekroczone.
2. **Input:**  $stopping\_time$ ;  $i$ ;  $prev\_assetPrice$ ;  $assetPrice$ ;  $B$ ;  $sigma$ ;  $Mx$ ;  $barrier\_type$ ;  $condition$  - zmienna logiczna mówiąca, czy bariere została przekroczone;  $dt$ ;  $monitoring\_type$ ;  $nodeWindow\_start$ ;  $nodeWindow\_end$
3. **Output:**  $stopping\_timeNew$  - zaktualizowany (na moment  $i$ -ty) wektor momentów zatrzymania. Uwaga: dla opcji knock-in-knock-out  $stopping\_time$  jest macierzą o dwóch kolumnach: w pierwszej z nich znajduje się moment przekroczenia bariery aktywującej opcję, w drugiej - dezaktywującej.

### 6.1.5 updateStoping\_time\_eur

1. **Opis:** Funkcja pomocnicza, która w  $i$ -tym węźle na siatce czasu sprawdza, na których ścieżkach została przekroczone bariere. Wówczas wybiera te ścieżki  $i$  jeśli do tej pory nie został dla nich wyznaczony moment przekroczenia bariery,

przypisuje mu wartość  $i$ . Przy aktualizowaniu momentów stopu nie jest uwzględniany Brownian bridge, jednak zamiast tego wczytywany wektor `bridgewhole` (iloczyn mostów Browna dla odcinków między dotychczasowymi węzłami) zostaje przemnożony o wartość kolejnego mostu.

2. **Input:** `stopping_time_eur`;  $i$ ; `prev_assetPrice`; `assetPrice`;  $B$ ;  $\sigma$ ;  $Mx$ ; `barrier_type`; `condition`;  $dt$ ; `monitoring_type`; `nodeWindow_start`; `nodeWindow_end`; `bridgewhole`.
3. **Output:** `stopping_time_eurNew`; `bridgewholeNew` - zaktualizowane wartości `updateStopping_time_eur` i `bridgewhole`.

### 6.1.6 EuropeanOptionPrice

1. **Opis:** Funkcja obliczająca cenę europejskiej opcji barierowej. To jest funkcja główna, która wywołuje funkcję pomocniczą `EuropeanOptionPrice_` wykonującą właściwe obliczenia. W tej funkcji obliczenia dla opcji knock-in-knock-out są sprowadzane do kombinacji liniowej prostszych opcji.
2. **Input:**  $S0$ ;  $K$ ;  $B$ ; `name`; `issue_date`; `expire_date`; `grid`; `discountCurve`; `discountDelta`; `discountCurveF`; `discountDeltaF`;  $tgrid$ ;  $dt$ ; `discountOSO`; `discountPPO`; `monitoring_type`; `monitoringGrid`; `nodeWindow_start`; `nodeWindow_end`; `barrier_type`.
3. **Output:**  $y$  - wektor kolumnowy o 7 wierszach zawierający cenę europejskiej opcji barierowej oraz greckie parametry (dla opcji z barierą amerykańską wszystkie greckie parametry mają wartość zero).

### 6.1.7 EuropeanOptionPrice\_

1. **Opis:** Właściwa funkcja obliczająca cenę europejskiej opcji barierowej. W tej funkcji robione są osobne obliczenia dla opcji z barierą obserwowaną w sposób ciągły, barierą obserwowaną na odcinku czasu (`Window`), barierą obserwowaną w dyskretnych momentach czasu oraz opcji, dla których brak bariery lub bariera jest typu europejskiego (obserwowana tylko w momencie zapadalności opcji).
2. **Input:**  $S0$ ;  $K$ ;  $B$ ; `name`; `issue_date`; `expire_date`; `grid`; `discountCurve`; `discountDelta`; `discountCurveF`; `discountDeltaF`;  $tgrid$ ;  $dt$ ; `discountOSO`; `discountPPO`; `monitoring_type`; `monitoringGrid`; `nodeWindow_start`; `nodeWindow_end`; `barrier_type`.
3. **Output:**  $y$  - wektor kolumnowy o 7 wierszach zawierający cenę europejskiej opcji barierowej oraz greckie parametry (dla opcji z barierą amerykańską wszystkie greckie parametry mają wartość zero).

### 6.1.8 LSMmodel

1. **Opis:** Funkcja oblicza cenę oraz greckie parametry dla opcji amerykańskiej wg modelu Longstaffa-Schwartz dla wybranego rodzaju opcji barierowych. Obliczane są także wartości dla opcji, dla których brak bariery. Funkcja potrafi także wyliczyć wartości dla opcji z barierą typu europejskiego (obserwowaną tylko w momencie zapadalności opcji), ale takie opcje jako opcje amerykańskie nie mają sensu finansowego.
2. **Input:** *S0, K, B, name, issue\_date, expire\_date, grid, discountCurve, discountDelta, discountCurveF, discountDeltaF, workingGrid, tgrid, dt, discountOSO, discountPPO, monitoring\_type, monitoringGrid, nodeWindow\_start, nodeWindow\_end, order, barrier\_type, OLS\_TLS* - wybór metody ('OLS' albo 'TLS') obliczania warunkowej wartości oczekiwanej przyszłej wypłaty, domyślnie 'OLS'.
3. **Output:** *textity* - wektor kolumnowy o 7 wierszach zawierający cenę amerykańskiej opcji barierowej oraz greckie parametry (dla opcji z barierą amerykańską wszystkie greckie parametry mają wartość zero).

### 6.1.9 CheckIfActive

1. **Opis:** Funkcja pomocnicza sprawdzająca czy w danej chwili *i* opcja barierowa jest aktywna.
2. **Input:** *stopping\_time, i, barrier\_type*
3. **Output:** *condition* - wektor logiczny tej samej długości co *stopping\_time*, w którym True oznacza, że opcja barierowa aktywowała się najpóźniej w chwili *i* (odpowiednio: nie dezaktywowała się wcześniej lub w chwili *i*).

### 6.1.10 CheckIfHappened

1. **Opis:** Funkcja pomocnicza sprawdzająca czy w danej chwili *i* cena instrumentu bazowego przekroczyła barierę.
2. **Input:** *assetPrice, B, barrier\_type*
3. **Output:** *condition* - wektor logiczny tej samej długości co *assetPrice*, w którym True oznacza, że cena instrumentu podstawowego przekroczyła barierę (odpowiednio: jedną z barier dla opcji double-knock-out). Dla opcji knock-in-knock-out tworzony jest wektor o dwóch kolumnach (każda o tej samej długości co *assetPrice*). Pierwsza kolumna podaje warunek przekroczenia bariery dolnej, a druga bariery górnej.

### 6.1.11 rozszerzony ExpectedValueCoef

1. **Opis:** Funkcja pomocnicza obliczająca współczynnik regresji. Ta funkcja nie jest wykorzystywana w obecnej wersji programu.

2. **Input:** *inTheMoney, assetPrice, order, optionPrice, OLS\_TLS*.
3. **Output:** *Coef* - współczynnik regresji przeprowadzonej metodą Ordinary Least Squares (zgodnie z oryginalnym algorytmem Longstaffa-Schwartz) albo Total Least Squares (zgodnie ze zmodyfikowanym algorytmem Longstaffa-Schwartz (TLSQM)).

#### 6.1.12 rozszerzony CalculateExpectedValue

1. **Opis:** Funkcja pomocnicza obliczająca wartość oczekiwaną opcji z regresji.
2. **Input:** *inTheMoney, assetPrice, order, optionPrice, OLS\_TLS*.
3. **Output:** *expectedValue* - wartość oczekiwana obliczona metodą OLS lub TLS.

#### 6.1.13 LaguerrePolyMatrix

1. **Opis :** Funkcja pomocnicza obliczająca macierz wielomianów Laguerre'a dla zadanego stopnia *order*.
2. **Input :** wektor *X*, *order*,
3. **Output :** wynikiem jest macierz *PolyMatrix*, gdzie w kolumnach znajdują się wartości dla każdego wielomianu Laguerre'a stopnia od 0 do *order* a argumentem każdego z nich jest wektor *X*;

#### 6.1.14 EurGreeks

1. **Opis:** Funkcja wylicza greckie parametry – deltę oraz gammę dla barierowych opcji europejskich. Ta funkcja nie jest wykorzystywana, ponieważ zawiera błędy.
2. **Input:** *S0, RNGSeeds, K, B, name, dt, tgrid, T, barrier\_type, monitoring\_type, discountCurve, discountDelta, discountCurveF, discountDeltaF, sigma, monitoringGrid, grid, Mx*.
3. **Output:** *Delta, Gamma*.

#### 6.1.15 dXcalculate

1. **Opis:** Funkcja pomocnicza wyliczająca przyrost logarytmu ceny instrumentu podstawowego dla danej chwili czasu.
2. **Input:** *RNGSeed, dt, disFacDelta, disFacDeltaF, Mx, sigma*
3. **Output:** *dX*

### 6.1.16 Psi

1. **Opis:** Funkcja klasy  $C^\infty$  względem  $Y$ , przyjmująca wartość 1 dla  $Y \leq a/2$ , zaś 0 dla  $Y > a$ .
2. **Input:**  $Y, a$
3. **Output:**  $PsiValue$

### 6.1.17 CalculateMC

1. **Opis:** Funkcja obliczająca ceny opcji europejskiej lub amerykańskiej dla różnych strategii. To jest funkcja główna wywołująca `CalculateMC_`, która jest funkcją pomocniczą wykonującą właściwe obliczenia. W tej funkcji obliczenia dla portfeli są sprowadzane do kombinacji liniowej prostszych opcji.
2. **Input:**  $name; K, B, issue\_date, expire\_date, DF\_QUOT, DF\_BASE, exercise\_type$  – 'American' dla obliczenia wartości opcji amerykańskiej lub 'European' dla opcji europejskiej,  $monitoring\_type, window\_start\_date, window\_end\_date, MF$  – częstotliwość (w dniach) dyskretnego monitorowania bariery,  $OLS\_TLS$ .
3. **Output:**  $y$  - wektor kolumnowy o 7 wierszach zawierający cenę opcji barierowej oraz greckie parametry dla opcji amerykańskiej lub europejskiej w zależności od wybranego  $exercise\_type$ .

### 6.1.18 CalculateMC\_

1. **Opis:** Właściwa funkcja obliczająca ceny opcji europejskiej lub amerykańskiej dla różnych strategii. T
2. **Input:**  $name; K, B, issue\_date, expire\_date, DF\_QUOT, DF\_BASE, exercise\_type$  – 'American' dla obliczenia wartości opcji amerykańskiej lub 'European' dla opcji europejskiej,  $monitoring\_type, window\_start\_date, window\_end\_date, MF$  – częstotliwość (w dniach) dyskretnego monitorowania bariery,  $OLS\_TLS$ .
3. **Output:**  $y$  - wektor kolumnowy o 7 wierszach zawierający cenę opcji barierowej oraz greckie parametry dla opcji amerykańskiej lub europejskiej w zależności od wybranego  $exercise\_type$ .

### 6.1.19 log\_two\_cosh

1. **Opis:** Wylicza  $\log(2*\cosh(x))$  radząc sobie z faktem że  $\cosh(x) = \text{Inf}$  dla  $\text{abs}(x) > 700$
2. **Input:**  $x$
3. **Output:**  $y$

#### 6.1.20 `generateSmoothFunc_all`

1. **Opis:** Zwraca funkcje służące do wyceny greckich parametrów. Ef - wypłata z put lub call, Bf - czynnik barriery - przyjmuje wartości z przedziału [0, 1] (np. Heaviside), B2f - (jedynie dla double-out) czynnik drugiej barriery. Prefix 'd' oznacza pochodną (ze względu na S0), 'd2' drugą pochodną. Dla opcji dwubarrierowych innych niż double-knock-out w Bf jest zawarta poprawka na obie barriery.
2. **Input:** *barrier\_type, smooth\_parameter, K, B, callput*
3. **Output:** *Bf, dBf, d2Bf, Ef, dEf, d2Ef, B2f, dB2f, d2B2f*

#### 6.1.21 `smoothed_price_and_greeks`

1. **Opis:** Oblicza cenę, delte i gamme, dobierając dla nich właściwe parametry epsilon.
2. **Input:** *assetMin, assetMax, assetPrice, barrier\_type, S0, K, B, callput, discountCurve, discountCurveF*
3. **Output:** *price, delt, gamm, deviations, smooth\_parameters*

#### 6.1.22 `smoothed_price_and_delta`

1. **Opis:** dobiera parametr wygładzający dla delty i zwraca deltę (oraz cenę przy tym samym parametrze)
2. **Input:** *assetMin, assetMax, assetPrice, barrier\_type, S0, K, B, callput, discountCurve, discountCurveF*
3. **Output:** *price, delt, deviation\_price, deviation\_delta, smooth\_parameter*

#### 6.1.23 `smoothed_gamma`

1. **Opis:** dobiera parametr wygładzający dla gammy i zwraca gamme
2. **Input:** *assetMin, assetMax, assetPrice, barrier\_type, S0, K, B, callput, discountCurve, discountCurveF*
3. **Output:** *gamm, deviation, smooth\_parameter*

#### 6.1.24 `smoothed_price_and_greeks_`

1. **Opis:** oblicza greki przy zadanym parametrze wygładzającym. Parametry *countPrice*, *countDelta*, *countGamma* wyznaczają które wartości mają faktycznie być obliczone (domyślnie wszystkie)
2. **Input:** *assetMin, assetMax, assetPrice, barrier\_type, S0, K, B, callput, smooth\_parameter, discountCurve, discountCurveF, countPrice=1, countDelta=1, countGamma=1*

3. **Output:** *price, delt, gamm, deviations*

#### 6.1.25 barrier\_opt\_price\_and\_greeks\_from\_asset\_double\_out

1. **Opis:** wykonuje rzeczywiste obliczenia dla opcji double-knock-out
2. **Input:** *assetMin, assetMax, assetPrice, barrier\_type, S0, K, B, callput, smooth\_parameter, discountCurve, discountCurveF, countPrice=1, countDelta=1, countGamma=1*
3. **Output:** *price, delt, gamm, deviations*

#### 6.1.26 barrier\_opt\_price\_and\_greeks\_from\_asset

1. **Opis:** wykonuje rzeczywiste obliczenia dla opcji różnych od double-knock-out
2. **Input:** *assetMin, assetMax, assetPrice, barrier\_type, S0, K, B, callput, smooth\_parameter, discountCurve, discountCurveF, countPrice=1, countDelta=1, countGamma=1*
3. **Output:** *price, delt, gamm, deviations*

## Literatura

- [1] V. Bally – An elementary introduction to Malliavin calculus, *INRIA Rapport de recherche*, No. 4718 (2003)
- [2] Raymond H. Chan, Chi-Yan Wong, Kit-Ming Yeung – Pricing Multi-asset American-Style Options by Memory Reduction Monte Carlo Methods, *Applied Mathematics and Computation*, **179** (2006), 535-544.
- [3] E. Gobet – Advanced Monte Carlo methods for barrier and related exotic options W; *Handbook of Numerical Analysis; Special Volume: Mathematical Modelling and Numerical Methods in Finance*, Bensoussan A., Zhang Q. (Eds.), Elsevier 2009, pp. 497–528.
- [4] E. Gobet – Revisiting the Greeks for European and American options, W: *Stochastic Processes and Applications to Mathematical Finance* Akahori, J., Ogawa, S., Watanabe, S. (Eds.), World Scientific 2004, pp. 53–71.
- [5] E. Gobet, A. Kohatsu-Higa – Computation of Greeks for barrier and lookback options using Malliavin calculus, *Electronic Communications in Probability*, **8** (2003), 51–62.
- [6] Francis A. Longstaff, Eduardo S. Schwartz – Valuing American Options by Simulation: A Simple Least-Squares Approach, *The Review of Financial Studies*, **14** (2001), 113-147.



- [7] K. Moon – Efficient Monte Carlo algorithm for pricing barrier options, *Communications of the Korean Mathematical Society*, **23** (2008), 285–294.
- [8] P.V. Shevchenko – Addressing the bias in Monte Carlo pricing of multi-asset options with multiple barriers through discrete sampling, *Computational Finance*, **6**(3) (2003), 1–20.
- [9] Howard Thom – *Longstaff Schwartz Pricing of Bermudan Options and their Greeks*. University of Oxford, Institute for Mathematical Finance, 2009.
- [10] Bogusław Wróblewski – *Raport i Dokumentacja: Obliczanie cen i parametrów greckich opcji walutowych w modelu Blacka-Scholesa*. Uniwersytet Warszawski, Wydział Matematyki, Informatyki i Mechaniki, 2011.
- [11] Anna Zaremba – *Metoda Longstaffa-Schwartza wyceny opcji amerykańskich*. Uniwersytet Warszawski, Wydział Matematyki, Informatyki i Mechaniki, 2005.
- [12] L.Zhang, W. Xu, X. Shi – A Modified Least-Squares Simulation Approach to Value AMERICAN Barrier Options, *Computational Economics*, **44** (2014), 489–506.
- [13] Basileios Papatheodorou – *Enhanced Monte Carlo Methods for Pricing and Hedging Exotic Options*. University of Oxford, 2005.