Dokumentacja

Wycena opcji za pomocą uogólnionego modelu Hestona i modeli pokrewnych

Mikołaj Bińkowski
Wiktor Gromniak
Karol Klimas
Łukasz Rajkowski
dodatki AP

Spis treści

1 Wstęp teoretyczny

Program implementuje symulacje Monte-Carlo dla szeregu modeli Stochastic Volatility.

We wszystkich modelach współczynnik dryfu aktywa bazowego modelowany jest za pomocą rzeczywistych krzywych dyskontowych; stąd parametr dryfu nie jest wprowadzany bezpośrednio przez użytkownika, lecz wyprowadzany jest z danych rynkowych z użyciem istniejących modułów projektu.

1.1 Rozpatrywane modele stochastycznej zmienności

1.1.1 Model Hestona

Model Hestona zadany jest przez układ stochastycznych równań różniczkowych

$$dS_t = r(t)S_t dt + \sqrt{V_t} S_t dW_t^S, \tag{1}$$
$$dV_t = \kappa(\theta - V_t)dt + \omega\sqrt{V_t} dW_t^V, \tag{2}$$

gdzie W^S, W^V są standardowymi procesami Wienera o współczynniku korelacji $\rho \in [-1, 1]$, tj. $d\langle W^S, W^V\rangle_t = \rho dt$.

1.1.2 Model Stein&Stein

Model Stein&Stein [6] zadany jest przez układ stochastycznych równań różniczkowych

$$dS_t = r(t)S_t dt + V_t S_t dW_t^S, \tag{3}$$
$$dV_t = \kappa(\theta - V_t)dt + \omega dW_t^V, \tag{4}$$

gdzie W^σ, W^V są standardowymi procesami Wienera o współczynniku korelacji ρ.

1.1.3 Model SABR

Model SABR zadany jest przez układ stochastycznych równań różniczkowych

$$dS_t = r(t)S_t dt + V_t S_t^\beta dW_t^S, \tag{5}$$
$$dV_t = \kappa(\theta - V_t)dt + \omega V_t dW_t^V, \tag{6}$$

gdzie W^S, W^σ są standardowymi procesami Wienera o współczynniku korelacji ρ.

1.1.4 Model Hull&White

Model Hull&White [3] zadany jest przez układ stochastycznych równań różniczkowych

$$dS_t = r(t)S_t dt + \sqrt{V_t} S_t dW_t^S, \tag{7}$$
$$dV_t = \kappa(\theta - V_t)dt + \omega V_t^\alpha dW_t^V, \tag{8}$$

gdzie W^S, W^V są standardowymi procesami Wienera o współczynniku korelacji ρ.

1.1.5 Uogólniony model Hestona

Uogólniony model Hestona zadany jest przez układ stochastycznych równań różniczkowych

$$
\begin{aligned}
dS_t &= r(t)S_t dt + \lambda\sqrt{V_t}S_t^\beta dW_t^S, && (9)\\
dV_t &= \kappa(\theta - V_t)dt + \omega V_t^\alpha dW_t^V, && (10)
\end{aligned}
$$

gdzie W^S, W^V są standardowymi procesami Wienera o współczynniku korelacji ρ. W szczególności model Hull&White może być traktowany jako wariant uogólnionego modelu Hestona.

1.1.6 Model CEV

Model CEV jest modelem lokalnej stochastycznej zmienności. Model jest opisany przez jedno stochastyczne równanie różniczkowe

$$dS_t = r_t S_t dt + \sigma_0 S_t^\beta dW_t, \tag{11}$$

gdzie W_t jest standardowym procesem Wienera. W modelu CEV stochastyczna zmienność zależy od ceny aktywa i wynosi $\sigma_0 S_t^{\beta-1}$.

1.2 Symulacja Monte Carlo

Symulacji ścieżek zmienności dokonano z wykorzystaniem Implicit Milstein Method, która wykazuje się przewagą pod względem szybkości zbieżności, co zbadano w [5].

W przypadku ceny instrumentu bazowego dokonano symulacji z wykorzystaniem metody Eulera zastosowanej do logarytmu ceny instrumentu (patrz [5] wzór (22)), za wyjątkiem przypadku modelu Hestona, gdzie zastosowano efektywniejszy IJK scheme pochodzący z pracy [4] z modyfikacjami zaproponowanymi w [1]. Pierwszy z poniższych wzorów przedstawia schemat losowania wartości procesu zmienności V_t:

$$V_{t+1} = \frac{1}{1+\kappa d_t}\Big(V_t + \kappa\theta d_t + \omega V_t^\alpha dW_{V,t} + \frac{1}{2}\omega^2\alpha V_t^{2\alpha-1}(dW_{V,t}^2 - d_t)\Big) \tag{12}$$

Drugi wzór odpowiada generowaniu procesu cen S_t wyłącznie dla modelu Hestona.

$$
\begin{aligned}
\ln S_{t+1} =& \ln S_t + \ln\frac{\Delta_{f,t}}{\Delta_t} - \frac{1}{4}\lambda^2 d_t(V_t + V_{t+1}) + \rho\lambda\sqrt{V_t}dW_{V,t} + \\
& \frac{1}{2}(\sqrt{V_t} + \sqrt{V_{t+1}})(dW_{S,t} - \rho dW_{V,t}) + \frac{1}{4}\lambda\omega\rho\sqrt{V_t}^{\alpha-\frac{1}{2}}\Big(dW_{V,t}^2 - d_t\Big)
\end{aligned} \tag{13}
$$

Ostatnie równanie opisuje pozostałe przypadki generowania procesu cen.

$$\ln S_{t+1} = \ln S_t + \ln\frac{\Delta_{f,t}}{\Delta_t} - \frac{1}{2}\lambda^2 S_t^{2\beta-1}|V_t|^{ss+1}d_t + \lambda|V_t|^{\frac{ss+1}{2}}S_t^{\beta-1}dW_{S,t} \tag{14}$$

W symulacjach zastosowano metodę zmiennych antytetycznych, co pozwala na zmniejszenie wariancji (por. [13]).

Celem obliczenia ceny opcji amerykańskiej, zastosowano metodę Longstaffa – Schwarza (LSM). Tę samą metodę wykorzystywano również do obliczeń ceny opcji amerykańskich w modelu Blacka-Scholesa (por. [8]), w tej samej dokumentacji można znaleźć dokładny opis algorytmu. Poniżej wymieniono tylko różnice, które wynikają ze specyfiki modeli SV oraz zastosowanych metod symulacji trajektorii.

1. Proces $X_t = (S_t, V_t)$ jest procesem wielowymiarowym, dlatego konieczne było wykorzystanie wielomianów wielu (dwóch) zmiennych. Wybrano standardową bazę, czyli wielomiany postaci $\sum_{k=1}^{n} C_k x^{n-k} y^k$, tak jak zaproponowano w pracy [2].

2. Wybrane kroki schematów różnicowych jako funkcje X_{t+1} zmiennej X_t nie są odwracalne (poza niektórymi przypadkami). Oznacza to, że przy generowaniu ścieżek algorytm LSM zachowanie tylko wartości procesu w ostatnim kroku czasowym T wymagałoby generowanie kroków wczeniejszych $T-1, T-2, \ldots$ od wartości początkowych. Wydłużyłoby to znacznie czas działania algorytmu. Z drugiej strony zapamiętywanie całe macierzy procesu X (wszystkich scieżek) powoduje ograniczenia pamięciowe (opisane w [8]). Wybrano kompromis pomiędzy złożonocią obliczeniową implikującą czas obliczeń, a liczbą symulacji, która wpływa na dokładność oszacowań.

Zamiast całych ścieżek zapamiętywane są wygenerowane wartości wygenerowanego procesu tylko dla części indeksów czasowych, co pozwala na odtworzenie symulacji dla dowolnego indeksu czasowego. Precyzyjniej rzecz ujmując, tworzone są specjalne tablice `ColsToRemS` i `ColsToRemV`, przechowujące kolumny $S_{i \cdot blockWidth+1}$, $V_{i \cdot blockWidth+1}$ odpowiednio dla $i = 0, 1, \ldots, (nOfBlocks - 1)$. Dzielimy w ten sposób uzyskane wartości wektora symulacji (S, V) w kolejnych krokach czasowych na **bloki**, w których wartości możemy odtworzyć posługując się „początkową” wartością $S_{i \cdot blockWidth+1}, V_{i \cdot blockWidth+1}$. W tej sytuacji, celem wykonania „kroku wstecz” przy stosowaniu metody Longstaffa – Schwarza, odtwarzamy najpierw ostatni blok (i obliczamy wartości opcji w obejmowanych przez ów blok krokach czasowych), następnie przedostatni i procedurę tę kontynuujemy aż do uzyskania wartości w pierwszym bloku, więc również w pierwszym kroku czasowym. Warto zwrócić uwagę że, ze względu na odtwarzanie całych bloków symulacji, obliczenia potrzebne do uzyskania wykorzystanych wcześniej wartości symulacji (S, V) wykonywane są tylko raz, zatem w tym względzie zaproponowane rozwiązanie nie ustępuje metodologii wykorzystanej w modelu Blacka Scholesa. Należy jednak zaznaczyć, że problem wykorzystanej pamięci nie zostaje w ten sposób zupełnie rozwiązany – z jednej strony potrzebujemy pamięci na zapamiętanie kolumn tabeli `ColsToRem`, z drugiej strony potrzebujemy zapamiętać odtwarzane bloki. Musi więc zatem zachodzić

$$\begin{aligned}
&\ln(\#symulacji \times \#zapamietanych + \#symulacji \times szerBloku) = \\
&\qquad = \ln(\#symulacji) + \ln(\#zapamietanych + szerBloku) \leq \ln(ogrPamieci)
\end{aligned} \tag{15}$$

Iloczyn liczby zapamiętanych kroków oraz szerokości bloku odpowiada liczbie kroków czasowych; nietrudno przekonać się, że jeśli spełniona jest nierówność

$$\ln(\#symulacji) + \frac{1}{2}\ln(\#krokowCzasowych) \leq \ln(ogrPamieci), \tag{16}$$

można w taki sposób dobrać parametry $\#zapamietanych$ oraz $szerBloku$, aby spełniona była nierówność (15). Opisywany program dobiera te parametry w taki sposób, aby minimalizować szerokość otrzymywanych bloków.

1.3 Parametry greckie

W programie są wyznaczane także parametry greckie. Ogólna metoda ich obliczania (metoda różnic skończonych) obarczona jest dużym błędem (por. [13]. Do zastosowania innych metod konieczna jest przynajmniej częściowa znajomość wzorów na cenę instrumentu bazowego, konkretnie zależności ceny od analizowanych parametrów modelu. Dlatego parametry greckie wyliczane są tylko dla modelu Hestona. Zastosowane zostały wzory z pracy [12].

- Δ czyli pochodna ceny opcji po cenie instrumentu podstawowego S_0. Zasosowano estymator metody różniczkowania po trajektoriach (PW):

  $$\frac{\partial \Pi_0}{\partial S_0} = \mathbb{E}e^{-r\tau}\frac{\partial f(S_\tau)}{\partial S_\tau}\frac{\partial S_\tau}{\partial S_0} = \mathbb{E}e^{-r\tau}\frac{\partial f(S_\tau)}{\partial S_\tau}\frac{S_\tau}{S_0}$$

  Zastosowanie metody (PW) oznacza, że może ona być stosowana tylko do ciągłych wypłat.

- ρ czyli pochodna ceny opcji po stopie procentowej. Oznacza to równoległe przesunięcie krzywej stóp procentowych w górę lub w dół. Ponownie zastosowano estymator (PW) postaci:

  $$\frac{\partial \Pi_0}{\partial r} = \mathbb{E}e^{-r\tau}\frac{\partial f(S_\tau)}{\partial S_\tau}\frac{\partial S_\tau}{\partial r} = \mathbb{E}e^{-r\tau}\tau\Big(\frac{\partial f(S_\tau)}{\partial S_\tau}S_\tau - f(S_\tau)\Big)$$

- Γ czyli druga pochodna ceny opcji po cenie instrumentu podstawowego S_0. Niemożliwe było zastosowanie estymatora metody różniczkowania po trajektoriach (PW), ponieważ pochodna wypłaty nie jest ciągła, dlatego zastosowano mieszany estymator (LR-PW). Metoda ilorazu wiarygodności zakłada różniczkowanie gęstości S_τ po wybranym parametrze. W praktyce oznacza to mnożenie wypłaty przez pochodną logarytmu gęstości po wybranym parametrze.

  $$\frac{\partial^2 \Pi_0}{\partial S_0^2} = \mathbb{E}e^{-r\tau}\frac{1}{S_0}\frac{Z}{S_0\sigma\sqrt{\tau}}\Big(\frac{\partial f(S_\tau)}{\partial S_\tau}S_\tau - f(S_\tau)\Big)$$

  Z jest zmienną losową o standardowym rozkładzie normalnym, która służyła do wygenerowania trajektorii S_τ, σ jest natomiast wartością średnią z procesu zmienności V na przedziale $(0,\tau)$.

2 Zależności

Program korzysta z funkcji kalendarzowych [11] oraz funkcji tworzących krzywe dyskontowe [10]. Poza tym program korzysta z zaimplementowanych dla ustalonej opcji funkcji dyskontowych oraz funkcji stóp procentowych a także funkcji wypłaty zdefiniowanych w pliku pomocniczym `calculate_functions.m`. Ich dokumentacja znajduje się w [7]

3 Inicjalizacja - zmienne globalne

Parametry instrumentów będących przedmiotem wyceny, modeli wyceniających oraz metody Monte Carlo są zapisane w programie w postaci zmiennych globalnych. Część zmiennych globalnych nie dotyczy wyceny opcji metodą Monte Carlo w modelu stochastycznej zmienności. Nie opisujemy tych zmiennych, nawet jeśli są one w obliczeniach wykorzystywane. Poniżej opisane są tylko zmienne specyficzne dla danych modeli oraz metody numerycznej.

3.1 Paramety kalendarzowe, parametry wypłaty

Podane parametry są identyczne jak w module wyceny opcji w modelu Blacka-Scholes metodą Monte Carlo [8].

- `issue_date` – data podpisania kontraktu opcyjnego, string o formacie 'dd-mm-yyyy'.
- `expire_date` – data wygaśnięcia kontraktu opcyjnego, string o formacie 'dd-mmm- yyy'.
- `Mt` – ilość kroków czasowy ch w ciągu życia opcji, integer (to jest wartość przybliżona, patrz [7]).
- `Mx` – ilość ścieżek instrumentu bazowego w symulacji Monte Carlo, integer.
- `DDC` – string, typ konwencji liczenia czasu dla opcji (wartość identyczna jak `VOL_DCC`).
- `FC_DOM` – string, nazwa giełdy na której wyceniamy opcję.
- `DFQ_type_opt` – string, rodzaj czynnika dyskontowego dla waluty kwotowania, może być `'Ask'`, `'Bid'` lub `'Mean'`.
- `DFB_type_opt` – string, rodzaj czynnika dyskontowego dla waluty bazowej (dotyczy tylko rynku FX), może być `'Ask'`, `'Bid'` lub `'Mean'`.
- `checkWorkDay` – '1' - realizujemy opcję tylko w dni robocze, '0' realizujemy opcje również w dni nierobocze (dotyczy wyłącznie opcji amerykańskich).

3.2 Parametry opcji

- `S0` - cena początkowa aktywa bazowego.
- `K` - cena wykonania.
- `B` - bariera (dotyczy europejskich opcji barierowych)
- `name` - string, nazwa opcji (definiuje funkcję wypłaty), np. `'call'`, `'put'`, `'straddle'`, `'riskrev'`, `'butterfly'` itp. (patrz [7]).

3.3 Parametry modelu

Dla ustalonego modelu istotne są tylko parametry jego dotyczące.

1. Model Hestona - κ, θ, ω, ρ, V_0.
2. Model Stein&Stein - κ, θ, ω, ρ, V_0.
3. Model SABR - β, κ, θ, ω, ρ, V_0.
4. Model Hull&White - κ, θ, ω, ρ, α, V_0.
5. Uogólniony model Hestona - λ, β, κ, θ, ω, ρ, α, V_0.
6. Model CEV - σ_0, β. Ten model nie jest zaimplementowany w ramach funkcji obliczających ceny opcji w modelu stochastycznej zmienności. Jest on zaimplementowany w ramach modelu lokalnej zmienności.

Powyższym parametrom odpowiadają następujące zmienne globalne:

- `sv_model` - indentyfikator modelu, string; przyjmuje następujące wartości: `"Heston"`, `"SteinStein"`, `"SABR"`. Jeśli string `sv_model` ma inna wartość jest to interpretowane jako model `"geralizedHeston"` (w szczególności model *Hull-White* jest traktowany jako szczególny przypadek *generalized Heston*).
- `V0` - V_0, początkowa wartość wariancji (kwadratu zmienności) – dotyczy modelu Hestona, Hull&White oraz uogólnionego modelu Hestona, lub zmienności – dotyczy modeli Stein&Stein oraz SABR.
- `alpha` - parametr α, wykładnik w modelu *generalized Heston*.
- `Beta` - parametr β, wykładnik w modelach *SABR* i *generalized Heston*.
- `rho` - parametr ρ, korelacja procesów Wienera. Dotyczy modeli Hestona, Stein&Stein, SABR, Hull&White oraz uogólnionego modelu Hestona.
- `lambda` - parametr λ. Dotyczy uogólnionego modelu Hestona.
- `kappa` - parametr κ, szybkość powrotu zmienności (wariancji) do średniej.
- `theta` - parametr θ, długoterminowa średnia zmienności (wariancji).
- `omega` - parametr ω, "zmienność zmienności".

4 Funkcje

4.1 SV_MC_price

Główna funkcja obliczająca cenę opcji.

Argumenty funkcji:

- `S0, K, B, name, issue_date, expire_date, sv_model` – opisane wcześniej,
- `grid` - integer, rzeczywista liczba kroków czasowych symulacji,
- `discountCurve` - wektor czynników dyskontowych dla narastających delt czasu (waluta kwotowania),
- `discountDelta` - wektor czynników dyskontowych dla delt czasu (waluta kwotowania),
- `discountCurveF` - wektor czynników dyskontowych dla narastających delt czasu (waluta bazowa),
- `discountDeltaF` - wektor czynników dyskontowych dla delt czasu (waluta bazowa),
- `tgrid` - wektor delt czasu narastająco,
- `dt` - delta czasu między kolejnymi krokami czasowymi, jako ułamek roku,
- `discountOSO` - wektor czynników dyskontowych dla delt czasu między datą realizacji opcji a datą wypłaty świadczenia,
- `discountPPO` - czynnik dyskontowy dla delty czasu między datą wystawienia opcji a datą płatności premii opcyjnej.

Wynik funkcji:

- `EurOptionPrice` – cena opcji europejskiej,
- `AmOptionPrice` – cena opcji amerykańskiej,
- `eurGreeksValue` – wektor $[\Delta, \Gamma, \rho]$ współczynników greckich dla opcji europejskiej,
- `amGreeksValue` – wektor $[\Delta, \Gamma, \rho]$ współczynników greckich dla opcji amerykańskiej,
- `greeksNames` – wektor nazw obliczonych współczynników greckich

Komentarz:
Dla `name` $\neq$ `Heston` wektory `eurGreeksValue` i `amGreeksValue` są wektorami zerowymi.

4.2 getAssetPrice_SV

Funkcja pomocnicza obliczająca ścieżki wartości instrumentu bazowego oraz jego zmienności.

Argumenty funkcji:

- `S0, grid, discountDelta, discountDeltaF, dt, Mx` – opisane wcześniej,
- `ss` – zmienna pomocnicza zależna od wybranego modelu ('0' – modele symulujące stochastyczną wariancję, '1' – modele symulujące stochastyczną volatility).

Wynik funkcji:

- `assetPrice` – wektor $2\texttt{Mx} \times 1$ zawierający symulacje S_T.
- `volatility` – wektor $2\texttt{Mx} \times 1$ zawierający symulacje V_T,
- `mySeeds` – macierz $625 \times \texttt{grid}$, której kolumnami są ziarna użyte do symulacji ΔW^S i ΔW^V w kolejnych krokach czasowych,
- `colsToRem_V` – macierz $2\texttt{Mx} \times \texttt{nOfBlocks}$, której i-ta kolumna zawiera symulacje $S_{t_{(i-1)\texttt{blockWidth}+1}}$,
- `colsToRem_S` – macierz $2\texttt{Mx} \times \texttt{nOfBlocks}$, której i-ta kolumna zawiera symulacje $V_{t_{(i-1)\texttt{blockWidth}+1}}$,
- `int_V_eur` – wektor $2\texttt{Mx} \times 1$ symulacji zmiennej $\int_0^T V_t \mathrm{d}t$.

Komentarz:
Parametr `nOfBlocks` ustalany jest w funkcji na $10^6/(2\texttt{Mx})$ – jest to możliwie duża wartość, pozwalająca na zapamiętanie tablicy $2\texttt{Mx}\times\texttt{nOfBlocks}$. Parametr `blockWidth` wynosi $\lceil \texttt{grid}/\texttt{nOfBlocks} \rceil$.

4.3 calculate_SV_MC

Jedyna funkcja wywoływana przez końcowego użytkownika programu.

Argumenty funkcji:

- `name, issue_date, expire_date, SV_model` – opisane wcześniej,
- `DF_QUOT` – tablica czynników dyskontowych dla waluty kwotowania,
- `DF_BASE` – tablica czynników dyskontowych dla waluty bazowej,

Wynik funkcji:

- `y` – macierz 4×2, której kolumny zawierają cenę i wartości współczynników Δ, Γ, ρ odpowiednio opcji europejskiej i amerykańskiej.

Komentarz:
Funkcja zamienia obliczanie portfeli opcji waniliowych na kombinację opcji *call* oraz *put* oraz wywołuje właściwą funkcje realizującą obliczenia `calculate_SV_MC_`.

4.4 calculate_SV_MC_

Funkcja pomocnicza wywołująca funkcje realizujące obliczenia.

Argumenty funkcji:

- argumenty funkcji `calculate_SV_MC` opisane wcześniej,
- `K` - cena realizacji (wektor dla portfeli opcji waniliowych),
- `B` - bariera (może być wektor dla opcji dwu-barierowych),

Wynik funkcji:

- `y` – macierz 4×2, której kolumny zawierają cenę i wartości współczynników Δ, Γ, ρ odpowiednio opcji europejskiej i amerykańskiej.

4.5 LaguerrePolyMatrix

Funkcja pomocnicza, tworząca macierz wartości wielomianów dwóch zmiennych na współczynnikach zadanych wektorów. Nazwę zachowano z modelu BS w celu zachowania zgodności.

Argumenty funkcji:

- `X, Y` – wektory o tej samej długości n,
- `order` – stopień najwyższego zastosowanego wielomianu.

Wynik funkcji:

- `PolyMatrix` – macierz $n \times (\texttt{order} + 1)$, w której na miejscu (i, j) znajduje się $\sum_{k=0}^{j} \binom{n}{k} \frac{x_i^k y_i^{n-k}}{k!}$.

4.6 forwardStep

Funkcja pomocnicza, obliczająca $(V_{t_{j+1}}, S_{t_{j+1}})$ w zależności od (V_{t_j}, S_{t_j}), $\Delta t = t_{j+1} - t_j$ oraz parametrów modelu. Dodatkowo zapamiętywana jest wartość ziarna użytego do zasymulowania $\Delta W_{t_j}^S$ i $\Delta W_{t_j}^V$. Wartość tego ziarna można również wymusić przy użyciu parametru `forcedSeed` – dzięki temu możemy odtwarzać „bloki” symulacji S_t oraz V_t.

Argumenty funkcji:

- `V` – wektor $2\texttt{Mx} \times 1$ zawierający symulacje V_{t_j},
- `S` – wektor $2\texttt{Mx} \times 1$ zawierający symulacje S_{t_j},
- `Mx` – liczba symulacji,
- `dt_j` – różnica między rozpatrywanymi krokami czasowymi, $t_{j+1} - t_j$,

- `discountDelta_j` – czynnik dyskontowy dla delt czasu między datą realizacji opcji a datą wypłaty świadczenia w punkcie t_j,
- `discountDeltaF_j` – czynnik dyskontowy dla delty czasu między datą wystawienia opcji a datą płatności premii opcyjnej w punkcie t_j,
- `ss` – opisane wcześniej,
- `forcedSeed` – domyślnie ustawiony na `false` i wówczas funkcja sama wybiera sobie ziarno. Jeśli parametr zostanie ustawiony na wektor długości 625, funkcja posłuży się nim jako ziarnem do zasymulowania $\Delta W^S_{t_j}$ i $\Delta W^V_{t_j}$.

Wynik funkcji:

- `V2` – wektor 2`Mx` $\times$ 1 zawierający symulacje $V_{t_{j+1}}|(V_{t_j}, S_{t_j})$,
- `S2` – wektor 2`Mx` $\times$ 1 zawierający symulacje $S_{t_{j+1}}|(V_{t_j}, S_{t_j})$,
- `mySeed` – wektor 625×1, będący ziarnem użytym do symulacji $\Delta W^S_{t_j}$ i $\Delta W^V_{t_j}$.

5 Przykład

Wyniki symulacji dla następującej zawartości pliku `fx_terminal.m`

```
Depo_EMA = "+1";
monitoring_type = "Window";
FOR_DCC = "ACT/360";
sigma0 = 0.3;
numerical_scheme = "Implicit";
IsBarrierAsian = 0;
Depo_DCC = "ACT/360";
VOL_EMA = "+1";
broad_model = 'Stochastic Volatility';
issue_date = "24-aug-2009";
sigma_init = 0;
CALIBR_DCC = "@";
calc_method = 'Monte Carlo';
FOR_EMA = "+1";
BAS_CURR = "EUR";
vol_start_date = "24-aug-2009";
FRA_ref_ind = "EURIBOR";
CURR_DOM = "EUR";
name = 'put';
FRA_DCC = "ACT/360";
rho_init = 0;
Depo_BDA = "mfbd";
CURR_FRAFUT = "FRA";
OSO = 2;
```

```
theta = 0.04;
CALIBR_EMA = "@";
barrier = 1.3;
barrier_up = 1.6;
barrier_start_date = "24-aug-2009";
Mv = 200;
monitoring_freq = 7;
Mx = 1000;
RR_delta_val = 0.25;
Mt = 100;
V0_init = 0;
FRA_day_to_spot = 2;
number_of_paths = 0;
kappa_init = 0;
Depo_day_to_spot = 2;
lambda = 1;
time_interp = "linear on variance";
FX_rate = [1.4105, 1.4165];
DF_QUOT_type = "Mean";
DFQ_type_opt = "Mean";
DFB_type_opt = "Mean";
interp_method = "linear on df";
QUO_CURR = "USD";
vol_fx_rate_type = "Mean";
CURR_PAIR = "1";
FOR_BDA = "mfbd";
Beta = 0.7;
pip_val = 0.0001;
strike = 1.60;
start_date = "24-aug-2009";
CURR_FOR = "USD";
IRS_day_to_spot = 2;
smile_interp = "vanna-volga";
PDR = 1;
IRS_DCC = "30/360";
strike_2 = 1.45;
VOL_DCC = "ACT/360";
barrier_end_date = "24-jan-2010";
kappa = 1.5;
premium_curr = "USD";
IRS_EMA = "+1";
dsigma = 0.0001;
CALIBR_BDA = "@";
%exercise_type = "American";
exercise_type = "European";
FC_FOR = {"newyork", "NYMEX"};
```

```
FRA_BDA = "mfbd";
DF_BASE_type = "Mean";
sv_model = "Heston";
lv_model = "CEV";
VOL_BDA = "mfbd";
FC_DOM = {"frankfurt", "TARGET"};
PPO = 2;
V0 = 0.05;
expire_date = "24-may-2010";
FRA_EMA = "+1";
Depo_ref_ind = "EURIBOR";
IRS_cf = 1;
IRS_BDA = "mfbd";
FOR_day_to_spot = 2;
rho = -0.9;
alpha = 5;
day_hat = 5;
omega = 0.3;
ST_BAS_QUO = "true";
strike_3 = 1.5;
theta_init = 0;
S0 = 1.45;
order = 5;
```

przedstawiono poniżej (M_x jest liczbą symulacji, pozostałe wartości są natomiast agregatami obliczonymi na podstawie dziesięciu uruchomień programu z ustalonym parametrem M_x)

M_x	$\overline{V}$	$V_{max} - V_{min}$	$\overline{\Delta}$	$\Delta_{max} - \Delta_{min}$	$\overline{\text{elapsed time}}$
100	0.212855	0.022665	-0.740989	0.120288	3.47
200	0.204844	0.015852	-0.714964	0.098191	3.68
400	0.199377	0.012898	-0.662596	0.052984	4.22
1600	0.195784	0.006268	-0.653915	0.043036	6.58
160000	0.190836	0.000286	-0.648270	0.002055	319.26

Literatura

[1] L. Andersen – *Simple and efficient simulation of the Heston stochastic volatility model*, Journal of Computational Finance, Vol. 11, No 3, 2008, 1-42.

[2] Y. Hiplish – *Fast Monte Carlo Valuation of American Options under Stochastic Volatility and Interest Rates*, August 2011.

[3] J. Hull, A. White – *The Pricing of Options on Assets with Stochastic Volatilities*, The Journal of Finance, Vol. 42, Issue 2, 1987, 281-300.

[4] C. Kahl, P. Jackel – *Fast strong approximation Monte-Carlo schemes for stochastic volatility models*, 2006.

[5] R. Lord, R. Koekkoek, D. van Dijk – *Comparison of biased simulation schemes for stochastic volatility models*, 2008.

[6] E.M. Stein, J.C. Stein – *Stock Price Distributions with Stochastic Volatility: An Analytic Approach*, The review of Financial Studies Vol. 4 No. 4, 1991.

[7] Many authors – *Dokumentacja – Funkcje pomocnicze w pliku calculate_functions*, 2014.

[8] D. Toczydłowska, P. Kosewski, K. Mioduszewska, P. Garbuliński, B. Głowinkowski, A. Palczewski – *Dokumentacja - Wycena opcji amerykańskich na rynku equity metodą Longstaffa-Schwartza*, 2013-2014.

[9] B. Wróblewski, A. Palczewski – *Dokumentacja - Obliczanie cen i parametrów greckich*.

[10] A. Ryterski, M. Sosnowski, B. Milczarek, A. Palczewski – *Dokumentacja - Struktura terminowa stóp procentowych*.

[11] M. Jabłoński, A. Kaszkowiak, K. Aródź, A. Palczewski – *Dokumentacja - Funkcje kalendarzowe*, wersja 2.

[12] M. Broadie, O. Kaya – *Exact simulation of option greeks under stochastic volatility and jump diffusion models*, Proceedings of the 2004 Winter Simulation Conference, 2006

[13] P. Glasserman – *Monte carlo methods in Financial Engineering*, 2003, Stochastic Modeling and Applied Probability, Springer