Estymacja parametrów modelu Hestona - dokumentacja implementacji

Maciej Kołodziejczyk, Michał Kowalski

8 maja 2009

Spis treści

1 Opis problemu

Głownym celem programu jest kalibracja modelu Hestona wyceny opcji ze stochastyczną zmiennością do danych rynkowych, czyli estymacja parametrów występujących w układzie równań różniczkowych opisujących dynamikę cen aktywa bazowego i zmienności:

$$\begin{cases} dS_t = \mu S_t dt + \sqrt{V_t} S_t dW_t^S, \\ dV_t = \gamma(\theta - V_t)dt + \varepsilon\sqrt{V_t} dW_t^V, \end{cases} \tag{1}$$

gdzie W_t^S, W_t^V to standardowe procesy Wienera, startujące z 0, skorelowane z współczynnikiem korelacji ρ. ($dW_t^S dW_t^V = \rho dt$). Dodatkowo, estymowanym parametrem będzie nieobserwowana na rynku początkowa wartość zmienności V_0.

Estymacja polega na minimalizacji funkcji celu - odległości w normie l_1 lub l_2 (do wyboru) cen opcji call (o zadanej cenie wykonania i terminie zapadalności): teoretycznej, wynikającej z modelu Hestona oraz ceny obserwowanej na rynku. Minimalizacja musi uwzględniać określone ograniczenia nierównościowe nałożone na parametry, wynikające z własności i interpretacji finansowej parametrów modelu. Zbiór dopuszczalnych parametrów ma postać:

$$\Theta_{adm} = \left\{ \Theta = (\gamma, \theta, \varepsilon, \rho, \lambda, V_0) : \gamma \geqslant 0, \theta \geqslant 0, \varepsilon \geqslant 0, \rho \in [-1, 1], V_0 \geqslant 0, \frac{2\gamma\theta}{\varepsilon^2} > 1 \right\} \tag{2}$$

Ostatni warunek zapewnia, że, z prawdopodobieństwem 1, V_t przyjmuje wartości dodatnie.

Podstawowym problemem jest zatem minimalizacja funkcji celu zadanej wzorem:

$$\min_{\Theta \in \Theta_{adm}} F(\Theta) = \sum_{i=1}^{n} \left(C_i^{Hes}(\Theta) - C_i^{obs} \right)^2, \tag{3}$$

gdzie n oznacza liczbę obserwacji rynkowej ceny aktywa bazowego i ceny opcji. Oczywiście celem minimalizacji jest znalezienie wektora parametrów Θ_{min}, minimalizującego funkcję celu, natomiast sama jest wartość nie jest szczególnie istotna.

2 Opis algorytmu

2.1 Algorytm minimalizacji funkcji celu

Zadanie minimalizacji (3) jest trudne z wielu powodów. Funkcja celu jest funkcją 5 parametrów o nieznanej z góry regularności. Może ona posiadać wiele minimów lokalnych i dobór parametrów początkowych dla określonego algorytmu minimalizacyjnego może być bardzo trudny. Może się bowiem okazać, że startując z określonego punktu $\Theta_0 \in \Theta_{adm} \subset \mathbb{R}^6$ trafimy w pewne minimum lokalne, podczas gdy minimum globalne w zbiorze dopuszczalnych parametrów może znajdować się zupełnie gdzie indziej.

W związku z tym zdecydowaliśmy się na dwustopniowy algorytm minimalizacyjny. Pierwszym krokiem jest wygenerowanie pięciowymiarowej siatki parametrów, należących do zbioru dopuszczalnych parametrów i wyliczenie w tych punktach wartości funkcji celu. Ze względu, że zbiór parametrów jest pięciowymiarowy[1] i problem ma złozoność obliczeniową $O(N^5)$, sugeruje się wybór do

[1] Zawsze przyjmujemy startową $\lambda = 0$

obliczeń od 5 do 10 wartości każdego z parametrów[2].

Funkcja `grid_search()` pozwala na wygenerowanie N punktów siatki, w których funkcja celu przyjęła najmniejsze wartości. Mając N punktów startowych, uruchamiamy N-krotnie algorytm minimalizujący funkcję celu w otoczeniu punktu startowego metodą największego spadku, czyli poruszamy się w kierunku przeciwnym do gradientu funkcji celu z punktu startowego. Służy do tego funkcja `steepest_descent()`. W każdym kroku metody gradientowej poruszamy się w kierunku przeciwnym do gradientu funkcji celu z ustalonym krokiem[3].

2.2 Obliczanie ceny teoretycznej opcji z modelu Hestona

Istotnym elementem obliczania funkcji celu dla zadanego wektora parametrów jest policzenie ceny teoretycznej opcji call, wynikającej z modelu Hestona. Zaimplementowana funkcja `Heston_call_price()`, wykorzystuje metodę podaną w pracy Hestona [1], s.328-331. Jedyną kwestią wymagającą zastanowienia jest przybliżenie całki danej wzorem (18) w pracy [1] odpowiednią kwadraturą. Kierując się wskazówką autorów tutorialu [2], całkę na przedziale $[0, \infty)$ przybliżamy kwadraturą Gaussa-Legendre'a na przedziale $[0, 100]$, dzieląc ten odcinek na 100 przedziałów, w każdym z przedziałów biorąc wartości w dwóch punktach.

Jednokrotne policzenie funkcji celu wymaga n-krotnego wyliczenia ceny opcji, gdzie n oznacza liczbę okresów obserwacji ceny opcji. Niech $i \in \{1 \dots n\}$ oznacza numer obserwacji. Dla $i = 1$ wartością zmienności wstawianą do funkcji jest parametr V_0. Dla $2 \leqslant i \leqslant n$, wstawiamy wartość oczekiwaną zmienności. Zadana jest ona wzorem:

$$\mathbb{E}V_t = \theta + e^{-\gamma t}(V_0 - \theta). \tag{4}$$

3 Dokumentacja programu

3.1 Opis działania funkcji

Poniżej znajduje się opis funkcji wykorzystywanychj do kalibracji modelu Hestona, zawartych w pliku `kalibracja_Hestona.m`.

```
Heston_call_price(gamma, theta, epsilon, rho, lambda, rd, rf, vt, St,
K, tau)
```

argumenty funkcji:
gamma - parametr gamma modelu Hestona

[2]Wybór 5 wartości dla każdego z parametrów wymaga $5^5 = 3125$ wyliczeń funkcji celu, wybór 10 wartości - $10^5 = 100000$ wyliczeń.

[3]W kodzie skalowaliśmy unormowany gradient przez 10^{-3}. Dobór tego parametru jest arbitralny, testy pokazały że wybór większej wartości powoduje na tyle duże oddalanie się od punktu startowego, że funkcja celu może przyjąć tam wieksze wartości. Z kolei mniejszy mnożnik sprawia, że oddalenie się od punktu startowego jest praktycznie zaniedbywalne i nieistotne dla wartości funkcji celu

theta - parametr theta modelu Hestona
epsilon - parametr epsilon modelu Hestona
rho - parametr rho modelu Hestona (współczynnik korelacji)
lambda - parametr lambda modelu Hestona
rd - krajowa stopa procentowa
rf - zagraniczna stopa procentowa (lub stopa dywidendy)
vt - początkowa zmienność
St - cena aktywa bazowego
K - cena wykonania opcji
tau - czas do wygaśnięcia opcji (w latach)

wynik działania funkcji: Skalar - cena opcji call w modelu Hestona.

opis działania funkcji: Dla zadanych parametrów modelu Hestona oraz charakterystyk opcji, funkcja oblicza cenę waniliowej opcji call w modelu Hestona.

`lik(X, rd, rf, K, T, wektor_czasu, wektor_aktywa, wektor_opcji, norma)`

argumenty funkcji:
X - 6-ciowymiarowy wektor parametrów modelu Hestona $(\gamma, \theta, \varepsilon, \rho, \lambda, v_0)$
rd - wektor krajowych stóp procentowych dla czasów takich jak w wektorze wektor_czasu
rf - wektor zagranicznych stóp procentowych (lub stóp dywidendy) dla czasów takich jak w wektorze wektor_czasu
K - cena wykonania opcji call użytej do estymacji
T - moment wykonania opcji call użytej do estymacji (w latach)
wektor_czasu - wektor czasów obserwacji cen aktywa i opcji
wektor_aktywa - wektor cen aktywa bazowego dla czasów takich jak w wektorze wektor_czasu
wektor_opcji - wektor cen opcji call dla czasów takich jak w wektorze wektor_czasu
norma - norma wektora parametrów dla funkcji celu (1 - norma w l_1, 2 - norma w l_2)

wynik działania funkcji: Skalar - wartość funkcji celu dla zadanych argumentów.
opis działania funkcji: Dla podanego wektora parametrów modelu Hestona oraz charakterystyk opcji call funkcja zwraca wartość funkcji celu, będącej normą różnicy między wektorem zaobserwowanych cen opcji a wektorem cen opcji uzyskanych z modelu Hestona.

`admissible(x)`

argumenty funkcji:

x - 6-ciowymiarowy wektor parametrów modelu Hestona

wynik działania funkcji: Wartość logiczna:
1 - jeżeli wektor x należy do zbioru parametrów dopuszczalnych dla modelu Hestona
0 - jeżeli wektor x nie należy do zbioru parametrów dopuszczalnych dla modelu Hestona

opis działania funkcji: Dla zadanego wektora parametrów funkcja sprawdza czy wektor ten jest wektorem parametrów dopuszczalnych dla modelu Hestona.

`grid_search(rd, rf, K, T, t, wektor_aktywa, wektor_opcji, norma, N, par_min, par_max, M)`

argumenty funkcji:
rd - wektor krajowych stóp procentowych dla czasów takich jak w wektorze t
rf - wektor zagranicznych stóp procentowych (lub stóp dywidendy) dla czasów takich jak w wektorze t
K - cena wykonania opcji call użytej do estymacji
T - moment wykonania opcji call użytej do estymacji (w latach)
t - wektor czasów obserwacji cen aktywa i opcji
wektor_aktywa - wektor cen aktywa bazowego dla czasów takich jak w wektorze t
wektor_opcji - wektor cen opcji call dla czasów takich jak w wektorze t
norma - norma wektora parametrów dla funkcji celu (1 - norma w l_1, 2 - norma w l_2)
N - ilość zapamiętywanych zestawów parametrów, przy których wartość funkcji celu jest najmniejsza na zadanej siatce
par_min - 5-cioelementowy wektor minimalnych wartości siatki dla estymowanych parametrów
par_max - 5-cioelementowy wektor maksymalnych wartości siatki dla estymowanych parametrów
M - ilość równorozłożonych wartości na siatce w każdym wymiarze (M^5 to zatem ilość punktów, w których badana jest wartość funkcji celu)

wynik działania funkcji: Dwuelementowa lista zawierająca kolejno:
1) wektor rozmiaru N wartości funkcji celu odpowiadających zestawom parametrów z macierzy z punktu 2)
2) macierz rozmiaru $N \times 6$ zawierająca N zestawów parametrów z siatki, dla których funkcja celu przyjmuje najmniejsze wartości

opis działania funkcji: Funkcja tworzy 5-ciowymiarową siatkę w przestrzeni parametrów modelu Hestona i wyznacza N zestawów parametrów z siatki, dla których wartość funkcji celu (funkcja `lik`) jest najmniejsza. Funkcja zwraca również wartość funkcji celu w tych N punktach. Nie tworzymy punktów siatki, dla parametru λ, zawsze przyjmujemy $\lambda = 0$.

```
steepest_descent(gamma0, theta0, espilon0, rho0, lambda0, v00, rd,
rf, K, T, wektor_czasu, wektor_aktywa, wektor_opcji, norma, max_iter)
```

argumenty funkcji:
gamma0 - początkowy parametr gamma modelu Hestona
theta0 - początkowy parametr theta modelu Hestona
epsilon0 - początkowy parametr epsilon modelu Hestona
rho0 - początkowy parametr rho modelu Hestona
lambda0 - początkowy parametr lambda modelu Hestona
v00 - początkowy parametr v0 modelu Hestona
rd - wektor krajowych stóp procentowych dla czasów takich jak w wektorze wektor_czasu
rf - wektor zagranicznych stóp procentowych (lub stóp dywidendy) dla czasów takich jak w wektorze wektor_czasu
K - cena wykonania opcji call użytej do estymacji
T - moment wykonania opcji call użytej do estymacji (w latach)
wektor_czasu - wektor czasów obserwacji cen aktywa i opcji
wektor_aktywa - wektor cen aktywa bazowego dla czasów takich jak w wektorze wektor_czasu
wektor_opcji - wektor cen opcji call dla czasów takich jak w wektorze wektor_czasu
norma - norma wektora parametrów dla funkcji celu (1 - norma w l_1, 2 - norma w l_2)
max_iter - maksymalna ilość iteracji algorytmu

wynik działania funkcji: Dwuelementowa lista zawierająca kolejno:
1) skalar będący minimalną wartością funkcji celu uzyskaną w trakcie działania algorytmu gradientowego
2) wektor 6-cioelementowy parametrów modelu Hestona odpowiadający wartości funkcji celu z punktu 1)

opis działania funkcji: Funkcja szuka minimum funkcji celu (funkcja `lik`) na zbiorze parametrów dopuszczalnych dla modelu Hestona stosując algorytm gradientowy i zaczynając z zestawu podanych parametrów początkowych. Funkcja zwraca minimalną wartość funkcji celu napotkaną podczas wykonywania iteracji algorytmu gradientowego oraz zestaw parametrów odpowiadający tej wartości.

```
best_parameters(B, best_lik, rd, rf, K, T, t, St, wektor_opcji, norma,
max_iter)
```

argumenty funkcji:
B - macierz zestawów parametrów $\gamma, \theta, \varepsilon, \rho, \lambda, v_0$) rozmiaru $N \times 6$, gdzie $N \in \mathbb{N}$
best_lik - wektor wartości funkcji celu odpowiadający zestawom parametrów z

macierzy B
rd - wektor krajowych stóp procentowych dla czasów takich jak w wektorze t
rf - wektor zagranicznych stóp procentowych (lub stóp dywidendy) dla czasów takich jak w wektorze t
K - cena wykonania opcji call użytej do estymacji
T - moment wykonania opcji call użytej do estymacji (w latach)
t - wektor czasów obserwacji cen aktywa i opcji
St - wektor cen aktywa bazowego dla czasów takich jak w wektorze t
wektor_opcji - wektor cen opcji call dla czasów takich jak w wektorze t
norma - norma wektora parametrów dla funkcji celu (1 - norma w l_1, 2 - norma w l_2)
max_iter - maksymalna ilość iteracji algorytmu

wynik działania funkcji: Dwuelementowa lista zawierająca kolejno:
1) skalar będący minimalną wartością funkcji celu uzyskaną w trakcie działania algorytmów gradientowych
2) wektor 6-cioelementowy parametrów modelu Hestona odpowiadający wartości funkcji celu z punktu 1)

opis działania funkcji: Funkcja szuka minimum funkcji celu (funkcja lik) na zbiorze parametrów dopuszczalnych dla modelu Hestona stosując algorytm gradientowy. Algorytm gradientowy wykonywany jest N-krotnie, dla punktów startowych będących kolejnymi wierszami macierzy B. Funkcja zwraca minimalną wartość funkcji celu napotkaną podczas wykonywania iteracji wszystkich N algorytmów gradientowych oraz zestaw parametrów odpowiadający tej wartości.

3.2 Wprowadzanie danych

Wprowadzanie danych wejściowych potrzebnych do inicjalizacji programu następuje z dwóch plików: `zad10_market_data.txt` oraz `zad10_control_data.txt`. Pliki te powinny, w chwili uruchomienia programu, znajdować się w tym samym folderze co plik z kodem. Poza tym, w pierwszej fazie działania programu użytkownik proszony jest o podanie normy (l_1 lub l_2), w której minimalizowana będzie odległość między rynkowymi cenami opcji a cenami implikowanymi przez model Hestona o poszukiwanych parametrach.

Pliki `zad10_market_data.txt` oraz `zad10_control_data.txt` powinny zawierać następujące dane, które są niezbędne do prawidłowego funkcjonowania programu (przy każdej zmiennej podany został jej typ oraz krótki opis).

`zad10_market_data.txt`:
K (scalar) - cena wykonania opcji call użytej do estymacji
wektor_aktywa (matrix, rows:1) - wektor cen aktywa bazowego dla czasów takich jak w wektorze wektor_czasu
wektor_opcji (matrix, rows:1) - wektor cen opcji call dla czasów takich jak w wektorze wektor_czasu

wektor_czasu (matrix, rows:1) - wektor czasów obserwacji cen aktywa i opcji
rf (matrix, rows:1) - wektor zagranicznych stóp procentowych (lub stóp dywidendy) dla czasów takich jak w wektorze wektor_czasu
rd (matrix, rows:1) - wektor krajowych stóp procentowych dla czasów takich jak w wektorze wektor_czasu
T (scalar) - moment wykonania opcji call użytej do estymacji (w latach)

`zad10_control_data.txt`:
N (scalar) - ilość zapamiętywanych zestawów parametrów z siatki, dla których funkcja celu jest najmniejsza (do zastosowania metody gradientowej)
max_iter (scalar) - maksymalna ilość iteracji w algorytmie gradientowym
par_min (matrix, rows: 1, columns: 5) - wektor minimalnych wartości siatki dla estymowanych parametrów
par_max (matrix, rows: 1, columns: 5) - wektor maksymalnych wartości siatki dla estymowanych parametrów
M (matrix, rows: 1, columns: 5) - ilość równorozłożonych wartości na siatce w każdym wymiarze

Do poprawnego działania programu niezbędne są dodatkowo następujące własności wprowadzanych z plików danych :
1) dane wektor_aktywa, wektor_opcji, wektor_czasu, rf, rd muszą być macierzami o takich samych wymiarach
2) dane N oraz max_iter muszą być liczbami naturalnymi

Przykładowe dane wejściowe do programu znajdują się w plikach `zad10_market_data.txt` oraz `zad10_control_data.txt` dołączonych do kodu programu.

3.3 Zwracanie wyników

Finalnym rezultatem działania programu jest otrzymanie wektora optymalnie estymującego parametry modelu Hestona w sensie podanym w opisie problemu i opisie działania algorytmu. Współrzędne tego wektora parametrów wyświetlane są na ekranie po zakończeniu działania programu w sposób umożliwiający ich natychmiastową identyfikację.

4 Wyniki testów programu

4.1 Testy funkcji `Heston_call_price()`

Choć funkcja ta jest obliczana na podstawie wzorów z pracy Hestona [1], to możemy porównać otrzymywane wyniki z wynikami symulacji Monte Carlo, przeprowadzanymi za pomocą programu napisanego w ramach Zadania 6. Można zobaczyć, czy cena opcji dla zadanych parametrów znajduje się w 95%-przedziale ufności wyznaczonym w Zadaniu 6[4]. Poniżej prezentujemy przykładowe rezultaty testów, do których używaliśmy kodu z Zadania 6[5].

Poniższa tabela prezentuje wyniki dla danych startowych: $S_0 = 4$, $K = 3.8$, $r = 0.04$ i $T = 1$ oraz wektorów parametów danych w kolumnie *Parametry* w kolejności: $(\gamma, \theta, \varepsilon, \rho, \lambda, V_0)$

Parametry	*Cena Hestona*	*95% P. U.*	*Czy należy do P.U.*
(8,0.05,0.2,-0.4,0,0.15)	0.58110	[0.57366, 0.58318]	TAK
(2,0.1,0.4,0.7,0,0.3)	0.83437	[0.81720, 0.83810]	TAK
(3,0.04,0.15,-0.6,0,0.2)	0.65447	[0.64887, 0.66009]	TAK
(5.5,0.1,0.5,-0.9,0,0.5)	0.82000	[0.80530, 0.81984]	NIE
(5.5,0.1,0.5,0,0,0.5)	0.81571	[0.80281, 0.81969]	TAK

4.2 Testy algorytmu minimalizacyjnego

Z uwagi na konstrukcję metody minimalizacji (najpierw wyliczamy wartości na pewnej siatce parametrów, następnie dla N najmniejszych wartości funkcji celu, poszukujemy minimum w otoczeniu danych punktów siatki metodą gradientową), najważniejsze było upewnienie się, że algorytm nie będzie w trakcie swojego działania przechodził do wektorów parametrów zwiększających w danym kroku funkcję celu.

Warto zauważyć, że nie zawsze zwiększenie liczby punktów siatki w których wyliczamy wartość funkcji celu (za pomocą funkcji `grid_search()`), prowadzi do znalezienia zestawu parametrów gwarantujących mniejszą wartość funkcji celu. Wielkość siatki podyktowana powinna być preferencjami użytkownika, który musi pamiętać o kosztach czasowych jej rozszerzania.

Przykładowe testy wykonaliśmy na danych rynkowych wygenerowanych następująco: Za pomocą programu z Zadania 6. wygenerowaliśmy trajektorię cen aktywa bazowego dla następujących danych: $S_0 = 4$, $K = 3.9$, $r_d = 0.05$, $r_f = 0$, $T = 1$ i parametru, którego będziemy szukać za pomocą naszego algorytmu:

$$\Theta^* = (\gamma, \theta, \varepsilon, \rho, \lambda, V_0)^* = (5.94, 0.045, 0.31, -0.576, 0, 0.15)$$

Wartości danych rynkowych znajdują się w dołączonym pliku z danymi `zad10_market_data.txt`. Wektor cen opcji wygenerowano używając funkcji `Heston_call_price()`. Wybraliśmy normę l_2.

Wyniki testów podsumowuje tabela, w kolumnach kolejno: numer testu, siatka, wektor parametrów minimalizujący funkcję celu, minimalna wartość funkcji celu, czas działania algorytmu.

[4] Ponieważ w Zadaniu 6 nie występuje parametr λ, oznaczający "cenę ryzyka zmienności", jak określa to Heston, testowaliśmy przyjmując $\lambda = 0$.

[5] W większosci przypadków cena z pracy Hestona mieści się w 95% przedziale ufności, nie stwierdziliśmy istotnych odchyleń. Przypadek pokazany w tabeli znajduje się bardzo blisko przedziału, jego wygenerowanie wymagało od nas bardzo dużo cierpliwości.

Lp.	*Siatka*	*Est. wektor parametrów*	*F. celu l_2*	*Sek.*
	norma l_2			
1	(5,5,5,5,5)	(5.48,0.049,0.7278,-0.034,0.009,0.1404)	3.83×10^{-5}	205
2	(6,6,6,6,6)	(9.99,0.049,0.6401,-0.208,0.000,0.1984)	1.10×10^{-4}	342
3	(10,5,5,10,5)	(5.98,0.048,0.6968,-0.119,0.009,0.1486)	1.54×10^{-5}	484
4	(5,5,5,20,5)	(5.48,0.048,0.7169,-0.052,0.010,0.1416)	3.01×10^{-5}	478
5	(15,5,5,5,5)	(5.48,0.049,0.7247,-0.034,0.010,0.1403)	3.83×10^{-5}	388
6	(5,15,5,5,5)	(5.49,0.049,0.6003,-0.001,0.000,0.1385)	7.70×10^{-5}	386
7	(5,5,15,5,5)	(5.48,0.049,0.7247,-0.034,0.010,0.1404)	3.83×10^{-5}	391
8	(7,7,7,7,7)	(8.50,0.049,0.6000,-0.301,0.000,0.1704)	6.94×10^{-5}	601
	norma l_1			
9	(5,5,5,5,5)	(5.50,0.049,0.6000,0.000,0.000,0.1374)	2.28×10^{-2}	206
10	(6,6,6,6,6)	(6.39,0.048,0.4357,-0.206,0.014,0.1455)	1.82×10^{-2}	338

4.3 Wnioski i obserwacje z testów

Testy numeryczne dostarczyły nam następujących wniosków dotyczących zachowania algorytmu:

1. Wyniki estymacji nie są satysfakcjonujące zwłaszcza dla parametrów ε (zmienność zmienności) oraz ρ (korelacja).

2. Nie zawsze ilość punktów siatki przekłada się na wartość funkcji celu.

3. Decydujący wpływ na wyestymowaną wartość parametrów ma dobór siatki, algorytm gradientowy w sposób śladowy wpływa na ostateczną wartość parametrów.

4. Na pewno duże znaczenie ma niewielka długość wektorów danych rynkowych, którą testowaliśmy w przykładzie ($n = 12$), lecz jej zwiększenie znacznie zwiększa czas obliczeń.

5. Wartości funkcji celu są większe w normie l_1 niż w l_2 - jest to efekt oczekiwany ze względu na to że minimalizujemy sumę różnic cen opcji.

Tabela prezentuje tylko przykład wyników testów, jakie przeprowadziliśmy. W trakcie pisania programu, znając a priori parametry, zawężaliśmy siatkę do przedziałów wokół "rzeczywistych punktów", uzyskując w ten sposób, dla normy l_2 wartości funkcji celu rzędu 10^{-6}, a w najlepszych przypadkach nawet 10^{-7}. Niemniej w ogólnym przypadku, musimy dopuszczać jak największy zakres parametrów. Niestety, pięciowymiarowość siatki sprawia, że koszt obliczeniowy rozdrabniania podziału jest ogromny: $O(N^5)$.

4.4 Czas działania programu

Czas działania programu zależy głównie od:

1. długości wektorów rynkowych cen aktywa bazowego/opcji call

2. wyboru liczby punktów siatki w funkcji `grid_search()`

3. wyboru parametru N - liczby najlepszych punktów z siatki, dla których zainicjowana zostanie funkcja `steepest_descent()`

Literatura

[1] Steven L. Heston, *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, The Review of Financial Studies, vol. 6, n.2, 1993, ss. 327-343.

[2] Rafał Weron, Uwe Wystup, *http://mars.wiwi.hu-berlin.de/tutorials/stfhtmlframe98.html.*