

Dokumentacja

Wycena opcji europejskich

w modelu Hestona

Konrad Stawski

Spis treści

1	Opis problemu	2
2	Opis dyskretyzacji problemu	3
3	Zmienne wykorzystywane w programie	6
4	Spis wykorzystywanych funkcji wraz z opisem działania	7
5	Analiza poprawności otrzymywanych wyników	9

1 Opis problemu

Głównym zadaniem, które ma realizować program, będzie wycena opcji europejskich w modelu Hestona ze stochastyczną zmiennością. W modelu tym dynamika cen aktywa oraz jego zmienności są opisane za pomocą następujących równań różniczkowych:

$$dS(t) = \mu S dt + \sqrt{v(t)} S dz_1(t) \quad (1)$$

$$d\sqrt{v(t)} = -\beta\sqrt{v(t)}dt + \delta dz_2(t) \quad (2)$$

Gdzie $dz_1(t)$, $dz_2(t)$ to dwa standardowe procesy Wienera, między którymi istnieje korelacja ρ , czyli $dz_1(t)dz_2(t) = \rho dt$. Wzór 2 można następnie przekształcić do następującej postaci:

$$dv(t) = (\delta^2 - 2\beta v(t))dt + 2\delta\sqrt{v(t)}dz_1(t) \quad (3)$$

$$dv(t) = \kappa(\theta - v(t))dt + \sigma\sqrt{v(t)}dz_1(t) \quad (4)$$

Przy tych założeniach możliwe jest wyprowadzenie cząstkowego równania różniczkowego dla modelu Hestona. Przyjmuje ono następującą postać:

$$\frac{\partial u}{\partial t} = \frac{1}{2}s^2v\frac{\partial^2 u}{\partial s^2} + \rho\sigma sv\frac{\partial^2 u}{\partial s\partial v} + \frac{1}{2}\sigma^2v\frac{\partial^2 u}{\partial v^2} + (r_d - r_f)s\frac{\partial u}{\partial s} + \kappa(\eta - v)\frac{\partial u}{\partial v} - r_d u \quad (5)$$

Gdzie: u - cena opcji, $0 \leq t \leq T$ - czas, $s > 0$ - cena aktywa, $v > 0$ - wariancja ceny aktywa, $\kappa > 0$ - współczynnik szybkości powrotu do średniej wariancji cen aktywa, $\eta > 0$ - długoterminowa średnia wariancja cen aktywa, $\sigma > 0$ - volatylity wariancji cen aktywa, $\rho \in [-1, 1]$ - korelacja między dwoma procesami Wienera dotyczącymi ceny aktywa i wariancji, r_d i r_f to odpowiednio stopy oprocentowania na rynku krajowym i zagranicznym.

Wycena opcji polega na rozwiązaniu tego równania cząstkowego przy zadanych warunkach brzegowych. Wypiszę teraz warunki brzegowe dla wszystkich rodzajów opcji, które będziemy wyceniać za pomocą tego modelu. Na początku założymy, że rozwiązania będziemy szukać na pewnym ograniczonym obszarze, takim że $s \in [0, S]$ oraz $v \in [0, V]$. K - cena wykonania opcji, B - wartość bariery dla opcji barierowych. Dla standardowej opcji call:

$$u(s, v, 0) = \max(0, s - K) \quad (6)$$

$$u(0, v, t) = 0 \quad (7)$$

$$u(s, V, t) = se^{-r_f t} \quad (8)$$

$$\frac{\partial u}{\partial s}(S, v, t) = e^{-r_f t} \quad (9)$$

Dla standardowej opcji put:

$$u(s, v, 0) = \max(0, K - s) \quad (10)$$

$$u(0, v, t) = Ke^{-r_d t} \quad (11)$$

$$u(s, V, t) = Ke^{-r_d t} \quad (12)$$

$$\frac{\partial u}{\partial s}(S, v, t) = 0 \quad (13)$$

Dla barierowej opcji call down-and-out:

$$u(s, v, 0) = \max(0, s - K) \quad (14)$$

$$u(B, v, t) = 0 \quad (15)$$

$$u(s, V, t) = \left(1 - \frac{B}{S}\right) s e^{-rft} \quad (16)$$

$$\frac{\partial u}{\partial s}(S, v, t) = e^{-rft} \quad (17)$$

Dla barierowej opcji put up-and-out:

$$u(s, v, 0) = \max(0, K - s) \quad (18)$$

$$u(B, v, t) = 0 \quad (19)$$

$$u(s, V, t) = \left(1 - \frac{S}{B}\right) K e^{-rat} \quad (20)$$

$$u(0, v, t) = K e^{-rat} \quad (21)$$

$$\frac{\partial u}{\partial s}(S, v, t) = e^{-rat} \quad (22)$$

2 Opis dyskretyzacji problemu

W naszym programie będziemy przyjmować, że $S = 8K$ i $V = 5$. Takie wartości umożliwiają uzyskanie bardzo małego błędu dla praktycznie każdego zestawu danych.

Najpierw musimy zdyskretyzować naszą przestrzeń $[0, S] \times [0, V]$. Będziemy stosować dyskretyzację nierównomierną, tak aby uzyskać większe zagęszczenie punktów dla s bliskich K , oraz dla v bliskich 0 . Najpierw wyznaczymy siatkę punktów na osi s . Niech m_1 oznacza liczbę punktów na osi s , a $c > 0$ będzie dowolną stałą. Wtedy dla $0 \leq i \leq m_1$

$$\xi_i = \sinh^{-1}(-K/c) + i\Delta\xi \quad (23)$$

$$\Delta\xi = \frac{1}{m_1} \left[\sinh^{-1}\left(\frac{S-K}{c}\right) - \sinh^{-1}\left(-\frac{K}{c}\right) \right] \quad (24)$$

Następnie nierównomierna siatka na dla s jest definiowana za pomocą przekształcenia

$$s_i = K + c \sin(\xi_i) \quad (25)$$

Stała c decyduje o zagęszczeniu punktów w pobliżu ceny wykonania. W naszym programie będziemy przyjmować, że $c = \frac{K}{5}$.

Analogicznie definiujemy siatkę dla zmiennej v . Niech m_2 oznacza liczbę punktów na osi v , $0 \leq j \leq m_2$, $d > 0$ będzie dowolną stałą.

$$\eta_j = j\Delta\eta \quad (26)$$

$$\Delta\eta = \frac{1}{m_2} \sinh^{-1}\left(\frac{V}{d}\right) \quad (27)$$

Parametr d decyduje tym razem o zagęszczeniu punktów w pobliżu $v=0$. W naszym programie będziemy przyjmować, że $d = \frac{V}{500}$. Ostatecznie punkty siatki dla zmiennej v definiowane są przez:

$$v_j = d \sinh(\eta_j) \quad (28)$$

Teraz musimy zdefiniować w jaki sposób będą aproksymowane pochodne cząstkowe dla poszczególnych zmiennych. W tym celu będziemy używali schematu różnic skończonych. Dla dowolnej funkcji f niech $x_0 < x_1 < \dots < x_m$ będą punktami siatki zmiennej x oraz $\Delta x_i = x_i - x_{i-1}$. Wtedy do aproksymacji pierwszej pochodnej funkcji f będziemy wykorzystywali trzy schematy:

$$f'(x_i) = \alpha_{i,-2}f(x_{i-2}) + \alpha_{i,-1}f(x_{i-1}) + \alpha_{i,0}f(x_i) \quad (29)$$

$$f'(x_i) = \beta_{i,-1}f(x_{i-1}) + \beta_{i,0}f(x_i) + \beta_{i,1}f(x_{i+1}) \quad (30)$$

$$f'(x_i) = \gamma_{i,0}f(x_i) + \gamma_{i,1}f(x_{i+1}) + \gamma_{i,2}f(x_{i+2}) \quad (31)$$

Gdzie współczynniki dane są przez:

$$\alpha_{i,-2} = \frac{\Delta x_i}{\Delta x_{i-1}(\Delta x_{i-1} + \Delta x_i)}, \alpha_{i,-1} = \frac{-\Delta x_{i-1} - \Delta x_i}{\Delta x_{i-1}\Delta x_{i-1}}, \alpha_{i,0} = \frac{\Delta x_{i-1} + 2\Delta x_i}{\Delta x_i(\Delta x_{i-1} + \Delta x_i)} \quad (32)$$

$$\beta_{i,-1} = \frac{-\Delta x_{i+1}}{\Delta x_i(\Delta x_i + \Delta x_{i+1})}, \beta_{i,0} = \frac{\Delta x_{i+1} - \Delta x_i}{\Delta x_{i-1}\Delta x_{i-1}}, \beta_{i,1} = \frac{\Delta x_i}{\Delta x_{i+1}(\Delta x_i + \Delta x_{i+1})} \quad (33)$$

$$\gamma_{i,0} = \frac{(-2\Delta x_{i+1} - \Delta x_{i+1})}{\Delta x_{i+1}(\Delta x_{i+1} + \Delta x_{i+2})}, \gamma_{i,1} = \frac{\Delta x_{i+1} + \Delta x_{i+2}}{\Delta x_{i+1}\Delta x_{i+2}}, \gamma_{i,2} = \frac{-\Delta x_{i+1}}{\Delta x_{i+2}(\Delta x_{i+1} + \Delta x_{i+2})} \quad (34)$$

Aby aproksymować drugą pochodną będziemy używać tylko jednego schematu:

$$f''(x_i) = \delta_{i,-1}f(x_{i-1}) + \delta_{i,0}f(x_i) + \delta_{i,1}f(x_{i+1}) \quad (35)$$

gdzie

$$\delta_{i,-1} = \frac{2}{\Delta x_i(\Delta x_i + \Delta x_{i+1})}, \delta_{i,0} = \frac{-2}{\Delta x_{i-1}\Delta x_{i-1}}, \delta_{i,1} = \frac{2}{\Delta x_{i+1}(\Delta x_i + \Delta x_{i+1})} \quad (36)$$

Do aproksymowania pochodnej cząstkowej mieszanej dla funkcji f dwóch zmiennych będziemy używali 33 do wyznaczenia współczynników $\beta_{i,k}$ i $\hat{\beta}_{i,k}$ odpowiednio po zmiennej x i zmiennej y . A następnie do aproksymacji mieszanej pochodnej używamy wzoru:

$$\frac{\partial f}{\partial x \partial y} = \sum_{k,l=-1}^1 \beta_{i,k} \hat{\beta}_{j,l} f(x_{i+k}, y_{j+l}) \quad (37)$$

Teraz przechodzimy do konkretnej dyskretyzacji cząstkowego równania różniczkowego dla modelu Hestona. Najpierw określamy siatkę w $[0, S] \times [0, V]$ w następujący sposób:

$$G = \{(s_i, v_j) : 1 \leq i \leq m_1, 0 \leq j \leq m_2 - 1\} \quad (38)$$

Następnie dla tak zadanej siatki każda pochodna występująca w 5 jest przybliżana za pomocą odpowiadającej jej centralnej różnicy skończonej, czyli 33, 35 lub 37, poza miejscami gdzie $v > 1$ i na barierach $v = 0$ i $s = S$.

Dla $v > 1$ do aproksymowania pochodnej $\frac{\partial u}{\partial v}$ wykorzystywany jest wzór 32.

Dla $v = 0$, pochodna $\frac{\partial u}{\partial v}$ jest aproksymowana za pomocą wzoru 34.

Dla $s = S$ aproksymujemy pochodną $\frac{\partial^2 u}{\partial s^2}$ używając wzoru 35 z wirtualnym punktem $S + \Delta s_{m_1}$, gdzie wartość w tym punkcie jest wyznaczana poprzez ekstrapolację przy użyciu warunku brzegowego.

Dyskretyzacja za pomocą różnic skończonych przedstawiona powyżej prowadzi do otrzymania układu równań różniczkowych, który można zapisać jako:

$$U'(t) = AU(t) + b(t) \quad (0 \leq t \leq T), \quad U(0) = U_0 \quad (39)$$

gdzie A to macierz $m \times m$, a $b(t)$ i U_0 to dane wektory m -elementowe dla $m = m_1 m_2$. Wektor U_0 jest otrzymywany bezpośrednio z warunku początkowego dla $t = 0$, natomiast wektor b otrzymywany jest zawsze z odpowiednich warunków brzegowych. Dla każdego $t > 0$ wektor rozwiązania $U(t)$ zawiera aproksymację rozwiązań $u(s, v, t)$ w punkcie siatki $(s, v) \in G$ posegregowane w ten sposób, że na pierwszych m_1 miejscach stoją wartości dotyczące $u(s, v_0, t)$, na kolejnych m_1 miejscach wartości dotyczące $u(s, v_1, t)$ itd...

Następnym krokiem jest dyskretyzacja czasu. Niech $\Delta t > 0$ będzie danym krokiem czasowym, wtedy kolejne punkty siatki po czasie będą dane przez $t_n = n\Delta t$ dla $n = 0, 1, 2, \dots$. Aby znacznie zmniejszyć czas działania programu będziemy wykorzystywać zmodyfikowanego schematu Craiga-Sneyda (MCS). Aby go zastosować najpierw dokonujemy rozkładu macierzy A na sumę macierzy:

$$A = A_0 + A_1 + A_2$$

gdzie A_0 zawiera wszystkie wyrazy dotyczące pochodnej mieszanej, A_1 wyrazy dotyczące pochodnych po s , natomiast A_2 wszystkie wyrazy dotyczące pochodnych po v . Wyraz r_{dv} występujący w 5 jest rozkładany po równo między A_1 oraz A_2 . Następnie rozkładamy $b(t)$ analogicznie na $b(t) = b_0(t) + b_1(t) + b_2(t)$. Następnie definiujemy F_j dla $j = 0, 1, 2$ przez:

$$F_j(t, U(t)) = A_j U(t) + b_j(t) \quad (40)$$

Oraz $F = F_0 + F_1 + F_2$. Następnie zastosowanie schematu MCS dla zadanego θ będzie w każdym kroku dawało aproksymacje U_n dla dokładnych wartości $U(t_n)$ po kolei dla każdego $n = 1, 2, 3, \dots$

Schemat MCS:

$$\begin{cases} Y_0 = U_{n-1} + \Delta t F(t_{n-1}, U_{n-1}) \\ Y_j = Y_{j-1} + \theta \Delta t (F_j(t_n, Y_j) - F_j(t_{n-1}, U_{n-1})) \quad (j = 1, 2) \\ \hat{Y}_0 = Y_0 + \theta \Delta t (F_0(t_n, Y_2) - F_0(t_{n-1}, U_{n-1})) \\ \check{Y}_0 = \hat{Y}_0 + (\frac{1}{2} - \theta) \Delta t (F(t_n, Y_2) - F(t_{n-1}, U_{n-1})) \\ \check{Y}_j = \check{Y}_{j-1} + \theta \Delta t (F_j(t_n, \check{Y}_j) - F_j(t_{n-1}, U_{n-1})) \quad (j = 1, 2) \\ U_n = \check{Y}_2 \end{cases} \quad (41)$$

W tak zdefiniowanym schemacie F_1 oraz F_2 są wyznaczone sposobem niejawnym (implicit), natomiast F_0 sposobem jawnym (explicit). W każdym kroku działania algorytmu program musi rozwiązać układ równań liniowych zawierający dwie macierze

$(I - \theta \Delta t A_j)$ dla $j = 1, 2$. Dzięki zastosowaniu schematu MCS są one niezależne od indeksu n , a zatem ich rozkład LU może zostać dokonany raz na początku, a następnie zastosowany do obliczania każdego U_n ($n \geq 1$). Dodatkowo macierze te mają stałą szerokość pasma, niezależną od m_1 oraz m_2 , a zatem ilość operacji w każdym kroku jest proporcjonalna do $m = m_1 m_2$ i umożliwia to bardzo dużą oszczędność czasu w porównaniu do innych schematów takich jak Cranka-Nicolson.

3 Zmienne wykorzystywane w programie

Zmienne pobierane z pliku `heston_data.txt`:

otype – zmienna dotycząca typu opcji. Może mieć następujące wartości:

- 0** – call
- 1** – put
- 80** – call down-and-out
- 81** – put up-and-out
- 801** – call down-and-in
- 811** – put up-and-in

spot – aktualna cena aktywa (waluty zagranicznej)

T – czas trwania opcji w latach

vol – volatylity cen aktywa (pierwiastek z wariancji)

rd – stopa oprocentowania na rynku krajowym

rf – stopa oprocentowania na rynku zagranicznym

K – cena wykonania opcji (strike)

B – ewentualna bariera. Dla opcji niebarierowych wartość tego parametru jest pomijana

Zmienne służące jedynie do kalibracji modelu:

sigma – volatylity wariancji cen aktywa (> 0) (domyślnie 0.04)

k – współczynnik szybkości powrotu do średniej wariancji cen aktywa (κ) (> 0) (domyślnie 3)

ro – korelacja między dwoma procesami Wienera dotyczącymi ceny aktywa i wariancji ($-1 \leq ro \leq 1$) (domyślnie 0.6)

eta – długoterminowa średnia wariancja cen aktywa (domyślnie 0.12)

m1 – liczba węzłów ze względu na zmienną s (ceny aktywa) (domyślnie 200)

m2 – liczba węzłów ze względu na zmienną v (wariancja cen aktywa) (domyślnie 100)

omesh – zmienna zero-jedynkowa. Dla wartości 1 generuje wykres Cena od s i v .

4 Spis wykorzystywanych funkcji wraz z opisem działania

Funkcje zawarte w programie w porządku występowania:

- MeshGen** – funkcja tworząca siatkę. Tworzy dwa wektory s i v . Wektor s zawiera kolejne punkty siatki ze względu na zmienną s , natomiast v zawiera punkty siatki ze względu na zmienną v . S i V to wartości największe, ustawione zgodnie z opisem na $S=8K$ oraz $V=5$.
- MeshGenDaO i MeshGenUaO** – funkcje tworzące siatkę do opcji typu down-and-out i up-and-out. Analogicznie do MeshGen, tylko uwzględniają bariery, zatem odpowiednio wartość minimalna lub maksymalna dla zmiennej s wynosi B .
- sB** – generuje 3 wektory współczynników estymacji pierwszych pochodnych po zmiennej s . Obliczane zgodnie ze wzorem 33
- sD** - generuje 3 wektory współczynników estymacji drugich pochodnych po zmiennej s . Obliczane zgodnie ze wzorem 35
- sD_Scall** – dla opcji call generuje 2 współczynniki estymacji drugich pochodnych po zmiennej s , które stosowane są dla największego s ($s=S$ lub $s=B$).
- sD_Sput** - dla opcji put. Generuje 2 współczynniki estymacji drugich pochodnych po zmiennej s , które stosowane są dla największego s ($s=S$ lub $s=B$). Dla tej bariery opcja ma wartość 0, stąd zmiany we współczynnikach w porównaniu do **sD_Scall**.
- vB** - generuje 3 wektory współczynników estymacji pierwszych pochodnych po zmiennej v . Liczone zgodnie ze wzorem 33
- vA** - generuje 3 wektory współczynników estymacji pierwszych pochodnych po zmiennej v . Liczone zgodnie ze wzorem 32. Stosowane zgodnie z opisem, gdy $v>1$.
- vG** - generuje 3 wektory współczynników estymacji pierwszych pochodnych po zmiennej v . Liczone zgodnie ze wzorem 34. Stosowane tylko dla $v=0$, więc wykorzystywane są tylko pierwsze komórki wygenerowanych wektorów.
- vD** - generuje 3 wektory współczynników estymacji drugich pochodnych po zmiennej v . Liczone zgodnie ze wzorem 35
- Long** – funkcja dla zadanych wektorów s, v tworzy dwie macierze diagonalne zawierające na diagonalach powielone odpowiednio wektory s i v . Tworzy je tak, aby Longs mnożone przez macierz A , mnożyło każdy wyraz macierzy A przez odpowiadający mu koordynat s . Analogicznie Longv. Wektory s_{small} i v_{small} powstają z wektorów s i v poprzez „odcięcie” $s=0$ i $v=V$.
- A1** – funkcja generująca pełną macierz współczynników estymacji pochodnych zmiennej s

- A2** – funkcja generująca pełną macierz współczynników estymacji pochodnych zmiennej v
- A0** – funkcja generująca pełną macierz współczynników estymacji pochodnych mieszanych.
- B0call, B0put, B0DaO, B0UaO** – dla funkcji odpowiednio call, put, call down-and-out, put down-and-out, zwraca wektor b_0 , czyli warunek brzegowy ze względu na współczynniki estymacji pochodnych mieszanych.
- B1call** – dla funkcji call lub call down-and-out zwraca wektor b_1 , czyli warunek brzegowy ze względu na współczynniki estymacji pochodnych po s .
- B1put** – dla funkcji put lub put up-and-out zwraca wektor b_1 , czyli warunek brzegowy ze względu na współczynniki estymacji pochodnych po v .
- B2call, B2put, B2DaO, B2UaO** – dla funkcji odpowiednio call, put, call down-and-out, put down-and-out, zwraca wektor b_2 , czyli warunek brzegowy ze względu na współczynniki estymacji pochodnych po zmiennej v .
- UstartCall, UstartPut** – dla opcji odpowiednio call lub call down-and-out i put lub put up-and-out funkcja zwraca wektor zawierający wypłaty z opcji w chwili końcowej. Wektor ten stosowany jest jako warunek początkowy dla funkcji Heston.
- Umatrix** – funkcja przekształca wektor U , na macierz. W wierszu i stoją dane dotyczące $v(i)$ w kolumnie j stoją dane dotyczące $s(j+1)$.
- FindPrice** – funkcja oblicza cenę dla zadanej macierzy U oraz konkretnych $sstart$ i $vstart$. W tym celu stosowana jest interpolacja między węzłami.
- FindGreeks** – funkcja oblicza greckie parametry (delta, gamma, vega) dla zadanej macierzy U oraz konkretnych $sstart$ i $vstart$. Stosowana jest w tym celu metoda five-point-stencil, służąca do estymacji pochodnych na podstawie pięciu wartości.
- FindTheta** – funkcja oblicza grecki parametr Theta dla zadanych macierzy $Ufull$, $UfullTF$ i $UfullTB$. $UfullTB$ powinno zawierać ceny dla czasu o jednostkę wstecz, $Ufull$ aktualne ceny, a $UfullTF$ ceny dla czasu o jednostkę do przodu. Estymacja za pomocą metody three-point-stencil, służącej do estymacji pochodnych za pomocą 3 wartości.
- AddBoundries** – funkcja dołącza do zadanej macierzy U warunki brzegowe wynikające z podanego typu opcji. Otrzymana macierz ma rozmiary $(m2+1)$ na $(m1+1)$.
- MCS** – funkcja realizująca schemat zmodyfikowanego Craig-Sneyda zgodnie ze wzorami 41. Wymaga wcześniejszego wyliczenia wszystkich potrzebnych elementów, a zatem początkowej macierzy U , macierzy A_0, A_1, A_2 zawierających współczynniki estymacji pochodnych (oraz rozkładów LU macierzy $(I - \theta A_i)$) oraz dostępu do funkcji generujących wektory warunków brzegowych b_0, b_1, b_2 .

Heston – główna funkcja. Dla podanego typu opcji (call, put, call down-and-out lub put up-and-out) i wszelkich jej parametrów wywołuje po kolei funkcje generujące współczynniki estymacji pochodnych, następnie wywołuje funkcje tworzące macierze A_0, A_1, A_2 i dokonuje rozkładu LU macierzy $(I - \theta A_i)$. Następnie używa funkcji MCS w celu obliczenia cen opcji dla wszystkich s i v , wynik znajduje się w wektorze U , który następnie jest przekształcany na macierz U_{full} . Ostatecznie na podstawie tej macierzy wyznacza cenę opcji oraz greckie parametry dla konkretnej ceny aktywa (s_{start}) i konkretnego volatylity ($v_{start} = vol^2$).

HestonBarrierIn – funkcja wyceniająca opcje typu call down-and-in oraz put up-and-in. Korzysta jedynie z parytetu, że suma opcji in oraz out to zwykła opcja waniliowa.

5 Analiza poprawności otrzymywanych wyników

Analiza wyników dla opcji waniliowych ze stałą wariancją wykazuje poprawną estymację wszystkich parametrów (z dokładnością do 0.001) poza parametrem gamma który może się odchyłać nawet o 0.1. Podobnie poprawne wydają się wyniki dla opcji barierowych przy stałej wariancji. Dla wariancji zgodnej z modelem Hestona trudniej weryfikować wyniki, jednak porównanie do modelu Monte-Carlo świadczy o tym, że z dużym prawdopodobieństwem wyniki te są także poprawne.