

Raport i dokumentacja
do projektu

Wycena opcji azjatyckich metoda PDE

Autor: Agnieszka Ulikowska

1. TEORETYCZNY OPIS ZAGADNIENIA

Wypłata z opcji azjatyckiej zależy od ceny instrumentu podstawowego w ustalonych momentach czasu. W ramach projektu rozpatrzone zostały następujące opcje azjatyckie o wypłatach:

- Opcja floating strike call $X_T = \max(S_T - \hat{S}_T, 0)$
- Opcja floating strike put $X_T = \max(\hat{S}_T - S_T, 0)$
- Opcja fixed strike call $X_T = \max(\hat{S}_T - K, 0)$
- Opcja fixed strike put $X_T = \max(K - \hat{S}_T, 0)$

gdzie $\hat{S}_T = \frac{1}{T} \int_0^T S_u du$

Cena $V = V(t, S_t, A_t)$ opcji azjatyckiej zależy od trzech zmiennych.

- t - czas
- S_t - proces cen akcji dany wzorem $S_t = S_0 \exp((r - \frac{1}{2}\sigma^2)t - \sigma W_t)$
- A_t - całka z funkcji procesu S_t , $A_t = \int_0^t f(S_u, u) du$

Dla opcji azjatyckich fixed / floating strike $f(S_t, t) = S_t$, $A_t = \int_0^t S_u du$.

Wiedząc, że $dA_t = f(S_t, t) dt$ i korzystając ze wzoru Ito zastosowanego do funkcji $V = V(t, S_t, A_t)$ otrzymujemy równanie różniczkowe

$$dV_t = \left(\frac{dV}{dt} + \mu S \frac{dV}{dS} + \frac{1}{2} \sigma^2 S^2 \frac{d^2 V}{dS^2} + f(S, t) \frac{dV}{dA} \right) dt + \sigma S \frac{dV}{dS} dW_t$$

Używając argumentów takich jak w przypadku równania Blacka-Scholesa dla europejskich opcji waniliowych dowodzimy, że

$$\frac{dV}{dt} + rS \frac{dV}{dS} + \frac{1}{2} \sigma^2 S^2 \frac{d^2 V}{dS^2} + f(S, t) \frac{dV}{dA} - rV = 0$$

Obecność składnika $f(S, t) \frac{dV}{dA}$ uniemożliwia transformację do równania dyfuzji. Rozwiązanie powyższego równania różniczkowego zdefiniowane jest na 3-wymiarowej przestrzeni $S > 0, A > 0, 0 \leq t \leq T$, co prowadzi do znacznych kosztów przy numerycznym rozwiązywaniu. W konkretnych przypadkach można jednak zredukować wymiar równania (do równania dwuwymiarowego) przez odpowiednią zamianę zmiennych – taka zamiana zmiennych jest możliwa przy wycenie opcji floating / fixed strike.

*OPCJA AZJATYCKA FLOATING STRIKE CALL

Wyplata z opcji floating strike call wynosi $X_T = \max(S_T - \hat{S}_T, 0) = S_T * \max(1 - \frac{1}{T * S_T} \int_0^T S_u du)$.

Definiujemy nową zmienną $R_t = \frac{1}{S_t} \int_0^t S_u du$.

Wyplata można wtedy zapisać jako $S_T * \max(1 - \frac{1}{T} R_T, 0) = S_T * \text{function}(R_T, T)$.

Jest to motywacja do szukania rozwiązania $V = V(t, S_t, A_t)$ w postaci $S_t * H(R_t, t)$, gdzie $H(R, t)$ jest pewną nieznaną funkcją.

Po przekształceniach otrzymujemy następujące równanie różniczkowe z warunkami terminowo-brzegowymi (pełne wyprowadzenie dostępne w „Tools for Computational Finance”, Seydel rozdział 6):

$$\frac{dH}{dt} + \frac{1}{2} \sigma^2 R^2 \frac{d^2 H}{dR^2} + (1 - rR) \frac{dH}{dR} = 0$$

$$H = 0 \text{ dla } R \rightarrow \infty$$

$$\frac{dH}{dt} + \frac{dH}{dR} = 0 \text{ dla } R = 0$$

$$H(R_T, T) = \max(1 - \frac{R_T}{T}, 0)$$

Rozwiązując to równanie numerycznie otrzymujemy wartości funkcji $H(R, t)$ - w rzeczywistości mamy obliczoną także cenę opcji.

*OPCJA AZJATYCKA FIXED STRIKE CALL

Wyplata z opcji azjatyckiej fixed strike call wynosi:

$$X_T = \max(\hat{S}_T - K, 0) = -\min(K - \hat{S}_T, 0) = S_T * (-\min(\frac{K - \frac{1}{T} \int_0^T S_u du}{S_T}, 0))$$

Jak poprzednio, definiujemy nową zmienną $Y_t = \frac{K - \frac{1}{T} \int_0^t S_u du}{S_t}$.

Wyplata można zapisać jako $X_T = S_T * (-\min(Y_T, 0)) = S_T * \text{function}(Y_T, T)$. Jest to motywacja do szukania rozwiązania $V = V(t, S_t, A_t)$ w postaci $S_t * H(Y_t, t)$ dla pewnej funkcji H . Można pokazać (szczegóły w artykule B.Alziary et al. / Journal of Banking & Finance 21 (1997) 613-640), że H jest rozwiązaniem równania różniczkowego:

$$\frac{dH}{dt} + \frac{1}{2} \sigma^2 Y^2 \frac{d^2 H}{dY^2} - \left(\frac{1}{T} + rY \right) \frac{dH}{dY} = 0$$

$$H(Y, T) = -\min(Y, 0) .$$

2. NUMERYCZNE ROZWIĄZANIE

Do rozwiązania równań użyłam schematów:

- schemat Laxa – Wendroffa
- schemat TVD

Przy wyborze schematów numerycznych do rozwiązywania powyższych równań, należy zwrócić szczególną uwagę na sposób przybliżenia pierwszej pochodnej zmiennej przestrzennej. Jako modelowe równanie przyjmijmy

$$\frac{du}{dt} + a(x) \frac{du}{dx} = b^2(x) \frac{d^2 u}{dx^2}$$

Stosując standardowy schemat drugiego rzędu FTCS (Forward Time Centered Scheme), tzn. przybliżając pierwszą pochodną za pomocą różnic centralnych możemy narazić się na problemy ze stabilnością i oscylacje numerycznego rozwiązania, w szczególności przy niewielkiej wartości współczynnika $b(x)$. Przy zerowym współczynniku dyfuzji w ogóle nie należy stosować powyższego schematu. Przyczyną są warunki gwarantujące stabilność nakładające ograniczenia na wielkość kroku czasowego i przestrzennego:

$$0 \leq \lambda \leq 0.5, \quad \lambda \left(\frac{a \Delta x}{b} \right)^2 \leq 2, \quad \text{gdzie } \lambda = \frac{b \Delta t}{\Delta x^2} .$$

Przy wyborze $\lambda = \frac{1}{2}$, ograniczenie na krok siatki przestrzennej przybiera formę $\Delta x \leq \frac{2b}{|a|}$.

Uwaga: Ze względu na istotność problemu przybliżania pierwszej pochodnej zmiennej przestrzennej, przy analizie schematów rozważane będą równania postaci:

$$(2.1) \quad \frac{du}{dt} + \frac{d}{dx} f(x) = 0 .$$

Druga pochodna zmiennej przestrzennej przybliżana jest w programie w sposób standardowy.

Oznaczenia: $u_{j,v}$ oznacza numeryczne rozwiązanie w punkcie (j, v) . Jest to aproksymacja rzeczywistego rozwiązania $u(x_j, t_v)$.

*SCHEMAT LAXA – WENDROFF'A

Schemat Laxa-Wendroff'a oparty jest na wzorze Taylora i postaci równania (2.1) .

$$u_{j,v+1} = u_{j,v} + \Delta t \frac{du_{j,v}}{dt} + O(\Delta t^2) = u_{j,v} - \Delta t \frac{df(u_{j,v})}{dt} + O(\Delta t^2)$$

Wykorzystuje także wartości obliczane w punktach nie będących węzłami siatki czasowo-przestrzennej, tzn. w punktach $(j \pm \frac{1}{2}, v \pm \frac{1}{2})$.

Formalnie, schemat można zapisać jako:

$$u_{j+\frac{1}{2},v+\frac{1}{2}} = \frac{1}{2}(u_{j,v} + u_{j+1,v}) - \frac{\Delta t}{2\Delta x}(f(u_{j+1,v}) - f(u_{j,v}))$$

$$w_{j-\frac{1}{2},v+\frac{1}{2}} = \frac{1}{2}(u_{j-1,v} + u_{j,v}) - \frac{\Delta t}{2\Delta x}(f(u_{j,v}) - f(u_{j-1,v}))$$

$$u_{j,v+1} = u_{j,v} - \frac{\Delta t}{\Delta x}(f(u_{j+\frac{1}{2},v+\frac{1}{2}}) - f(u_{j-\frac{1}{2},v+\frac{1}{2}}))$$

*METODY TVD

Idea metod TVD: schemat numeryczny musi gwarantować, że całkowita wariacja rozwiązania nie będzie zwiększać się w miarę upływu czasu. Numeryczne rozwiązanie musi zatem spełniać poniższy warunek:

$$TV(u^{n+1}) \leq TV(u^n), \quad \text{gdzie } TV(u^n) = \sum_j |u_{j+1,v} - u_{j,v}|$$

Metoda zaimplementowana w projekcie oparta jest na odpowiednim połączeniu schematów pierwszego i drugiego rzędu.

Wyjściowe równanie (2.1) przedstawiamy jako:

$$\frac{du_i}{dt} + \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{\delta x} = 0, \quad \text{gdzie } f_{i+\frac{1}{2}} = f_{i+\frac{1}{2}}^L + \Phi_{i+\frac{1}{2}} * (f_{i+\frac{1}{2}}^H - f_{i+\frac{1}{2}}^L)$$

($f_{i-\frac{1}{2}}$ zdefiniowane jest analogicznie, z odpowiednio przesuniętym indeksem i). Występujące powyżej składniki $f_{i\pm\frac{1}{2}}$ to tzw. *numerical fluxes*, które powstają poprzez odpowiednie połączenie *flux'ów* niższego i wyższego rzędu. Dla ustalenia uwagi przyjmijmy, że równanie ma postać $\frac{du}{dt} + a \frac{du}{dx} = 0$, gdzie $a > 0$.

Najczęściej przyjmuje się, że:

$$f_{i+\frac{1}{2}}^L = au_i \quad \text{oraz} \quad f_{i+\frac{1}{2}}^H = a \frac{u_{i+1} + u_i}{2} . \quad \text{Po podstawieniu otrzymujemy:}$$

$$f_{i+\frac{1}{2}} = au_i + \frac{a}{2} \Phi_{i+\frac{1}{2}} * (u_{i+1} - u_i) .$$

Funkcja Φ (flux limiter) powinna przyjmować wartości bliskie 0, gdy numeryczne rozwiązanie wykazuje duże skoki gradientu (wtedy schemat jest pierwszego rzędu) i wartości bliskie 1 w przeciwnym przypadku (wtedy schemat jest drugiego rzędu). W przypadku $a > 0$ można zdefiniować zależność $\Phi_{i+\frac{1}{2}} = \Phi(r_i)$, gdzie $r_i = \frac{u_i - u_{i-1}}{u_{i+1} - u_i}$. Współczynnik ten jest ujemny w pobliżu lokalnych ekstremów, stosunkowo niewielki dla gładkich danych i zwiększa się, gdy numeryczne rozwiązanie wykazuje znaczne i gwałtowne zmiany wartości.

Można stosować różne limity, w programie zostały zaimplementowane cztery rodzaje:

$$\text{MinMod: } \Lambda(a, b) = S(a, b) \min(|a|, |b|)$$

$$\text{Van Leer: } \Lambda(a, b) = S(a, b) \frac{2|ab|}{|a| + |b|} , \text{ gdzie}$$

$$\text{MC: } \Lambda(a, b) = S(a, b) \min(0.5|a+b|, 2|a|, 2|b|)$$

$$\text{Superbee: } \Lambda(a, b) = S(a, b) \max(\min(2|a|, |b|), \min(2|b|, |a|))$$

$$S(a, b) = \frac{1}{2} (\text{sign}(a) + \text{sign}(b)), \quad \Lambda(1, r) = \Phi(r) .$$

Zalety metody:

1. zapobiega powstawaniu oscylacji numerycznego rozwiązania
2. łączy w sobie zalety schematów pierwszego i drugiego rzędu
3. prosta idea

3.WYNIKI

*WYCENA OPCJI FIXED STRIKE CALL

cena spot S:	Monte Carlo	PDE – metoda TVD	PDE – Lax_Wendorff
85	2,71	2,70	2,59
90	5,19	5,10	4,88
95	8,55	8,29	8,07
100	12,60	12,27	11,91
105	17,07	16,81	16,26
110	21,78	21,63	20,88
115	26,59	26,52	25,59
120	31,45	31,41	30,33

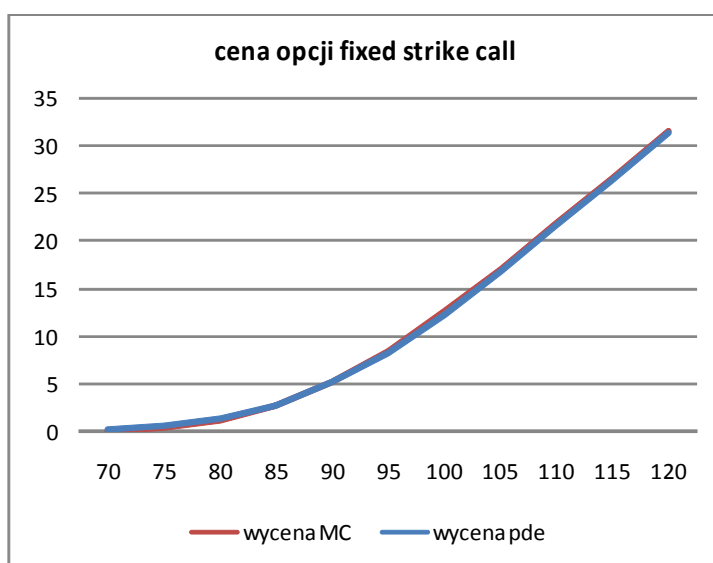
Parametry:

$$K = 90$$

$$r = 0.05$$

$$\sigma = 0.2$$

$$T = 1$$



cena spot S:	Monte Carlo	PDE – metoda TVD	PDE – Lax_Wendorff
85	8,44	8,44	8,13
90	11,08	11,05	10,65
95	14,09	14,04	13,54
100	17,43	17,37	16,78
105	21,06	21,02	20,31
110	24,93	24,84	24,03
115	29,02	28,98	28,05
120	33,29	33,22	32,19

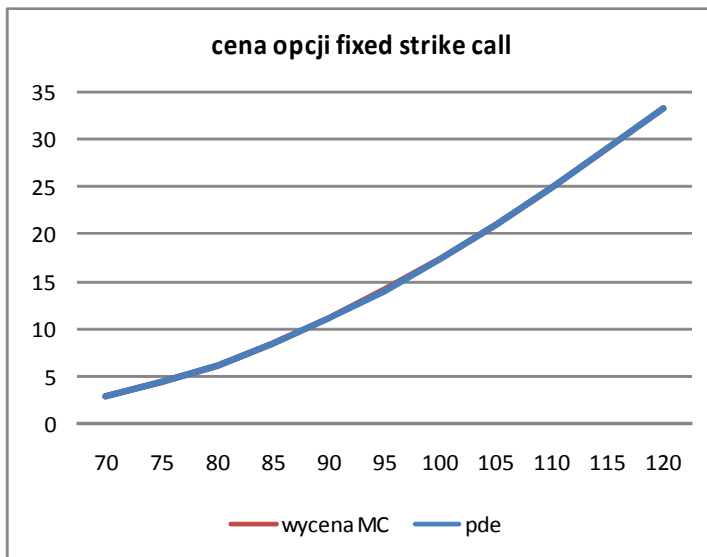
Parametry:

$$K = 90$$

$$r = 0.05$$

$$\sigma = 0.5$$

$$T = 1$$

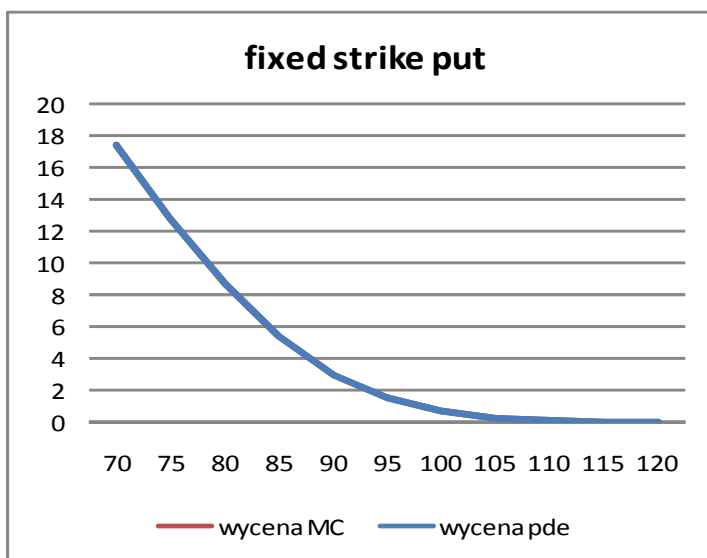


***WYCENA OPCJI FIXED STRIKE PUT**

Parametry:

$K = 90$
 $r = 0.05$
 $\sigma = 0.2$
 $T = 1$

cena S0	Monte Carlo	PDE – metoda TVD	PDE – Lax_Wendorff
70,00	17,43	17,52	17,47
75,00	12,85	13,02	12,91
80,00	8,75	8,91	8,8
85,00	5,41	5,47	5,26
90,00	3,01	2,91	2,69



cena S0	Monte Carlo	PDE – metoda TVD	PDE – Lax_Wendorff
70	20,21	20,25	20,14
75	16,79	16,92	16,64
80	13,76	13,82	13,54
85	11,14	11,12	10,72
90	8,91	8,84	8,45

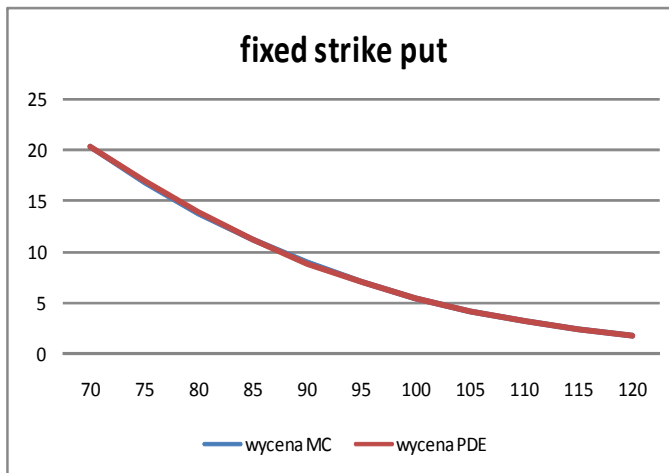
Parametry:

$$K = 90$$

$$r = 0.05$$

$$\sigma = 0.5$$

$$T = 1$$



4.LITERATURA

* "High-resolution FEM-TVD schemes base on a fully multidimensional flux limiter", D.Kuzmin & S.Turek, Institute of Applied Mathematics, University of Dortmund, rozdział 2: One dimensional TVD schemes.

* "Comparison of sinite difference methoda for the numerical simulation of reacting flow", K. Alhumaizi, Computers & Chemical engineering 28 (2004), 1579 – 1769

* "Tools for computational finance", R.U. Seydel

5.DOKUMENTACJA

DANE W PLIKU wejściowym:

Dane w pliku powinny mieć następującą kolejność:

- > rodzaj opcji
- > rodzaj schematu
- > stopa %
- > zmienność
- > cena spot
- > strike
- > liczba punktów siatki przestrzennej
- > liczba punktów siatki czasowej
- > T_min = 0

-> termin zapadalności
-> $R_{min} = 0$
-> maksymalna wartość siatki przestrzennej. Jeśli 0, to zostaną ustawione wartości domyślne.
-> wielkość epsilon (przy liczeni pochodnych)

OZNACZENIA i DOSTEPNE FUNKCJE:

M - ilosc wezlow siatki przestrzennej
N - ilosc wezlow siatki czasowej
siatka_R - siatka przestrzenna
siatka_R2 - siatka przerstrzenna pomocnicza
siatka_t - siatka czasowa
delta_R - odstep miedzy punktami siatki przestrzennej
delta_t - odstep miedzy punktami siatki czasowej
wsp_delta - wspolczynnik pomocniczy. $wsp_delta = delta_t / delta_x$
wsp_delta2 - wspolczynnik pomocniczy. $wsp_delta2 = delta_t / delta_x^2$
wsp_a - wspolczynnik rownania rozniczkowego
wsp_a_HalfStep - wspolczynnik rownania rozniczkowego
wsp_b - wspolczynnik rownania rozniczkowego
wsp_V - wspolczynnik rownania rozniczkowego
gamma - wspolczynnik pomocniczy. $gamma = wsp_a * wsp_delta$
lambda - wspolczynnik pomocniczy. $lambda = wsp_b * wsp_delta2$
beta - wspolczynnik pomocniczy. $beta = delta_R * (wsp_a./wsp_b)$
Boundary_T0 - warunek koncowy
lo_BC_0 - warunek brzegowy
hi_BC_R - warunek brzegowy
Option - rodzaj opcji
R_min - minimalna wartosc w siatce przestrzennej
R_max - maksymalna wartosc w siatce przestrzennej
T_min - minimalna wartosc w siatce czasowej. $T_{min} = 0$
T_max - maksymalna wartosc w siatce czasowej
r - stopa procentowa
sigma - zmienosc
S0 - cena spot
K - strike
scheme - rodzaj schematu

```
function [y] = SynchSigns(a)
#funkcja sprawdza czy liczby/wspolrzedne wektorow sa jednakowych znakow. Jesli
jednakowych znakow i dodatnie zwraca 1, jesli jednakowych znakow i ujemne zwraca
-1, w przeciwnym przypadku 0.
#na wejsciu: macierz wymiaru M na 2. W kolumnach porownywane liczby/wektory
#przykladowe wywolanie:
> a = [2,-1;0,1];
> SynchSigns(a);
```

```
function [y] = MinMod(a)
#rodzaj LIMITERA
#na wejsciu: macierz wymiaru M na 2. W kolumnach liczby/wektory
#przykladowe wywolanie:
> a = [2,-1;0,1];
> MinMod(a);
```

```
function [y] = MC(a)
#rodzaj LIMITERA
#na wejsciu: macierz wymiaru M na 2. W kolumnach liczby/wektory
#przykladowe wywolanie:
> a = [2,-1;0,1];
> MC(a);
```

```
function [y] = VanLeer(a)
#rodzaj LIMITERA
```

```

#na wejsciu: macierz wymiaru M na 2. W kolumnach liczby/wektory
#przykladowe wywołanie:
> a = [2,-1;0,1];
> VanLeer(a);

function [y] = superbee(a)
#rodzaj LIMITERA
#na wejsciu: macierz wymiaru M na 2. W kolumnach liczby/wektory
#przykladowe wywołanie:
> a = [2,-1;0,1];
superbee(a);

function [R_max] = SetMaximalGridPoint(S0, K, T_max, Option)
#wstawia odpowiednia siatke przestrzenna w zaleznosci od rodzaju opcji

function [Boundary_T0, lo_BC_0, hi_BC_R] = SetBoundaryConditions(M, N, r, K,
T_max, siatka_t, siatka_R, Option)
#ustawia warunki poczatkowo-brzegowe w zaleznosci o rodzaju opcji

function [wsp_a, wsp_a_HalfStep, wsp_b, wsp_V, gamma, lambda, beta] =
Coefficients(r, sigma, T_max, siatka_R, siatka_R2, wsp_delta, Option)
#oblicza wspolczynniki rownania rozniczkowego

***
function [V] = LaxWendroff(M, N, delta_t, wsp_a, wsp_a_HalfStep, wsp_b, wsp_V,
wsp_delta, lambda, Boundary_T0, hi_BC_R, lo_BC_0, Option)

#implementacja schematu Laxa-Wendroffa
#opis schematu w dokumencie Ulikowska_opis_problemu_i_algorytmow.txt

***
function [V] = Upwind(M, N, delta_t, wsp_a, wsp_V, wsp_delta, gamma, lambda,
Boundary_T0, hi_BC_R, lo_BC_0, Option)

#implementacja schematu Upwind. Uproszczona wersja funkcji HighResolution [gdy
theta jest rowne zero]
#opis schematu w dokumencie Ulikowska_opis_problemu_i_algorytmow.txt

***
function [V] = HighResolution(M, N, delta_t, wsp_a, wsp_V, wsp_delta, gamma,
lambda, Boundary_T0, hi_BC_R, lo_BC_0, scheme, Option)

#implemetacja metod HighResolution
#zmienna IsPositive: okresla znak wspolczynnika przy pochodnej przestrzennej
#zmienna slope: okresla stosunek gradientow w trzech kolejnych punktach siatki
przestrzennej. slope = [w(j,v) - w(j-1,t)]/[w(j+1,t) - w(j,t)]
#zmienna slope_final_ones: zmienna pomocnicza
#zmienna theta_r: w zaleznosci od stosunku gradientow zmienna przyjmuje
wartosci:
# -> bliskie 0 w okolicach znacznego stosunku gradientow [gdy theta = 0 schemat
upwind]
# -> bliskie 1 gdy numeryczne rozwiazanie jest dostatecznie gladkie [gdy theta
= 1 schemat roznic centralnych]
#theta zalezy takze od wyboru LIMITERA
#zmienna FLUX: "numerical flux"
#opis schematu w dokumencie Ulikowska_opis_problemu_i_algorytmow.txt

***
function [floating_put] = FloatingPutPrice(floating_call, S0, T_max, r)
#oblicza cene opcji floating strike PUT jesli znana jest cena opcji call
(floating_call)

function [fixed_put] = FixedPutPrice(fixed_call, S0, T_max, r, K)
#oblicza cene opcji fixed strike PUT jesli znana jest cena opcji call

```

```
(fixed_call)
```

```
***
```

```
function [price] = OptionPrice(S0, K, V, R_min, R_max, T, T_max, r, K, Option)
#oblicza wartosc opcji na podstawie macierzy V(j, t)
#wartosci V(j, T_max) znajduja sie w ostaniej kolumnie
#jesli wartosc opcji ma byc obliczona dla t < T_max, to nalezy uzyc parametru T.
Moment wyceny, T_max i T powiazane sa rownaniem:  $t = T_{max} - \text{delta}_t * T$ 
```

```
***
```

```
function [price] = StartCalculatingOptionPrice(Option, scheme, r, sigma, S0, K,
M, N, T_min, T_max, R_min, R_max, h)
#nadzowruje dzialanie innych funkcji: funkcja "porzadkowa" ktora wywoluje
kolejno podrzedne funkcje
```