Dokumentacja
Wycena opcji amerykańskich za pomocą równań różniczkowych cząstkowych

Katarzyna Derlatka

19 sierpnia 2011

Spis treści

Treść zadania

Celem projektu jest wycena amerykańskich opcji, w tym barierowych, z arbitralne zadaną funkcją wypłaty oraz prawie arbitralnie dobraną barierą. Wycena nastąpi poprzez rozwiązanie równania różniczkowego cząstkowego Blacka-Scholesa. Dodatkowo policzone zostaną współczynniki wrażliwości: Delta, Gamma, Theta i Vega.

Poza amerykańskimi opcjami waniliowymi rozpatrzone będę amerykańskie opcje barierowe typu:

- down-and-out z wypłatą $X_t = f(S_t)\,\mathbb{1}_{\left\{\forall_{u\leqslant t}\, S_u > B_{lower}\right\}}$
- up-and-out z wypłatą $X_t = f(S_t)\,\mathbb{1}_{\left\{\forall_{u\leqslant t}\, S_u < B_{upper}\right\}}$
- down-and-in z wypłatą $X_t = f(S_t)\,\mathbb{1}_{\left\{\exists_{u\leqslant t}\, S_u < B_{upper}\right\}}$
- up-and-in $X_t = f(S_t)\,\mathbb{1}_{\left\{\exists_{u\leqslant t}\, S_u > B_{lower}\right\}}$
- double-knock-out $X_t = f(S_t)\,\mathbb{1}_{\left\{\forall_{u\leqslant t}\, B_{lower} < S_u < B_{upper}\right\}}$
- double-knock-in $X_t = f(S_t)\,\mathbb{1}_{\left\{\exists_{u\leqslant t}\, S_u < B_{lower} \text{ lub } S_u > B_{upper}\right\}}$

gdzie f jest funkcją wypłaty opcji waniliowej, a B_{lower} i B_{upper} to bariery.
W przypadku opcji typu 'knock-in', program wyceni tylko te opcje, które w chwili końcowej mają niezerowe wypłaty, tzn. dla opcji 'knock-in' call $K < B$, dla opcji 'knoci-in' put $K > B$.

Równanie Blacka-Scholesa i problem granic swobodnych

Zdyskontowana cena instrumentu pochodnego typu europejskiego jest martyngałem w mierze wolnej od ryzyka. Stąd, stosując wzór Ito, można wyprowadzić równanie różniczkowe cząstkowe Blacka-Scholesa

$$\frac{\partial V(s,t)}{\partial t} + rs\frac{\partial V(s,t)}{\partial s} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 V(s,t)}{\partial s^2} - rV(s,t) = 0 \tag{1}$$

W przypadku instrumentu typu amerykańskiego cena jest rozwiązaniem następującego problemu o granicach swobodnych (free-boundary problem, linear complementarity problem LCP)

$$\begin{cases} \text{Spełnione jest równanie (1) lub } V(s,t) = f(s,t) \\ V(s,t) \geqslant f(s,t) \\ \frac{\partial V(s,t)}{\partial t} + rs\frac{\partial V(s,t)}{\partial s} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 V(s,t)}{\partial s^2} - rV(s,t) \leqslant 0 \\ \text{odpowiednie warunki końcowe i brzegowe, wyznaczone na podstawie własności } f(s,t) \end{cases}$$

W szczególności dla opcji barierowych (X - bariera) warunki brzegowe mają postać

- dla opcji 'down-and-out'

  $$\text{warunek brzegowo-końcowy dla } V(S,t) = \begin{cases} 0 & , S = X \\ \text{warunek brzegowo-końcowy dla opcji waniliowej} & , S > X \end{cases}$$

- dla opcji 'up-and-out'

  $$\text{warunek brzegowo-końcowy dla } V(S,t) = \begin{cases} 0 & , S = X \\ \text{warunek brzegowo-końcowy dla opcji waniliowej} & , S < X \end{cases}$$

- dla opcji 'down-and-in'
  - warunek końcowy $V(S,T) = 0$ dla $S > X$.
  - warunek brzegowy

  $$\begin{array}{ccc} V(S,t) & \overset{S\to\infty}{\longrightarrow} & 0 \\ V(X,t) & = & \text{cena opcji waniliowej} \end{array}$$

- dla opcji 'up-and-in'
  - warunek końcowy $V(S,T) = 0$ dla $S < X$.
  - warunek brzegowy

  $$\begin{array}{ccc} V(S,t) & \overset{S\to 0}{\longrightarrow} & 0 \\ V(X,t) & = & \text{cena opcji waniliowej} \end{array}$$

W przeciwieństwie do równań dla opcji europejskich, nie ma analitycznych wzorów na wycenę barierowych opcji amerykańskich. Powyższe układy wymagają rozwiązania numerycznego. Standardowe metody rozwiązywania równań różniczkowych cząstkowych nie dają dobrych rezultatów dla LCP (zbieżność pierwszego rzędu, duża złożonośc obliczeniowa). Z tego powodu do wyceny opcji amerykańskich zastosowałam tak zwaną metodę współczynnika kary (penalty method), polegającej na przekształceniu równania (1) do postaci

$$\frac{\partial V(s,t)}{\partial t} + rs\frac{\partial V(s,t)}{\partial s} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 V(s,t)}{\partial s^2} - rV(s,t) + \rho \max\left(f(s,t) - V(s,t)\right) = 0 \tag{2}$$

gdzie ρ jest dodatnim współczynnikiem kary. Można pokazać, że dla $\rho \to \infty$ rozwiązanie powyższego równania spełnia $V \geqslant f$. Metoda ta szczególnie przyśpiesza rozwiązywanie LCP, gdy znany jest dokładny warunek końcowy, jak w przypadku opcji. Ponadto, można ją uogólnić również na przypadek wielowymiarowy. Wadą jest natomiast zbieżność tylko pierwszego rzędu. Można jednak poprawić zbieżność, stosując zmienny krok czasowy. Niestety szybkość i dokładność rozwiązania zależy od dobranych dodatkowych trzech parametrów.

Numeryczne rozwiązanie

Do rozwiązania równania różniczkowego użyłam schematów różnic skończonych

- implicit - niejawna metoda Eulera ($\theta = 0$)
- schemat Cranka - Nicolson ($\theta = \frac{1}{2}$)

W obu przypadkach zastosowałam jednorodną siatkę w przestrzeni. Wyjątek stanowią opcje barierowe typu 'knock-in', gdzie sztucznie powiększyłam wektor przestrzenny o ewentualne bariery. Dla wymiaru czasowego zaimplementowałam dwie wersje: jednorodną oraz ze zmiennym (ale obustronnie ograniczonym) krokiem.

Po zastosowaniu schematu różnic skończonych równanie (2) sprowadza się do

$$\begin{aligned} V_i^{n+1} - V_i^n = (1-\theta)\left(\Delta\tau \sum_{j=i\pm1} \left(V_j^{n+1} - V_i^{n+1}\right) - r\Delta\tau V_i^{n+1}\right) \\ + \theta * \left(\Delta\tau \sum_{j=i\pm1} \left(\gamma_{ij} + \beta_{ij}\right)\left(V_j^n - V_i^n\right) - r\Delta\tau V_i^n\right) \\ + P_i^{n+1}\left(f(S_i) - V_i^{n+1}\right) \end{aligned} \tag{3}$$

,gdzie $\tau = T - t$ oraz

$$P_i^{n+1} = \begin{cases} Large & , V_i^{n+1} < f(S_i), \\ 0 & , wp.p. \end{cases}$$

$$\gamma_{ij} = \frac{\sigma^2 S_i^2}{|S_j - S_i|\left(S_{i+1} - S_{i-1}\right)}$$

$$\beta_{ij} = \begin{cases} \frac{rS_i}{S_{i+1} - S_{i-1}} & , 2\frac{\sigma^2 S_i^2}{|S_j - S_i|} + rS_i \geqslant 0, \\ \\ 2\frac{max(0, rS_i)}{S_{i+1} - S_{i-1}} & , wp.p. \end{cases}$$

Zauważmy, że w tak zapisanym schemacie można zastosować zmiennym krok czasowy lub przestrzenny.

Forsyth i Vetzal w [1] dowiedli następujące twierdzenie o zbieżności penalty method.

Twierdzenie. *Jeżeli*

$$\gamma_{ij} + \beta_{ij} \geqslant 0,$$

$$2 - \theta\left(\Delta\tau \sum_{j=i\pm1} \left(\gamma_{ij} + \beta_{ij}\right) + r\Delta\tau\right) \geqslant 0,$$

$$\frac{\Delta\tau}{\Delta S} < const.$$
$$\Delta\tau, \Delta S \longrightarrow 0 \text{ , } gdzie\ \Delta S = min_i\left(S_{i+1} - S_i\right),$$

to schemat (3) rozwiązuje zagadnienie LCP dla opcji amerykańskiej z osłabionym warunkiem

$$V_i^{n+1} - f\left(S_i\right) \geqslant -\frac{C}{Large},$$
$$|V_i^{n+1} - f\left(S_i\right)| \leqslant \frac{C}{Large} \text{ ,gdy nie jest spełnione równanie (1),}$$

gdzie C - stała niezależna od Large, $\Delta\tau$, ΔS.

Adaptive mesh method

Zbieżność drugiego rzędu można zapewnić stosując zmienny krok przestrzenny. W metodzie adaptive mesh nowy krok przestrzenny wylicza się w zależności od pewnej funkcji monitorującej, która dla schematu różnic skończonych ma postać $\tilde{V}\left(S,t\right) = |V^{(3)}|^{\frac{1}{2}}$, oraz zależnej od niej funkcji oceniającej postaci $\xi\left(S,t\right) = \frac{\int_0^S \tilde{V} dS}{\int_0^{S_\infty} \tilde{V} dS}$.

Algorytm adaptacyjny tak wybiera punkty, by $\xi\left(S_i,t\right) \approx \frac{i}{n}$. W tym celu zmieniamy rozmieszczenie węzłów, gdy ten warunek nie jest spełniony (z pewną dokładnością zapewnianą przez współczynnik α). Dokładniej, sprawdzamy czy $rdrift = max_i \left\{ \int_{S_{i-1}^j}^{S_i^j} \tilde{V} dS \right\} / \int_0^{S_\infty} \tilde{V} dS$ jest większe niż α/n. Jeśli tak, poprawiamy rozkład węzłów zgodnie z metodą Newtona, tj.

$$S_i^{j,nowe} = S_i^{j,poprz} - \frac{\xi\left(S_i^{j,poprz}\right) - \frac{i}{n}}{\xi'\left(S_i^{j,poprz}\right)} \approx S_i^{j,poprz} - \frac{\int_0^{S_i^{j,poprz}} \tilde{V}^{j,poprz} dS - \frac{i}{n}\int_0^{S_\infty} \tilde{V}^{j,poprz} dS}{\tilde{V}^{j,poprz}} \tag{4}$$

$xmin$ i $xmax$ pozostawiamy bez zmian, powyższa zmiana dotyczy pozostałych węzłów.

Algorytm adaptacyjny - krok z t_{j-1} do t_j

1. Stosując iterację penalty oblicz V^j. Jeśli $j < 10$, idź do następnego kroku czasowego (czas na wygładzenie).
2. Sprawdź czy spełniony jest warunek $rdrift < \alpha/n$.
3. Jeśli tak, idź do następnego kroku czasowego
4. Jeśli nie, zastosuj (4) oraz
   - Jeśli $j < b + 10$, interpoluj V^{j-1} na nową siatkę przestrzenną i za pomocą iteracji penalty oblicz nowe V^j.
   - Jeśli $j \geqslant b$, interpoluj pierwotne V^j na nową siatkę przestrzenną.

Metodę Newtona możemy stosować pod warunkiem, że pochodna jest niezerowa, czyli $\tilde{V}^j <> 0$, ale dla opcji mocno out-of-the-money lub mocno in-the-money już $V^{(1)}$ jest praktycznie zerowa. Dlatego powyższy algorytm należy poprawić, na przykład poprzez:

1. przed pierwszą zmianą zgodnie z (4) skupienie jednorodnej siatki(poza krańcami) na zbiorze $\{x : \xi\left(x\right) > tol1,\ \xi\left(x\right) < 1 - tol2\}$
2. w każdym kroku sprawdzenie, czy punkty nie zostały przesunięte w niepożądany obszar i ewentualne poprawienie jak w punkcie 1.

Dokumentacja programu

american_PDE

Główną funkcją obliczającą cenę amerykańskiej opcji z ewentualną barierą jest

american_PDE(S0, K, r, T, sigma, Mx, Mt, xmin, xmax, scheme, opcja, Large, tol, B1, B2, typ)

bold Algorytm nie wycenia opcji down-and-in call dla $K > B1$, opcji up-and-in put dla $K < B2$. Argumenty funkcji

- S0 - cena początkowa instrumentu bazowego
- K - cena wykonania opcji
- r - ciągła stopa procentowa wolna od ryzyka (w skali roku)
- T - czas życia opcji liczony w latach
- sigma - zmienność instrumentu bazowego
- Mx - liczba punktów na siatce przestrzennej
- Mt - liczba punktów na siatce czasu
- xmin - dolna granica cen instrumentu bazowego
- xmax - górna granica cen instrumentu bazowego
- scheme - wybór schematu metody różnic skończonych; 1 dla schematu niejawnego (implicit), 2 dla schematu Cranka - Nicolson
- opcja - wybór rodzaju opcji; 1 dla opcji call, 2 dla opcji put
- tol - współczynnik tolerancji błędu; domyślna wartość $1e-6$
- Large - współczynnik kary; domyślna wartość $1e6$ (zgodnie z [1] sugerowana wartość: $1/tol$, gdyż dla węzłów, w których $V_i^{n+1} - f(S_i) < 0$, zachodzi $|V_i^{n+1} - f(S_i)| \approx \frac{|f(S_i)|}{Large}$)
- B1 - dolna bariera; dla opcji waniliowych wartość pomijana przez funkcję, domyślnie: xmin
- B2 - górna bariera; dla opcji waniliowych wartość pomijana przez funkcję, domyślnie: xmax
- typ - typ bariery; możliwe wartości: 'up-and-out', 'down-and-out', 'up-and-in', 'down-and-in', 'double-knock-out', 'double-knock-in'; inna wartość oznacza opcję waniliową

Wartości funkcji

- price - interpolowana cena opcji w punkcie S0; poza węzłami interpolacja splajnami stopnia 3; skalar
- w - wektor cen opcji na przedziale $[xmin, xmax]$; wektor Mx+1 - wymiarowy
- x - wektor cen instrumentu bazowego, którym odpowiadają ceny opcji w; wektor Mx+1 - wymiarowy
- blad - wartość maksymalnego relatywnego błędu ograniczenia amerykańskiego, tzn. $max_{n,i} \frac{max(0, f(S_i) - V_i^n)}{max(1, f(S_i))}$
- UWAGA! W przypadku opcji typu 'knock-in' długości wektorów x i w mogą być większe z powodu wydłużenia x o punkty $B1$ lub $B2$ oraz, dla każdego typu opcji, również o strike K. Natomiast dla opcji 'knock-out' argumenty zostają skrócone do przedziału ograniczonego barierą/barierami

american_PDE_TSEL

Wersja american_PDE ze zmiennym krokiem czasowym (time selector).

american_PDE_TSEL(S0, K, r, T, sigma, Mx, Mt, xmin, xmax, scheme, opcja, Large, tol, B1, B2, typ, t0, dnorm, maxdt, D)

W tej funkcji zastosowałam przekształcenie kroku czasowego $\Delta\tau^{n+2} = \Delta\tau^{n+1} min_i \left(\frac{dnorm}{\frac{|V(S_i,\tau^{n+1})-V(S_i,\tau^n)|}{max(D,|V(S_i,\tau^{n+1})|,|V(S_i,\tau^n)|)}} \right)$.

Argumenty

- jak dla american_PDE
- Mt - maksymalna liczba punktu czasu, powyżej której algorytm wyrzuci błąd
- t0 - początkowy krok czasowy; domyślnie: T/1000
- dnorm - średnia relatywna zmiana kroku czasowego; domyślnie: 0.15
- maxdt - maksymalna wielkość kroku czasowego; domyślnie: T/100
- D - kolejny parametr zabezpieczajacy przed nadmierna liczba kroków czasowych przy małej wartości V. Sugerowana wartość: 1

Wartości funkcji - jak dla american_PDE.

american_adapt_PDE

Wersja american_PDE ze zmiennym krokiem przestrzennym(adaptive mesh).

american_adapt_PDE(S0, K, r, T, sigma, Mx, Mt, xmin, xmax, scheme, opcja, Large, tol, B1, B2, typ, alfa, b);

Argumenty funkcji

- jak dla american_PDE
- alfa - parametr określający czy potrzebna jest zmiana rozmieszczenia, dokładniej w opisie metody.
- b - liczba pętli zwykłego american_PDE przed uruchomieniem metody adaptive mesh.

Wartości funkcji - wektor [Price, w, w1, x, blad, v3]:

- Price, w, x, blad - jak dla american_PDE
- w1 - wektor cen opcji waniliowej (w przypadku wyceny opcji barierowych.
- v3 - trzecia pochodna funkcji wyceniającej na zakończenie obliczeń.

Greki

Skrypt zawiera również funkcje liczące parametry greckie zadanej opcji metodą ilorazu różnicowego centralnego. Wszystkie funkcje liczące pojedynczy współczynnik wrażliwości zwracają wartości postaci [grekaAM, allAM, x], gdzie $grekaAM$ jest wartością parametru w punkcie $S0$, $allAM$ jest wektorem wartości parametru w punktach x (wektor zwracany przez funkcję wyceniającą), co umożliwia narysowanie od razu wykresu danego parametru.

Delta

deltaAM_PDE(S0, dS, K, r, T, sigma, Mx, Mt, xmin, xmax, scheme, opcja, Large, tol, B1, B2, typ)

Argumenty

- jak dla american_PDE
- $\pm dS$ - zmiana ceny instrumenty bazowego

Gamma

gammaAM_PDE(S0, dS, K, r, T, sigma, Mx, Mt, xmin, xmax, scheme, opcja, Large, tol, B1, B2, typ)

Argumenty

- jak dla american_PDE
- $\pm dS$ - zmiana ceny instrumenty bazowego

Rho

rhoAM_PDE(S0, dr, K, r, T, sigma, Mx, Mt, xmin, xmax, scheme, opcja, Large, tol, B1, B2, typ)

Argumenty

- jak dla american_PDE
- $\pm dr$ - zmiana stopy procentowej

Vega

vegaAM_PDE(S0, dsigma, K, r, T, sigma, Mx, Mt, xmin, xmax, scheme, opcja, Large, tol, B1, B2, typ)

Argumenty

- jak dla american_PDE
- $\pm dsigma$ - zmiana zmienności instrumentu bazowego

Theta

thetaAM_PDE(S0, dT, K, r, T, sigma, Mx, Mt, xmin, xmax, scheme, opcja, Large, tol, B1, B2, typ)

Argumenty

- jak dla american_PDE
- $\pm dT$ - zmiana czasu

Greeks

Funkcja licząca wszystkie współczynniki greckie

greeksAM_PDE(S0, dS, dT, dr, dsigma, K, r, T, sigma, Mx, Mt, xmin, xmax, scheme, opcja, Large, tol, B1, B2, typ)

Argumenty

- jak dla poszczególnych współczynników greckich

Wartości funkcji: wektor kolejnych współczynników - [deltaAM, thetaAM, gammaAM, rhoAM, vegaAM]

funkcje pomocnicze

Powyższe funkcje odwołują się do kilku pomocniczych funkcji.

- payoff(S, K, opcja)

Funkcja wypłaty. Przyjmuje i zwraca wektory. Domyślna funkcja - wypłata z opcji waniliowej.

- Value(y,x,w)

Funkcja interpolująca wartości wektora w z x na y. Wektory x i w muszą być tej samej długości. Wektor/skalar y - dowolny.

- out(x, B1, B2, typ)

Funkcja wyliczająca mnożnik wypłaty dla opcji barierowych typu knock-out. Wejście: wektor x, B1 i B2 - bariery, typ - jak w american_PDE.

- check(S0, K, xmin, xmax, typ, B1, B2, opcja)

Funkcja sprawdzająca, czy granice S0 znajduje się w zakresie (nietrywialnej) wyceny. Zwraca błąd dla $S0 \notin [xmin, xmax]$.

- war_brzeg(K, xmin = min(x), xmax = max(x), B1, B2, typ, opcja)

Funkcja wyliczająca warunki brzegowe. Odpowiednio powiązana z funkcją $payoff$.

Testy programu

Numerycznych wyników wyceny amerykańskich opcji barierowych jest w literaturze stosunkowo niewiele. W poniższych przykładach ceny referencyjne zostały zaczerpnięte z

- [1] dla opcji typu knock-out, obliczone algorytmem penalty method
- [6] dla opcji down-and-in put, obliczone zmodyfikowanym algorytmem dwumianowym dla 2000 okresów

down-and-in put

Ceny dla parametrów: K = 100, r = 0.06, sigma = 0.2, T = 0.5, xmin = 0, xmax = 200, Mx = 200, Mt = 200

Źrodło/Metoda	S0	B1	Cena	CPU (sec)	Cena (Mx, Mt = 500)	CPU
[6]	75	70	17.3004		17.3004	
penalty implicit	75	70	17.2817	1	17.2968	13
penalty Crank-Nicolson	75	70	17.2982	1	17.3034	13
[6]	110	90	1.2532		1.2532	
penalty implicit	110	90	1.249	1	1.2493	14
penalty Crank-Nicolson	110	90	1.2528	1	1.2518	13
[6]	100	90	4.1178		4.1178	
penalty implicit	100	90	4.0996	1	4.1059	14.4
penalty Crank-Nicolson	100	90	4.1112	1	4.1126	15
[6]	85	80	12.4360		12.4360	
penalty implicit	85	80	12.4317	1	12.4317	14
penalty Crank-Nicolson	85	80	12.4358	1	12.4358	14
variable timestep implicit	85	80	12.4343	1.7	12.4397	6.5
variable timestep Crank-Nicolson	85	80	12.4504	1.7	12.4473	6.5
[6]	100	80	1.7849		1.7849	
penalty implicit	100	80	1.786	1.4	1.7845	14
penalty Crank-Nicolson	100	80	1.788	1.5	1.7853	13
variable timestep implicit	100	80	1.8022	1.3	1.8038	6.4
variable timestep Crank-Nicolson	100	80	1.8032	1.3	1.7966	6.4

W tym przypadku widzimy, że zmienny krok czasowy daje gorszą dokładność niż zastosowanie stałego kroku z mniejszą liczba punktów czasoprzestrzeni. W związku z większą wrażliwością na parametry funkcji, doradzam raczej stosowanie stałego kroku czasowego w przypadku opcji knock-in.

down-and-out call

Ceny opcji down-and-out call dla parametrów: S0 = 100, K = 100, r = 0.1, sigma = 0.2, T = 0.5, B1 = 99.9

Źrodło/Metoda	Mx	Mt	Cena	CPU (sec)
[4]			0.164	
penalty implicit	200	200	0.1622	0.5
penalty implicit	500	500	0.1648	9
penalty Crank-Nicolson	200	200	0.1622	0.5
penalty Crank-Nicolson	500	500	0.1648	8.7
variable timestep implicit	200	104	0.1649	0.6
variable timestep implicit	500	104	0.1648	2.5
variable timestep Crank-Nicolson	200	105	0.1652	0.6
variable timestep Crank-Nicolson	500	105	0.1658	2.5

double-knock-out call

Ceny opcji double-knock-out call dla parametrów: S0 = 100, K = 100, r = 0.1, sigma = 0.2, T = 0.5, B1 = 95, B2 = 125

Źródło/Metoda	Mx	Mt	Cena	CPU (sec)
[4]			5.462	
penalty implicit	200	200	5.4623	1
penalty implicit	500	500	5.4685	8.6
penalty Crank-Nicolson	200	200	5.467	1
penalty Crank-Nicolson	500	500	5.4703	8.7
variable timestep implicit	200	116	5.467	0.7
variable timestep implicit	500	116	5.4681	2.7
variable timestep Crank-Nicolson	200	117	5.4776	0.7
variable timestep Crank-Nicolson	500	117	5.4782	2.7

Bibliografia

[1] P. A. Forsyth, K. R. Vetzal, *Quadratic convergence for valuing American options using a penalty method*, SIAM J. Sci. Comput., 23 (2002), pp. 2095–2122

[2] R. Zvan, K. R. Vetzal, P. A. Forsyth, *PDE methods for pricing barrier options*, (1997)

[3] C. C. Christara, D. M. Dang, *Adaptive and higher-order methods for valuing American options*, Journal of Computational Finance, (2010)

[4] R. Zvan, K. R. Vetzal, P. A. Forsyth, *PDE methods for pricing barrier options*, Journal of Economic Dynamics and Control, (2000)

[5] A. Palczewski, P. Kowalczyk, Wykłady *Finanse Obliczeniowe* na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego

[6] F. Aitsahlia, L. Imhof, T. L. Lai, *Pricing and hedging of American knock-in options*, (2004)