Dokumentacja

Kalibracja powierzchni lokalnej zmienności (local volatility) przy użyciu powierzchni natychmiastowej zmienności implikowanej

Łukasz Adamski i Mateusz Kapturski

Wstęp

Kalibracja powierzchni lokalnej zmienności (ang. *local volatility*) należy do największych wyzwań matematyki finansowej. Pomimo, iż rezultat teoretyczny, uzyskany przez Dupire znany jest kilkunastu lat, efektywna implementacja tych idei nastręcza wielu problemów praktykom, głównie ze względu na zbyt małą ilość potrzebnych informacji rynkowych. Innym powodem jest wysoka niestabilność numeryczna tego problemu (ang. *ill-posedness*). Ponadto, wzór teoretyczny, uzyskany przez Dupire'a posiada niewielką wartość praktyczną ze względu na fakt, iż numeryczne rozwiązywanie zaproponowanych równań prowadzi do bardzo dużych oscylacji. Próba poradzenia sobie z powyższymi trudnościami stanowi główną treść podejścia zastosowanego przez nas a opisanego przez Gabriela Turinici'ego, którego podejście wykorzystujemy. Implementacja algorytmu została przeprowadzona w środowisku obliczeniowym Matlab, które z uwagi na lepiej rozbudowany interfejs graficzny nadawało się do testowania poszczególnych funkcji składowych.

Model Rynku

Podstawowym założeniem, jakie przyjmujemy jest założenie o procesie ceny aktywa bazowego, wyrażającego się równaniem:

$$\frac{dS_t}{S_t} = (r(t) - q(t))dt + \sigma dW_t \tag{1}$$

$$S|_{t=0} = S_0 \tag{2}$$

r(t) to zależna od czasu ciągła stopa procentowa wolna od ryzyka, zaś q(t) to stopa ciągłej dywidendy. W naszym przypadku rozważać będziemy parametr $\sigma = \sigma(S, t)$ jako funkcję zależną od czasu oraz ceny aktywa bazowego. Ponadto, używać będziemy oznaczenia $C(S, t; K, T, \sigma)$ jako cena opcji o cenie wykonania K, okresie zapadalności T i zmienności σ, obserwowana w czasie t, przy wartości aktywa bazowego S.

Implied Volatility (zmienność implikowana)

Z rozważań Dupire'a mamy następującą zależność:

$$\sigma(K, T) = \sqrt{2\frac{C_T + rKC_K}{K^2 C_{KK}}} \tag{3}$$

gdzie zmienne z indeksami oznaczają odpowiednie pochodne cząstkowe.

Jak już wspomniano, zazwyczaj dysponujemy zbyt małą ilością danych, aby można było w efektywny sposób uzyskać odpowiednią aproksymację zmienności lokalnej. Dlatego naszym podstawowym źródłem informacji do rozwiązania zadania będzie powierzchnia implikowanej zmienności $\sigma^I(K,T)$ taka, że zachodzi

$$C^{B-S}(S_0,0;K,T,\sigma^I(K,T)) = C^{quoted}(K,T) \tag{4}$$

Zmienność implikowana to taka wartość, która wstawiona do wzoru Blacka-Scholesa wraz z pozostałymi aktualnymi danymi daje cenę równą rynkowej cenie opcji o danej cenie wykonania i terminie zapadalności.

Zależności między implied oraz local volatility

Obecnie możemy już zdefiniować podstawowe obiekty służące opisowi rozwiązywanego przez nas problemu.Pierwszy z nich to przekształcenie z przestrzeni local volatility w przestrzeń cen opcji zadane wzorem:

$$\mathcal{P}(\sigma)(K,T) = C(S_0,0;K,T,\sigma) \tag{5}$$

Drugi zaś to przekształcenie $\mathcal{V}$ z kwadratu lokalnej zmienności v (lokalnej wariancji) w kwadrat implikowanej zmienności $\omega = (\sigma^I(K,T))^2$:

$$\mathcal{V}(v)(K,T) = \omega = (\sigma^I(K,T))^2 \tag{6}$$

Sformułowanie problemu

Jak już wspomniano, kalibracja lokalnej zmienności jest w ogólności czynnością bardzo trudną, z mnóstwem pułapek natury numerycznej, optymalizacyjnej oraz chronicznym niedoborem danych rynkowych włącznie. Konieczne jest zatem odpowiednie podejście do tego zagadnienia, aby zminimalizować wpływ niekorzystnych czynników. Aby poradzić sobie z powyższymi problemami sprowadzimy kalibrację lokalnej zmienności do zagadnienia optymalizacyjnego, co więcej, sama przestrzeń lokalnej zmienności będzie przestrzenią sparametryzowaną, co umożliwi znalezienie rozwiązania. Należy podkreślić jednak, że sam model zmienności jest nieparametryczny. Problem optymalizacyjny dotyczył będzie odwracania przekształcenia $\mathcal{V}$. Mianowicie będziemy dla danego zestawu cen opcji szukać rozwiązania $v_0 = \sigma_0^2$, Pamiętając, że $\mathcal{V}(v)$ to wyjściowa przestrzeń implied volatility, my zaś szukamy local volatility. Odwracamy zatem

przekształcenie:

$$\sigma^2 = v \rightarrow \omega = \mathcal{V}(v) \tag{7}$$

Pomimo, iż przekształcenia $\mathcal{P}$ oraz $\mathcal{V}$ mogłyby posłużyć do znalezienia v, decydujemy się na to drugie przekształcenie z racji istnienia pewnych udowodnionych własności tego przekształcenia, w szczególności pewne własności asymptotyczne, a także fakt, iż przekształcenie

$$v \in L^2(0,T) \rightarrow ||\mathcal{V}(v) - \mathcal{V}(v_0)||^2 \tag{8}$$

jest ściśle wypukłe, co jest dla nas informacją istotną, gdyż implikuje fakt, że równanie:

$$\mathcal{V}(v) = \mathcal{V}(v_0) \tag{9}$$

posiada jednoznaczne rozwiązanie. Kłopotliwe jednak pozostaje złe postawienie problemu optymalizacyjnego. Małe zaburzenie $\mathcal{V}(v_0)$ powoduje, że rozwiązanie problemu odwracania tego przekształcenia zmienia się znacząco. Autor sugeruje pozbycie się tego problemu poprzez dodanie do funkcji optymalizacyjnej tzw. regularyzacji Tychonoffa, co w naszym przypadku oznacza składnik $||v'||^2_{L^2}$. Autor dowodzi, iż dodanie tego składnika poprawia stabilność rozwiązania na tyle, że w skończonej liczbie kroków można je odnaleźć. Jednak jak wynika z naszych obserwacji, numeryczna implementacja składnika regularyzacji sama w sobie stanowi spory problem, zaś parametr wagi, z jaką dodajemy regularyzację do funkcji celu ma duży wpływ na wynikową powierzchnię. Istnieje kilka podejść do problemu regularyzacji [patrz: Crepey]. Regularyzacja, na którą się zdecydowaliśmy jest postaci

$$\int_0^\infty \int_0^\infty (\partial_t f_{ij}(S,t))^2 + (\partial_S f_{ij}(S,t))^2 dt dS \tag{10}$$

Powyższe niewłaściwe całki zostały obcięte do S_{max}, czyli wartości aktywa bazowego, które jest użyte jako końcowa wartość przy rozwiązywaniu poniżej opisanych równań różniczkowych.

Numeryczna implementacja problemu

Rozwiązanie naszego problemu, zaczynamy od wyjściowej powierzchni zmienności implikowanej $\mathcal{V}(v_0)(K_l, T_l)$, otrzymanej z rynkowych cen opcji $C(S_0, 0; K_l, T_l, \sigma_0)$. Poszukuje takiej powierzchni lokalnej zmienności v, która będzie minimalizowa-

ła następujący funkcjonał:

$$J^{\varepsilon}(v) = \varepsilon||v'||^2_{L^2} + \sum_{l=1}^{L}(\mathcal{V}(v)(K_l, T_l) - \mathcal{V}(v_0)(K_l, T_l))^2 \tag{11}$$

Istotną zaletą tego funkcjonału jest jego różniczkowalność w rozwiązaniu oraz w punkcie będącym rzutem zmienności implikowanej na przestrzeń zmienności lokalnej. Zadanie to będziemy rozwiązywali stopniowo, stosując sekwencyjne programowanie kwadratowe i znajdując pośrednie powierzchnie v^k. Mając dane v^k będziemy znajdować k-tą poprawkę p^k w taki sposób, że $v^{k+1} = v^k + p^k$. Mając takie sformułowanie problemu możemy rozwinąć funkcjonał $J^{\varepsilon}(v)$ w szereg Taylora wokół punktu v^k otrzymując problem optymalizacyjny dla k-tej iteracji:

$$\min_p\{J^{\varepsilon}(v^k) + D_v J^{\varepsilon}(v^k)(p) + \frac{1}{2}D_{vv}J^{\varepsilon}(v^k)(p,p)|v^k + p\} \tag{12}$$

Okazuje się, że składnik zawierający drugą różniczkę zbiega bardzo szybko do 0, dzięki czemu wystarczy obliczyć $D_v J^{\varepsilon}(v^k)(p)$. Nie jest to zadanie łatwe, a Turinici używa do tego celu techniki stanów sprzężonych (ang. *adjoint state technique*). Jeśli ceny kwotowanej na rynku l-tej opcji oznaczymy przez $C^l(S, t; K_l, T_l, \sigma) = C^l(S, t)$, to spełnia ona znane równanie Blacka-Scholesa:

$$\frac{\partial C^l}{\partial t} + \frac{\partial C^l}{\partial S}(r - q)S + \frac{1}{2}\frac{\partial^2 C^l}{\partial S^2}\sigma S^2 = rC^l \tag{13}$$

$$C^l(S, T_l) = (S - K_l)^+ \tag{14}$$

Z rozważań teoretycznych otrzymujemy wzór zawierający tzw. zmienną sprzężoną χ:

$$D_v C^l(S_0, 0) = \frac{S^2}{2}\frac{\partial C^l}{\partial SS}\chi \tag{15}$$

przy czym χ spełnia równanie stanu sprzężonego:

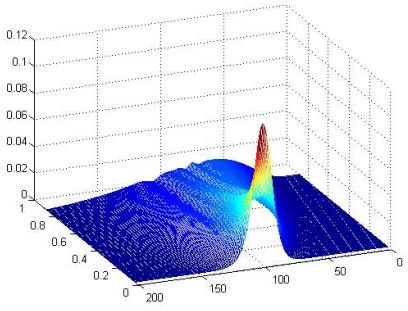
$$\frac{\partial \chi}{\partial t} + \frac{\partial((r - q)S\chi)}{\partial S} + \frac{\partial\left(\frac{vS^2}{2}\chi\right)}{\partial S^2} = r\chi \tag{16}$$

$$\chi(S, t = 0) = \delta_{S=S_0} \tag{17}$$

Równanie to udało nam się rozwiązać, a wynik prezentuje Rysunek 1.

Należy zaznaczyć, iż v w powyższym wzorze jest zmienną, a nie stałą. Jest to przybliżenie zmienności lokalnej w k-tej iteracji. Rozwiązanie jest wspólne dla wszystkich opcji dzięki temu, iż cała płaszczyzna zmienności lokalnej jest wykorzystywana w celu jego znalezienia. Warto zwrócić uwagę na charakterystyczne nierówności, które pojawiają się na tym rozwiązaniu. Są one konsekwencją wysokiej nieregularności funkcji lokalnej zmienności. Im bliżej chwili obecnej i ceny spot, tym wpływ tych wahań jest mniejszy, gdyż zbliżamy się do punktu osobliwego, jakim jest cena spot w chwili 0. Warto dodać, że równanie stanu sprzężonego jest takie samo dla każdej opcji, w związku czym równanie to rozwiązuje się je tylko raz w danej iteracji algorytmu.

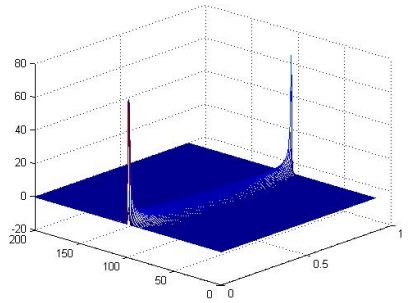
Rysunek 1. Rozwiązanie równania sprzężonego do równania Blacka.



Źródło: Opracowanie własne.

Rysunek 2. przedstawia płaszczyznę $D_v C^l$, która wizualnie jest identyczna z płaszczyzną przedstawioną w pracy Turinici'ego.

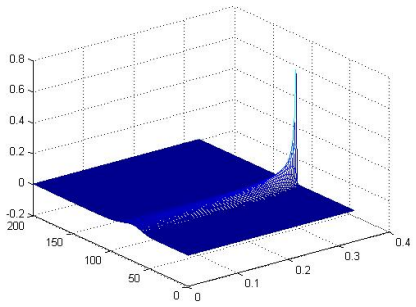
Rysunek 2. Płaszczyzna DvC.



Źródło: Opracowanie własne.

Pierwsza z osobliwości widoczna na Rysunku 2. jest konsekwencją postaci rozwiązania równania sprzężonego (a w zasadzie warunku brzegowego dla tego problemu numerycznego). Drugi to konsekwencja postaci gammy funkcji ceny opcji. Gammę prezentuje Rysunek 3.

Rysunek 3. Gamma opcji Call z momentem zapadalności $t = 0.4$.



Źródło: Opracowanie własne.

Aby otrzymać potrzebną różniczkę $D_v\mathcal{V}(v)$ wystarczy zauważyć, że można zastosować zależność wprost z formuły Blacka-Scholesa:

$$D_v\mathcal{V}(v)(K_l, T_l) = D_v(\sigma^I)^2 = 2\sigma^I \frac{\partial \sigma^I}{\partial C^l} D_v C^l \tag{18}$$

przy czym $\frac{\partial C^l}{\partial \sigma^I}$ to po prostu Vega ze wzorów Blacka Scholesa. Otrzymujemy zatem:

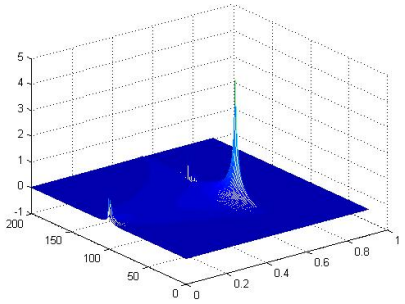
$$D_v\mathcal{V}(v)(K_l, T_l) = 2\sqrt{\mathcal{V}(v)(K_l, T_l)} \frac{1}{Vega_{BS}} D_v C^l \tag{19}$$

Płaszczyznę różniczki $D_v\mathcal{V}(v)(K_l, T_l)$ przedstawia Rysunek 4.

Należy podkreślić, że obliczanie powyższej płaszczyzny jest zazwyczaj związane z dużymi trudnościami numerycznymi i nie zawsze płaszczyzna ta jest tak regularna, jak przedstawiona powyżej. Jest związanie z operacją dzielenia po współrzędnych przez Vegę, czyli pochodną rozwiązania równania Blacka po zmienności implikowanej. Parametr ten ma jednak własność wygasania w sytuacji, gdy cena nie jest bliska cenie wykonania (dla opcji bardzo w cenie lub poza nią). Dzieje się to tym szybciej, im bliższy jest moment wykonania tej opcji. Uwagi te ilustruje Rysunek 5.

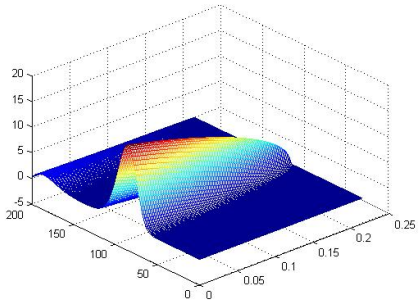
Parametryzacja przestrzeni rozwiązań

Rysunek 4. Płaszczyzna $D_v\mathcal{V}$.



Źródło: Opracowanie własne.

Rysunek 5. Vega opcji Call z momentem zapadalności T=0.25.



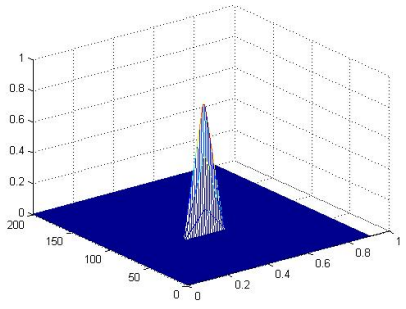
Źródło: Opracowanie własne.

Jak wspomniano powyżej, w celu przeprowadzenie optymalizacji operujemy na sparametryzowanej przestrzeni lokalnej zmienności. Przestrzeń ta rozpięta jest przez bazę, złożoną z ostrosłupów takich jak na Rysunku 6.

Należy zaznaczyć, iż parametryzacja płaszczyzny rozwiązań powinna być niezależna od płaszczyzny zmienności implikowanej. Są to dwie różne siatki, które niekoniecznie muszą się pokrywać w każdym punkcie. Zbyt rzadka siatka stożków powoduje, że ich suma (czyli płaszczyzna zmienności lokalnej w k-tej iteracji) jest dodatkowo nieregularna ze względu na charakter tej bryły.

Ponieważ parametryzujemy przestrzeń funkcji zależnych od dwóch zmiennych, wprowadzając odpowiednią dyskretyzację po obu z nich, pojedynczy ostrosłup bazowy oznaczamy $f_{i,j}(S,t)$, zaś całą przestrzeń lokalnej zmienności zapi-

Rysunek 6. Funkcja bazowa (stożkowa).



Źródło: Opracowanie własne.

sujemy tym samym jako:

$$v(S,t) = \sum_{i,j} \alpha_{i,j} f_{i,j}(S,t) \tag{20}$$

Warto w tym miejscu podkreślić, że przestrzeń stożków jest zupełnie niezależna od siatki danych rynkowych.

Przybliżona formuła wartości przekształcenia $\mathcal{V}$

Nietrudno zauważyć, iż w naszym problemie zmiennymi, które będą optymalizowane są współczynniki parametryzacji płaszczyzny lokalnej zmienności, dokładniej zaś parametry odpowiedniej poprawki. Opisujemy je jako współczynniki $\alpha_{i,j}$. Z tak sparametryzowaną przestrzenią wartość funkcjonału obliczona za pomocą rozwinięcia Taylora do wyrazu pierwszego rzędu ma postać:

$$\mathcal{V}(v+\sum_{i,j} \alpha_{i,j} f_{i,j}(S,t))(K_l,T_l) = \mathcal{V}(v)(K_l,T_l) + \sum_{i,j} D_v\mathcal{V}(v)(K_l,T_l)(f_{i,j})\alpha_{i,j} + o(\alpha) \tag{21}$$

Wyrażenie $D_v\mathcal{V}(v)(K_l,T_l)(f_{i,j})$ to iloczyn skalarny w przestrzeni L^2:

$$D_v\mathcal{V}(v)(K_l,T_l)(f_{i,j}) =< D_v\mathcal{V}(v)(K_l,T_l), f_{i,j} >_{L^2} \tag{22}$$

Iloczyn ten to element macierzy, oznaczony jako $Z^v_{l;i,j}$. Pozostaje jeszcze wyliczenie składników regularyzacji Tychonoffa, czyli niektórych elementów

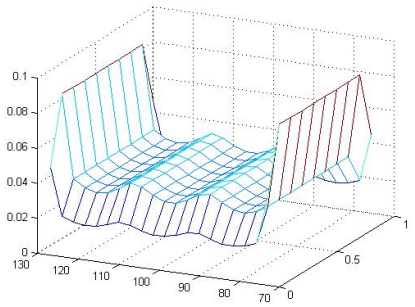
macierzy czterowymiarowych postaci:

$$(Q_s)_{ij;kl} =< \frac{\partial f_{i,j}}{\partial S}(S,t), \frac{\partial f_{k,l}}{\partial S}(S,t) >_{L^2} \tag{23}$$

$$(Q_t)_{ij;kl} =< \frac{\partial f_{i,j}}{\partial t}(S,t), \frac{\partial f_{k,l}}{\partial t}(S,t) >_{L^2} \tag{24}$$

Dzięki odpowiedniemu doborowi bazy do parametryzacji przestrzeni lokalnej zmienności macierze te mają stosunkowo prostą strukturę, w szczególności są macierzami rzadkimi. Na kolejnym rysunku widać efekt tych obliczeń-powierzchnię składnika funkcji celu jakim jest regularyzacja Tychonoffa:

Rysunek 7. Wykres wartości parametru regularyzacji Tichonoffa dla poszczególnych funkcji bazowych.



Źródło: Opracowanie własne.

Nasze zagadnienie będziemy rozwiązywali sekwencyjnie. Teoretycznie nic nie stoi na przeszkodzie, aby optymalnej powierzchni szukać w jednym kroku. Jednak ze względu na dużą niestabilność problemu bezpieczniej jest stopniowo poruszać się w kierunku optymalnej powierzchni. Należy zatem w problemie optymalizacyjnym uwzględnić ograniczenia na zakres możliwej modyfikacji zmiennych optymalizowanych. Jeden z możliwych wyborów, zaproponowanych przez autora to:

$$v(t,S) \in [v_{min}, v_{max}] \tag{25}$$

przy czym:

$$v_{min} = (0.5 \min\{\sigma^I_{K_l,T_l}; l = 1,\ldots,L\})^2 \tag{26}$$

$$v_{max} = (2 \max\{\sigma^I_{K_l,T_l}; l = 1,\ldots,L\})^2 \tag{27}$$

Numeryczna implementacja algorytmu optymalizacyjnego

Naszą procedurę rozpoczynamy od zrzutowanie wyjściowej powierzchni implied volatility, otrzymanej jako dane wejściowe na płaszczyznę rozpiętą przez bazę ostrosłupów $\{f_{ij}\}_{ij}$. Otrzymujemy v_0 i pierwszy zestaw współczynników parametryzacji płaszczyzny local volatility. Następnie w k-tym kroku:

Mając dane $v^k = \sum_{i,j} \beta_{i,j} f_{i,j}(S,t)$ ustalamy wektor:

$$(B^k)_l = \mathcal{V}(v^k)(K_l, T_l) - \mathcal{V}(v_0)(K_l, T_l) \tag{28}$$

następnie obliczamy macierz Z^{v^k} i rozwiązujemy zadanie programowania kwadratowego:

$$\min_{\alpha}\{||Z^{v^k}\alpha + (B^k)_l||^2 + \varepsilon < \alpha + \beta, (Q_s + Q_t)(\alpha + \beta) > |v_{min} - \beta \leq \alpha \leq v_{max} - \beta\} \tag{29}$$

Rezultatem będzie optymalne α^*, które jest zestawem współczynników optymalnej poprawki w k-tym kroku, co wykorzystujemy tworząc powierzchnię:

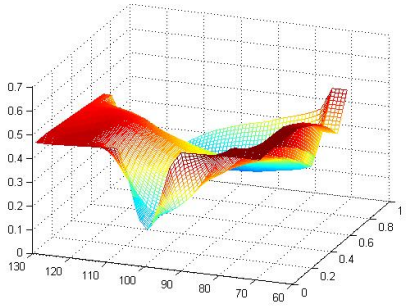
$$v^{k+1} = \sum_{ij} (\alpha^*_{i,j} + \beta_{i,j}) f_{i,j}(S,t) \tag{30}$$

Jeśli nowoutworzona powierzchnia spełnia kryterium błędu (maksymalna zmiana wysokości każdego stożka nie powinna przekraczać 1% zmienności lokalnej), następuje zakończenie algorytmu, jeśli nie, cała procedura powtarzana jest od początku. Algorytm minimalizacji jest przeprowadzany z użyciem funkcji fmincon. Jest to standardowa funkcja Matlaba, która jest rozwijana o kolejne algorytmy minimalizacji z ograniczeniami. Użyliśmy algorytmu zwanego metodą barier lub punktu wewnętrznego (ang. interior point metod).Istotne jest, aby punkt startowy tego algorytmu był równy macierzy zerowej. W przeciwnym razie ryzykujemy, że utkniemy w bardzo dalekim minimum lokalnym, co zmieni diametralnie charakter płaszczyzny zmienności lokalnej. Zazwyczaj algorytm kończy się po kilku do kilkunastu iteracjach. Nie mniej jednak otrzymywana płaszczyzna bardzo istotnie zależy od parametru regularyzacji. Przykładowa płaszczyzny zmienności lokalnej prezentują poniższe rysunki.

Obserwacje i wnioski

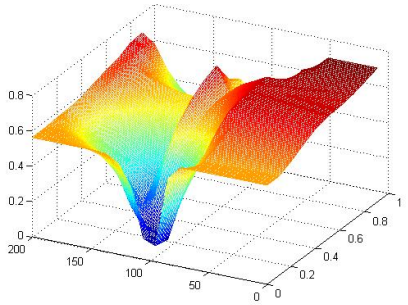
W trakcie implementacji natrafiono na szereg trudności, z których najistotniejszą jest bardzo duża zależność wyniku od parametru regularyzacji Ti-

Rysunek 8. Przykładowa płaszczyzna zmienności lokalnej.



Źródło: Opracowanie własne.

Rysunek 9. Przykładowa płaszczyzna zmienności lokalnej.

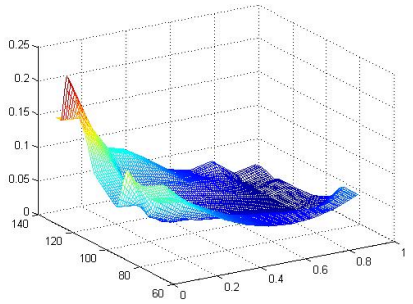


Źródło: Opracowanie własne.

chonoffa. Jeśli będzie on zbyt duży, charakter zadania optymalizacyjnego zmienia się na tyle, iż uzasadnione zdaje się stwierdzenie, iż nie zostało rozwiązane właściwe zadanie optymalizacyjne. Jeśli natomiast jest on zbyt mały, to może się okazać, że algorytm sobie nie poradzi. Utknięcie w minimum lokalnym byłoby dobrą wiadomością. Okazuje się, że często algorytm unosi równolegle płaszczyznę zmienności lokalnej ku górze i nie ma podstaw, by twierdzić, że procedura ta kiedyś się zakończy. Ponadto dla różnych płaszczyzn zmienności różne poziomy tego parametru zdają się dobrze działać. Zazwyczaj jest to około wartości 0.0005, ale nie jest to parametr uniwersalny, właściwy do rozwiązania każdego problemu. Stąd zasadniczo niska przydatność tego algorytmu w praktyce, gdyż każdy przypadek powinien być przebadany i sprawdzony osobno.

Inną kwestią, która niepokoi jest wcześniej wspomniana nieregularność

Rysunek 10. Przykładowa płaszczyzna zmienności lokalnej.



Źródło: Opracowanie własne.

płaszczyzny różniczki $D_v\mathcal{V}$. Podobnie w tym przypadku trudno jest podać rozsądne wyjście z sytuacji. Taka jest po prostu natura numeryczna tych wielkości.

W trakcie implementacji musieliśmy zwrócić szczególną uwagę na kwestię dyskontowania i używania struktury terminowej stóp procentowych. Dla każdej opcji konieczne jest użycie właściwie interpolowanej struktury podobnie jak dla rozwiązania równania sprzężonego do równania Blacka.

Istotną kwestią jest optymalizacja kodu. Dla przykładu warto, aby płaszczyzny stożków zostały obliczone tylko raz, a następnie przechowywane były w tablicach wielowymiarowych, aby zaoszczędzić moc obliczeniową. Wprowadzenie tego ulepszenia w życie doprowadziło do prawie trzykrotnego przyśpieszenia wykonania algorytmu.

W celu przeprowadzenie algorytmu optymalizacji w k-tej iteracji można zasadniczo użyć dowolnego algorytmu. Poprawne sformułowanie funkcji celu jest istotne, by oddzielić kwestie finansowe od zadania optymalizacyjnego. Rozwiązanie zadania optymalizacyjnego dostarcza informacji o zmienności lokalnej w obszarze danych rynkowych. Poza tym obszarem zmienność lokalna jest ekstrapolowana liniowo.

Rozwiązując równania różniczkowe najczęściej korzystaliśmy ze schematu zamkniętego, który jest bezwarunkowo stabilny, ale zmiana jednego parametru pozwoli na użycie na przykład schematu Cranka-Nicolson.

W k-tej iteracji wielkość zmiany płaszczyzny zmienności lokalnej jest ograniczona co do modułu do wielkości (vmin-vmax)/10. Zasadniczo to ograniczenie wymusza na nas procedurę sekwencyjnej optymalizacji. W celu przepro-

wadzenia optymalizacji wykorzystaliśmy również algorytm symulowanego wyżarzania (ang. simulated annealing), ale wyniki pokazują, że otrzymywana płaszczyzna zmienności lokalnej jest wysoce nieregularna. To sugeruje minimum lokalne. Ograniczenie wielkości kroku daje szansę, że algorytm optymalizacji nie pomyli się kilka razy z rzędu. Dywagacje te nie są jednak natury formalnej, a jedynie pokazują intuicję i motywację stojącą za przyjęciem takiego, a nie innego rozwiązania.

Największy koszt obliczeniowy jest ponoszony w celu obliczenia elementów macierzy Z, która musi zostać zaktualizowana w każdej iteracji. Problemu tego nie ma z macierzą Q, która jest taka sama dla każdej iteracji.

Dokumentacja funkcji generujących płaszczyznę zmienności lokalnej

function [LV] = findLocalVol(VM,T,K,S0,r,q,Mx,Mt,nSi,ntj,epsilon,optionType)

%FINDLOCALVOL find local volatility surface on the basis of implied

% variance as first approximation

% INPUT:

% VM - implied volatility matrix of size length(K)xlength(T)

% T,K - vector of strikes and maturities

% S0 - spot price of an underlying asset

% r,q - interest rate and dividend yield term structure (1xlength(T))

% Mx,Mt - density of lattice in space and time direction

% nSi, ntj - number of cones in space and time direction respectively

% epsilon - Tikhonov regularisation parameter

% optionType - indicator 1 for Call option, 0 for Put option

% OUTPUT:

% LV - local volatility mmatrix

Funkcja findLocalVol jest stanowi główną funkcjonalność projektu. Dane wejściowe zostały opisane powyżej. Dla prostoty przyjęliśmy, że zmienna VM jest macierzą, co oznacza, że dla każdego terminu zapadalności mamy dostępne tyle samo opcji o ustalonej liczbie takich samych cen wykonania. Założenie to należy uchylić w warunkach rynkowych. Ponadto, powyższa funkcja potrzebuje struktury terminowej stóp procentowych oraz dywidendy dla wspomnianych powyżej terminów zapadalności. Wartości pośrednie zostaną automatycznie interpolowane liniowo. Nieznaczna zmiana programu pozwoli na wprowadzenie uprzednio obliczonej dowolną metodą struktury stóp procentowych. Pozostałe

dwie pary parametrów podkreślają, że obliczenia są przeprowadzane na trzech "siatkach". Pierwsza jest siatką zmienności implikowanych przez rynek. Druga jest siatką pośrednią dla stożków, które budują płaszczyznę lokalnej zmienności. Jest ona konieczna do sformułowania i rozwiązania problemu optymalizacyjnego. Trzecia siatka jest najbardziej gęsta spośród wszystkich wymienionych i służy do numerycznego rozwiązywania równań różniczkowych Blacka oraz doń sprzężonych.

function [v0 beta] = v0Eval(VM,S,t,K,T,Si,tj)

%V0EVAL calculate projection of v0 on the basis of Volatility Matrix VM

% INPUT:

% VM - volatility matrix of size compatible with K and T

% K,T - strikes and maturities vectors

% S,t - vectors of dense time and space grid respectively

%OUTPUT:

% v0 - projection of implied volatility on Vectf_ij

% beta - initial basic shapes heights

Funkcja v0Eval zwraca rzut powierzchni implikowanej zmienności na przestrzeń stożków (ang. *basic shapes*). Ponadto zwraca także wysokości stożków, które konstytuują tę płaszczyznę. Zmienność implikowana jest pierwszym przybliżeniem zmienności lokalnej. W kolejnych iteracjach płaszczyzna ta jest zmieniana, aby minimalizować opisany wcześniej funkcjonał. Oprócz opisanych wcześniej wielkości przyjmuje ona wektory S oraz t, które odpowiadają gęstej siatce, na której są rozwiązywane równania różniczkowe. Należy zaznaczyć, iż powyższa funkcja zakłada, iż zmienność implikowana poza siatką rynkową jest stała i równa wartości z brzegu tejże.

function [vk_out] = extrapolate(vk)

%EXTRAPOLATE extrapolates k-th aproximation of local volatility outside of

%the scope of cone grid

%INPUT:

% vk - k-th approximation of local volatility(of size length(S)xlength(t))

%OUTPUT:

% vk_out - extrapolated surface

Funkcja extrapolate przeprowadza opisaną powyżej ekstrapolację poza

siatką rynkową. Jako parametr wejściowy przyjmuje k-te przybliżenie zmienności lokalnej na gęstej siatce.

function [Qt_out] = Qt(i,j,k,l,S,t,cones)

%Qt calculate the scalar product in L2 space of first derivatives with

%respect to time of base elements f_ij and f_kl

% INPUT:

% i,j,k,l - cones' coordinates

% S,t - vectors of dense time and space grid respectively

% cones - previously computed basic shape functions values

% OUTPUT:

% Qt_out - value of scalar product of two cones' derivatives

function [Qs_out] = Qs(i,j,k,l,S,t,cones)

%QS calculate the scalar product in L2 space of first derivatives with

%respect to price of base elements f_ij and f_kl

% INPUT:

% i,j,k,l - cones' coordinates

% S,t - vectors of dense time and space grid respectively

% cones - previously computed basic shape functions values

% OUTPUT:

% Qs_out - value of scalar product of two cones' derivatives

Bliźniacze funkcje Qt oraz Qs wyliczają wartość iloczynu skalarnego pochodnych funkcji bazowych (stożkowych) względem czasu i ceny odpowiednio. Dokładniej, obliczany jest iloczyn skalarny funkcji fi,j(S,t) oraz fk,l(S,t). Warto zaznaczyć, iż dla większości przypadków wartość tego iloczynu będzie zerowa, o ile tylko |i-k|>2 oraz |j-l|>2. Stąd rzeczywiście czterowymiarowe macierze QS oraz Qt są rzadkie.

function [out_surface] = fij(i,j,S,t,Si,tj)

%FIJ return surface 1 for S=S(i) and t=t(j) 0 for other prices and times

% from default it uses linear interpolation for consecutive points

% INPUT:

% i,j - price and time coordinates

% S,t - vectors of dense time and space grid respectively

% Si,tj - vectors of cone grids for space and time respectively

% OUTPUT:

% out_surface - basic shape surface of size length(S) x length(t)

Funkcja fij zwraca funkcję stożkową na gęstej siatce, która jest równa 1 dla (Si,tj) oraz zero dla wszystkich pozostałych węzłów. Pomiędzy węzłami dokonuje się interpolacja liniowa, ale zgodnie z sugestią Turinici'ego można użyć dowolnej innej techniki interpolacji (np. interpolacji kubicznej).

function [DvV_out] = DvV(S,timeGridCut,t,S0,r,q,Kk,Tl,v,chi_cut,Mt,Mx,optionType)

%DVV return real matrix containing surface of first derivative of C_k^l

%with respect to local volatility v

%INPUT:

% S,timegridCut,t - space and two time grids

% S0 - spot price

% r,q - interest rate and dividend yield term structure (size 1 x Mt+1)

% up to the maturity i.e. for the interval [0, Tl]

% Kk,Tl - strike and time to maturity of an option

% sigma - implied volatility for the considered option

% chi_cut - solution of the adjoint PDE (cut to the interval [0;Tl])

% Mt, Mx - number of internal grids for time and space

% optionType - indicator 1 for Call option, 0 for Put option

%OUTPUT:

% DvV_out - surface of the derivative of C_k^l w.r. to local volatility

Funkcja $D_v\mathcal{V}$ wylicza płaszczyznę różniczki przekształcenia $\mathcal{V}$ po lokalnej zmienności v. W tym celu konieczne jest uprzednie wyliczenie rozwiązania równania sprzężonego, czyli płaszczyzny ?. W celu obliczenia $D_v\mathcal{V}$ konieczne jest również skorzystanie z Gammy oraz Vegi w model Blacka-Scholesa. Program wykorzystuje metodę różnicową w celu obliczenia odpowiednich przybliżeń wspomnianych wielkości. Można użyć również bezpośrednio wzorów na te parametry greckie, gdyż są one znane, ale jest to podejście wątpliwe i zakłada brak konieczności rozwiązywania L równań różniczkowych dla wszystkich opcji obecnych na rynku. Wyniki otrzymywane tą alternatywną metodą są istotnie różne od przybliżeń różnicowych. Należy zaznaczyć, iż płaszczyzna wynikowa jest płaszczyzną, która zostanie wykorzystana do obliczenia elementów macierzy czterowymiarowej Z. Jest ona bowiem uprzednio interpolowana na prostokąt [0; Smax]x[0;tmax]. Uzasadnia to fakt, iż dynamika ceny opcji opisana równaniem Blacka obejmuje wyłącznie czasy do terminu zapadalności tej opcji.

function [W] = chiPDE(S0,X,tau,v,dv,ddv,r,q,Mt,Mx)

% CHIPDE solve adjoint PDE of the Black PDE on the basis of k-th

% approximation of local volatility

% INPUT:

% S0 - spot price of an underlying asset

% tau, X - dense vectors of lattice on time and space respectively

% v - volatility of an underlying asset (the whole volatility surface)

% dv, ddv - first and second derivatives of volatility with respect to

% space (the whole surface)

% r,q - risk free rate and dividend yield respectively

% Mt, Mx - number of grids in time and space respectively

% OUTPUT:

% W - matrix of the solution for all times and space points

Funkcja chiPDE wyznacza rozwiązanie równania sprzężonego do cząstkowego równania różniczkowego Blacka. Aby uzyskać jak największą pewność otrzymania gładkiego rozwiązania używany jest schemat explicit, który jest bezwarunkowo stabilny. Algorytm używa całej płaszczyzny zmienności lokalnej, którą mamy do dyspozycji w k-tym kroku, a także jej pierwszej i drugiej pochodnej względem ceny aktywa bazowego. Równanie to jest rozwiązywane na tej samej gęstej siatce, na której rozwiązywane jest równanie różniczkowe Blacka. Należy zaznaczyć, iż nie czynimy w tym miejscu żadnego założenia co do zmienności, którą wykorzystujemy. Używana jest cała dostępna w k-tym kroku płaszczyzna zmienności. Podejście takie jest uzasadnione z punktu widzenia tezy, iż rozwiązanie równania sprzężonego jest wspólne dla wszystkich opcji obecnych na rynku.

function [W] = OptionPDE(optionType,S0,sigma,r0,r1,T,K,Mt,Mx)

% OPTIONPDE solve Black PDE in order to get European option values for all times and

% all spot prices using FDM in Black-Scholes model

% INPUT:

% optionType - indicator 1 for Call option, 0 for Put option

% S0 - spot price of an underlying asset

% sigma - volatility of an underlying asset

% r0,r1 - risk free rate and dividend yield term structures respectively

% T - time to maturity of the option

% K - strike

% Mt, Mx - number of grids in time and space respectively

% RETURN:

% W - matrix of all prices up to S_max of all times up to maturity

Funkcja OptionPDE służy do wcześniej wspomnianego rozwiązywania równania różniczkowego Blacka dla zadanej opcji. Na wyjściu otrzymujemy płaszczyznę ceny opcji na prostokącie [0;Smax]x[0;Tl]. Funkcja została napisana w ten sposób, że zmiana jednego wewnętrznego parametru umożliwia użycie innego niż zamkniętego schematu rozwiązywania równań różniczkowych cząstkowych. Sprawdzono jednak, że schemat zamknięty dobrze radzi sobie ze znajdowaniem gładkiego powierzchni ceny opcji europejskiej (Call lub Put) w zależności od czasu do zapadalności oraz ustalonej ceny spot aktywa bazowego.

function [vk_out] = extrapolate(vk)

%EXTRAPOLATE extrapolates k-th aproximation of local volatility outside of

%the scope of cone grid

%INPUT:

% vk - k-th approximation of local volatility

%OUTPUT:

% vk_out - extrapolated surface

O ile w kwestii siatki stożków względem osi czasu nie ma problemu, gdyż składa się ona z równoodległych punktów z odcinka [0; maxl Tl], to nie jest uzasadnione, by stożki pokrywały obszary, dla których wszystkie opcje z rynku są istotnie OTM lub ITM. Dlatego konieczna jest liniowa ekstrapolacja płaszczyzny zmienności lokalnej dla tych obszarów. Funkcja extrapolate dokonuje tej operacji zwracając pełną płaszczyznę lokalnej zmienności. Uzupełnienie tych obszarów jest istotne z punktu widzenia rozwiązywania wcześniej opisanego równania sprzężonego do równania Blacka, które potrzebuje całej płaszczyzny.

Podziękowanie

Dziękujemy Panu Profesorowi Andrzejowi Palczewskiemu za pomoc okazaną w trakcie realizacji powyższego projektu.

Bibliografia

1. Berestycki, H., Busca, J., Florent, I. Asymptotics and calibration of local volatility models. Quantitative Finance 2(1), 61-69.

2. Engl H. W., Hanke M., Neubauer A.. Regularization of Inverse Problems. Kluwer, Dordrecht, 1996.

3. Gatheral, J. The Volatility Surface: A Practitioner's Guide. Wiley, New York 2006.

4. Turinici G., Calibration of local volatility using the local and implied instantaneous variance, The Journal of Computational Finance (1-18) Volume 13/Number 2, Winter 2009/10.