Dokumentacja

Uśmiech zmienności implikowanej

na rynku walutowym.

Joanna Pieniak + AP + Piotr Wiązecki

Spis treści

1 Opis problemu

Celem pracy jest napisanie funkcji, która zwraca wartość zmienności implikowanej dla danych startowych: ceny wykonania, daty początkowej i daty zapadalności, a tym samym wygenerowanie funkcji zmienności od ceny wykonania (zwanej dalej uśmiechem zmienności) i funkcji krzywej zmienności w zależności od czasu pozostałego do zapadalności (powierzchni zmienności). W tym celu zaimplementowałam metody interpolacji zmienności jako funkcji czasu i ceny.

2 Opis teoretyczny

2.1 FX Spot Rate S_t

Bieżąca wartość instrumentu opcyjnego S_t =FOR/DOM oznacza ilosć jednostek waluty krajowej potrzebnej do zakupu jednej jednostki waluty obcej w chwili t. Zatem jeżeli $S_t = 3.8914$ EUR/PLN, to wówczas jeden EUR jest wart 3.8914 PLN, gdzie PLN jest walutą krajową, a EUR zagraniczną.

2.2 FX Outright Forward Contract

Jednym z najbardziej popularnych kontraktów zabezpieczających jest *FX Outright Forward Contract*, zawarty w chwili t, zobowiązuje do wymiany N jednostek waluty obcej na $Nf(t,T)$ jednostek waluty krajowej w chwili T po ustalonej na początku cenie wykonania $f(t,T)$, gdzie

$$f(t,T) = S_t DF_f(t,T)/DF_d(t,T),$$

$DF_f(t,T)$ - czynnik dyskontowy dla waluty zagranicznej,
$DF_d(t,T)$ - czynnik dyskontowy dla waluty krajowej.

Początkowa wartość kontraktu zawartego w chwili t_0 ($t_0 < T$) wynosi zero, a wraz z upływem czasu zmienia się według następującej formuły

$$V_f(t,T) = DF_d(t,T)(f(t,T) - K) = S_t DF_f(t,T) - K DF_d(t,T),$$

gdzie K to cena wykonania ustalona w chwili początkowej.

Wartość tego kontraktu jest wyrażona w jednostkach waluty krajowej.

2.3 FX Vanila Options

Na rynku walutowym mamy także do czynienia z kontraktami opcyjnymi. Istotne dla nas będą tylko opcje waniliowe. Kupujący waniliowy kontrakt opcyjny call FOR (put DOM) zyskuje prawo do wymiany N jednostek waluty obcej po kursie K. Wartość kontraktu jest wyliczana ze standardowego wzoru Blacka – Scholsa, wynosi

$$V(S_t, K, \sigma, \omega) = \omega DF_f(t,T) S_t N(\omega d_+) - DF_d(t,T) K N(\omega d_-)] =$$

$$= \omega DF_f(t,T)[f(t,T)N(\omega d_+) - KN(\omega d_-)],$$

gdzie
$d_+ = \frac{\ln(\frac{f(t,T)}{K})+\frac{1}{2}\sigma^2\tau}{\sigma\sqrt{\tau}}$,
$d_- = d_+ - \sigma\sqrt{\tau}$,
K - cena wykonania dla opcji,
σ - zmienność,
$\omega = 1$ dla opcji call, $\omega = -1$ dla opcji put,
$N(x)$ - dystrybuanta rozkładu normalnego.

Wartość tego kontraktu jest wyrażona w jednostkach waluty krajowej. Równoważnie można zapisać wartość kontraktu w walucie obcej $\frac{v}{S_t}$. Waluta, w której wyrażamy wartość opcji jest nazywana **premium currency**. Przyjmiemy oznaczenie, premium currency = FOR lub DOM, odpowiednio dla waluty obcej lub krajowej.

2.4 Delta - różne konwencje

Procentowa ilość obcej waluty, jaką należy kupić, aby sprzedając kontrakt wciąż utrzymać pozycję zabezpieczającą (tzw. hedging) nazywamy deltą. Na rynku walutowym wyróżniamy następujące rodzaje delt.

Delta spot zabezpiecza transakcje na rynku natychmiastowym, wrażliwość wartości waniliowych opcji na zmiany S_t jest pochodną dana wzorem

$$\Delta_S(K, \sigma, \omega) = \frac{\partial v}{\partial S} = v_S.$$

Korzystając ze wzoru Blacka – Scholsa otrzymujemy następujący wynik

$$\Delta_S(K, \sigma, \omega) = \omega DF_f(t,T)N(\omega d_+).$$

Delta forward zabezpiecza transakcje na rynku FX forward. Wyliczamy ją z następującej formuły.

$$\Delta_f(K, \sigma, \omega) = \frac{\partial v}{\partial v_f} = \frac{\partial v}{\partial S}\frac{\partial S}{\partial v_f} = \frac{\partial v}{\partial S}\left(\frac{\partial v_f}{\partial S}\right)^{-1} = \omega N(\omega d_+).$$

W powyższych przypadkach, wartość kontraktu była wyrażona w walucie krajowej. Jeżeli jest inaczej, czyli premium currency = FOR, to wówczas mamy do czynienia z deltą premium odpowiednio na rynku FX Spot i FX Forward. Należy zmienić ilość waluty zabezpieczającej według poniżej opisanych formuł.

Premium Adjusted Spot Delta wyraża się wzorem

$$\Delta_{S,pa} = \Delta_S - \frac{v}{S}.$$

W tym scenariuszu hedgingowym, należy kupić $N(\Delta_S - \frac{v}{S})$ jednostek waluty obcej (równoważnie sprzedać $N(\Delta_S S_t - v)$ jednostek waluty krajowej), aby zabezpieczyć krótką pozycję waniliową. Odwracając równanie otrzymujemy ilość obcej waluty, którą należy kupić, aby wciąż utrzymać pozycję hedgingową. Dowód tego twierdzenia pominiemy. Korzystając ze wzoru Blacka–Scholsa uzyskujemy jawny wzór na deltę.

$$\Delta_{S,pa}(K, \sigma, \omega) = \omega DF_f(t, T) \frac{K}{f} N(\omega d_-).$$

Premium Adjusted Forward Delta obliczamy korzystając z podobnych analogii jak dla Premium Adjusted Spot Delta, co daje następujący wzór:

$$\Delta_{f,pa}(K, \sigma, \omega) = \omega \frac{K}{f} N(\omega d_-).$$

Zauważmy, że Premium Adjusted Forward Delta jest funkcją monotoniczną względem zmiennej K, co będzie pomocne przy implementacji.

Konwencja doboru delty dla danej pary walutowej

W konstrukcji uśmiechu zmienności, istotny jest wybór delty dla danej pary walutowej. Poniżej przedstawiam algorytm doboru odpowiedniej konwencji. Jeżeli zarówno waluta krajowa, jaki i obca należy do grupy krajów OECD (USD, EUR, JPY, GBP, AUD, NZD, CAD, CHF, NOK, SEK, DKK) i czas pozostały do momentu zapadalności jest nie dłuższy niż 1Y, to należy zastosować wzory dla delty spot, w przeciwnym wypadku wybieramy deltę forward.

Aby określić czy delta jest premium adjusted czy unadjusted korzystam z następującej hierarchii walut:

$$\text{USD} > \text{EUR} > \text{GBP} > \text{AUD} > \text{NZD} > \text{CAD} > \text{CHF} > \text{NOK} > \text{SEK} > \text{DKK} > \text{CZK} > \text{PLN} > \text{TRY} > \text{MXN} > \text{JPY}.$$

Jeżeli indeks waluty krajowej w powyższym porządku jest większy od indeksu waluty obcej to mamy obliczamy deltę typu unadjusted, w przeciwnym wypadku wybieramy deltę premium adjusted. Używając pojęcia Premium Currency można to wyrazićkonwencją: jeśli Premium Currency jest walutą obcą, to stosuje się deltę premium adjusted, jeśli Premium Currency jest walutą krajową, to stosuje się deltę premium unadjusted.

2.5 Definicje At–The–Money

Definicja *at–the–money* nie jest jednoznaczna. Poniżej przedstawiam różne konwencje w zależności od ceny wykonania K. Wyróżniamy:

- ATM–spot, dla $K = S_0$,

- ATM–fwd, dla $K = f$,
- ATM–value–neutral, dla K, takiego że wartość V opcji call = wartość V opcji put,
- ATM–Δ–neutral, dla K, takiego że Δ opcji call = $-\Delta$ opcji put.

Zauważmy, że z parytetu dla opcji call – put bezpośrednio wynika, że ATM–value–neutral jest równa ATM–fwd. Program przyjmuje domyślnie konwencję ATM–Δ–neutral, jeżeli czas pozostały do zapadalności jest krótszy niż rok, w przeciwnym wypadku wybiera ATM–fwd. Użytkownik może ją zmienić, jeżeli uzna to za słuszne.

Poniższa tabela przedstawia relację miedzy rodzajami delty a definicjami ATM.

	ATM Δ-neutral Strike	ATM fwd Strike	ATM Δ-neutral Delta	ATM fwd Delta
Spot Delta	$fe^{0.5\sigma^2\tau}$	f	$0.5\phi DF_f$	$\phi DF_f N(0.5\phi\sigma\sqrt{\tau})$
Forward Delta	$fe^{0.5\sigma^2\tau}$	f	0.5ϕ	$\phi N(0.5\phi\sigma\sqrt{\tau})$
Spot Delta p.a.	$fe^{-0.5\sigma^2\tau}$	f	$0.5\phi DF_f e^{-0.5\sigma^2\tau}$	$\phi DF_f N(-0.5\phi\sigma\sqrt{\tau})$
Forward Delta p.a.	$fe^{-0.5\sigma^2\tau}$	f	$0.5\phi e^{-0.5\sigma^2\tau}$	$\phi N(-0.5\phi\sigma\sqrt{\tau})$

2.6 Delta Strike Conversion

Dla delt typu unadjusted możemy przekonwertować wzory tak, aby obliczyć wartość strike K, dla odpowiadających im zmienności i delt. Są to odpowiednio, dla delty spot i forward:

$$K = fe^{-\omega N^{-1}(\omega DF_f(t,T)\Delta_S)\sigma\sqrt{\tau}+\frac{1}{2}\sigma^2\tau},$$
$$K = fe^{-\omega N^{-1}(\omega\Delta_f)\sigma\sqrt{\tau}+\frac{1}{2}\sigma^2\tau}.$$

Dla premium adjusted spot delta i premium adjusted forward delta, relacja między deltą a strikiem

$$\Delta_{S,pa}(K,\sigma,\omega) = \omega DF_f(t,T)N(\omega\frac{\ln(\frac{f}{K}) - \frac{1}{2}\sigma^2\tau}{\sigma\sqrt{\tau}}), \tag{1}$$

$$\Delta_{f,pa}(K,\sigma,\omega) = \omega\frac{K}{f}N(\omega\frac{\ln(\frac{f}{K}) - \frac{1}{2}\sigma^2\tau}{\sigma\sqrt{\tau}}), \tag{2}$$

nie może być wyrażona wzorem analitycznym. Zatem, aby wskazać K, należy skorzystać z metod numerycznych obliczania miejsc zerowych. Na potrzeby programu wykorzystano wbudowaną w Octave'a funkcję `fzero`, a także jej drobną modyfikację `ffzero`. W przypadku delty put funkcja jest monotoniczna, dlatego znalezienie miejsca zerowego nie przysparza żadnych problemów. W przypadku delty call, funkcja najpierw rośnie do pewnego momentu, a potem maleje, dwukrotnie przecinając zero (interesuje nas drugie miejsce zerowe). Modyfikacja algorytmu funkcji fzero polega ograniczeniu przedziału szukania, gdzie za K_{min} przyjmuję argument, dla którego funkcja osiąga maksimum.

3 Konstrukcja uśmiechu zmienności implikowanej na rynku walutowym

Na rynku walutowym zmienność jest kwotowana w deltach, zależy więc od konwencji wyboru delty i tym samym pary walutowej oraz czasu pozostałego do zapadalności. Wyrózniamy:

- zmienność ATM, dla której wartość K jest tak dobrana, że wartość delty call i wartość delty put mają tę samą wartość, ale z przeciwnym znakiem,
- zmienność xxRR (risk reversal), zmienność implikowana z rynku dla różnicy delty opcji call o wartości $0.xx$ $(xx/100)$ i delty put o wartości delty $-0.xx$.
- zmienność 2-vol-xxBF (butterfly), zmienność implikowana dla różnicy połowy sumy delty opcji call o wartości $0.xx$ i delty put o wartości delty $-0.xx$ oraz zmienności ATM.

Korzystając z powyższych danych rynkowych możemy wyliczyć zmienności implikowane odpowiadające odpowiednio opcji kupna, dla której wartość delty wynosi $0.xx$ (σ_{xxC}) i opcji sprzedaży, dla której wartość delty wynosi $-0.xx$ (σ_{xxP}). Najbardziej popularne jest kwotowanie zmienności dla $xx = 25$, ale częśsto dodatkowo podawane są też wartości dla $xx = 10$.

Po prostych przekształceniach uzyskujemy następujące wzory:

$$\sigma_{xxC} = \sigma_{ATM} + \sigma_{BFxx(2vol)} + 0.5\sigma_{xxRR},$$

$$\sigma_{xxP} = \sigma_{ATM} + \sigma_{BFxx(2vol)} - 0.5\sigma_{xxRR}.$$

Problem polega na tym, że zmienność $\sigma_{BFxx(2vol)}$ nie jest bezpośrednio kwotowana na rynku. Zamiast niej obserwujemy $\sigma_{BFxx(1vol)}$ czyli zmienność nazywaną *quoted strangle*, którą interpretuje się następująco:

- definiujemy zmienność *broker's stragle* jako

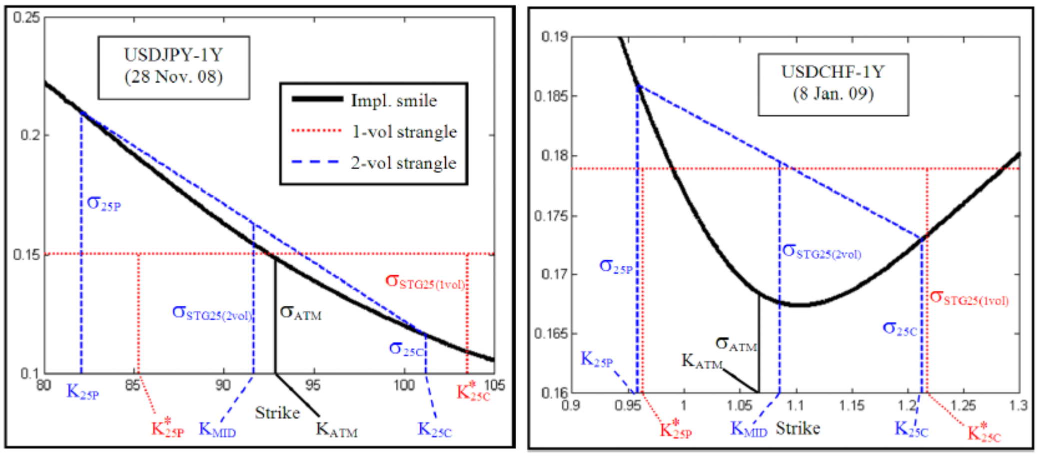
$$\sigma_{STGxx(1vol)} = \sigma_{ATM} + \sigma_{BFxx(1vol)}$$

- stosujemy wzory z poprzedniego podrozdziału (2.6) w celu otrzymania strike'ów K^*_{xxC} oraz K^*_{xxP}, tzn. cen wykonania, dla których delta opcji call jest równa xx, zaś delta opcji put jest równa $(-xx)$ przy użyciu w obu przypadkach zmienności $\sigma_{STGxx(1vol)}$
- zakładając, że mamy poprawnie skalibrowaną do rynku funkcję $K \mapsto \sigma(K)$, przyporządkującą danemu strike'owi odpowiednią wartość zmienności implikowanej, wielkość $\sigma_{BFxx(1vol)}$ jest tak dobrana, aby zachodziła równość

$$\begin{aligned} &Call(K^*_{xxC}, \sigma_{STGxx(1vol)}) + Put(K^*_{xxP}, \sigma_{STGxx(1vol)}) = \\ &= Call(K^*_{xxC}, \sigma(K^*_{xxC})) + Put(K^*_{xxP}, \sigma(K^*_{xxP})), \end{aligned} \tag{3}$$

gdzie $Call$ oraz Put oznaczają odpowiednio cenę opcji call i put o zadanych parametrach, zgodnie ze wzorem podanym w podrozdziale (2.3)

Poniższy rysunek ilustruje różnicę między $\sigma_{BFxx(1vol)}$ a $\sigma_{BFxx(2vol)}$ dla dwóch różnych rynków (rysunek pochodzi z pracy Bossensa [4]):



Pierwszym etapem jest więc odzyskanie wartości $\sigma_{BFxx(2vol)}$ z notowań rynkowych σ_{ATM}, σ_{xxRR}, $\sigma_{BFxx(1vol)}$. Jest to procedura o tyle skomplikowana, że na zależność między $\sigma_{BFxx(1vol)}$ a $\sigma_{BFxx(2vol)}$ wpływa kształt całej krzywej uśmiechu zmienności, więc algorytm prowadzący do uzyskania wielkości $\sigma_{BFxx(2vol)}$ musi zawierać w sobie konstrukcję krzywej zmienności. W programie zastosowano następujący iteracyjny algorytm:

1. wybierz początkową wartość $\sigma_{BFxx(2vol)}$
2. stwórz krzywą zmienności implikowanej przy jej użyciu (w sposób podany niżej)
3. oblicz różnicę

$$\begin{aligned} &Call(K^*_{xxC}, \sigma_{STGxx(1vol)}) + Put(K^*_{xxP}, \sigma_{STGxx(1vol)}) \\ &- [Call(K^*_{xxC}, \sigma(K^*_{xxC})) + Put(K^*_{xxP}, \sigma(K^*_{xxP}))] \end{aligned}$$

4. jeżeli różnica przekracza ustalony poziom telorancji, dostosuj wartość $\sigma_{BFxx(2vol)}$ i wróć do punktu 2.

Po znalezieniu wartości $\sigma_{BFxx(2vol)}$ kolejnym etapem jest wyliczenie strików dla danej σ_{ATM} i uzyskanych σ_{xxC}, σ_{xxP}. W przypadku σ_{ATM} posłużymy się konwencją ATM–Δ–neutral i zachodzących dla niej zależności dla danych σ_{ATM}, S_0, czasu pozostałego do momentu zapadalności $\tau = T - t$ i współczynników dyskontowych

$$DF_f(t,T)\Phi(\frac{ln(S_0/f) + 0.5\sigma^2_{ATM}\tau}{\sigma_{ATM}\sqrt{\tau}}) = DF_f(t,T)\Phi(-\frac{ln(S_0/f) + 0.5\sigma^2_{ATM}\tau}{\sigma_{ATM}\sqrt{\tau}}).$$

Wówczas wartość K_{ATM} wynosi

$$K_{ATM} = S_0 DF_f(t,T)/DF_d(t,T) + exp(0.5\sigma_{ATM}^2\tau).$$

W przypadku σ_{xxC}, σ_{xxP} i przyjętych konwencji obliczania delt, stosujemy wzory z poprzedniego podrozdziału (2.6)

$$\Delta_{(\sigma_{xxC},K_{xxC})} = 0.xx,$$

$$\Delta_{(\sigma_{xxP},K_{xxP})} = -0.xx,$$

dla obliczenia K_{xxC}, K_{xxP}.

W tym celu w programie posłużymy się funkcją `VOL` opisaną w dokumentacji.

4 Interpolacja

4.1 Interpolacja metodą Vanna–Volga

Aby uzyskać krzywą zmienności, dla danych cen wykonania K_{xxP}, K_{ATM}, K_{xxC} oraz odpowiadającym im zmiennościach implikowanych σ_{xxP}, σ_{ATM}, σ_{xxC} przeprowadzamy interpolację między punktową. W tym celu zaimplementujemy metodę Vanna–Volga.

Upraszczając notację przyjmujemy ceny wykonania, jako K_i, dla $i = 1, 2, 3$. Metoda Vanna–Volga opiera się na skonstruowaniu portfeli replikujących składające się odpowiednio x_1, x_2, x_3, jednostek opcji call o cenach wykonania K_1, K_2, K_3 oraz Δ_t ilości aktywa, na które wystawiona jest opcja. Przeprowadzając dość skomplikowany rachunek uzyskujemy następujące wzory na Vegę, Vannę i Volgę (stąd nazwa metody) odpowiednio:

$$\frac{\delta C_{BS}(t;K)}{\delta\sigma} = \sum_{i=1}^{3} x_i(t,K)\frac{\delta C_{BS}(t;K_i)}{\delta\sigma},$$

$$\frac{\delta^2 C_{BS}(t;K)}{\delta\sigma^2} = \sum_{i=1}^{3} x_i(t,K)\frac{\delta^2 C_{BS}(t;K_i)}{\delta^2\sigma},$$

$$\frac{\delta^2 C_{BS}(t;K)}{\delta\sigma\delta S_t} = \sum_{i=1}^{3} x_i(t,K)\frac{\delta^2 C_{BS}(t;K_i)}{\delta\sigma\delta S_t}.$$

Wzory Blacka Scholesa pozwalają uzyskać analityczną postać:

$$\nu(t,K) := \frac{\delta C_{BS}(t;K)}{\delta\sigma}$$

$$\frac{\delta C_{BS}(t;K)}{\delta\sigma} = S_t DF_f(t,T)\sqrt{\tau}\phi(d_+(t,K)),$$

$$\frac{\delta^2 C_{BS}(t;K)}{\delta\sigma^2} = \frac{\nu(t,K)}{\sigma} d_+(t,K)d_-(t,K),$$

$$\frac{\delta^2 C_{BS}(t;K)}{\delta\sigma\delta S_t} = -\frac{\nu(t,K)}{S_t\sigma\sqrt{\tau}} d_-(t,K).$$

Opierając się na powyższych rachunkach i przyjmując za zmienność σ_{ATM} otrzymujemy wagi $x_i(t,K)$:

$$x_1 = \frac{\nu(t,K)ln(K_2/K)ln(K_3/K)}{\nu(t,K_1)ln(K_2/K_1)ln(K_3/K_1)},$$

$$x_2 = \frac{\nu(t,K)ln(K_1/K)ln(K_3/K)}{\nu(t,K_2)ln(K_1/K_2)ln(K_3/K_2)},$$

$$x_3 = \frac{\nu(t,K)ln(K_1/K)ln(K_2/K)}{\nu(t,K_3)ln(K_1/K_3)ln(K_2/K_3)}.$$

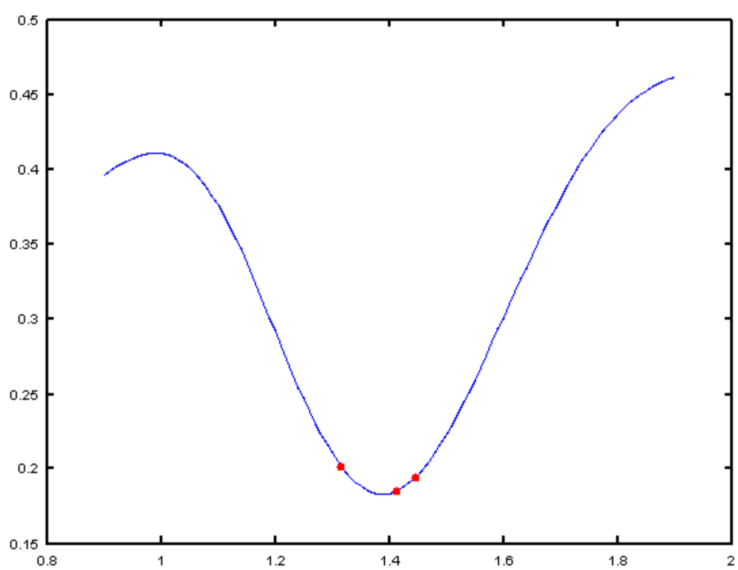
Kwintesencją powyższych rozważań jest alternatywny wzór na cenę opcji dla danego t i K, który implikuje krzywą zmienności

$$C(t,K) = C_{BS}(t,K) + \sum_{i=1}^{3} x_i(t,K)(C_{MKT}(t,K_i) - C_{BS}(t,K_i)), \qquad (4)$$

gdzie $C_{MKT}(t,K_i)$ jest jest ceną opcji call wyliczoną ze wzoru Blacka Scholesa z ceną wykonania K_i i zmiennością σ_i, a $C_{BS}(t,K_\alpha)$ z ceną wykonania K_α i zmiennością σ_{ATM}.

Odwracając powyższą formułę względem σ, można policzyć zależność od każdego K i tym samym krzywą uśmiechu zmienności.

Przykładowa krzywa zmienności implikowanej dla USD/EUR (FOR/DOM) i czasu pozostałego do zapadalności τ = 1Y wygląda następująco.



4.2 Interpolacja metodą jądrową

Metoda jądrowa posłuży do interpolacji między uzyskanymi krzywymi uśmiechów zmienności na przestrzeni czasu, a zatem na płaszczyźnie trzywymiarowej (można ją też wykorzystać do interpolacji na płaszczyźnie (K, σ), ale wyniki empiryczne dowodzą, że nie jest to wskazane).

Dla zbioru obserwacji (x_i, Y_i), dla $i = 1, ..., n$ szacujemy wartość Y, dla danego punktu x korzystamy z następującego wzoru

$$\hat{Y}(x) = \frac{\sum_{i=1}^{n} Y_i F(\frac{x_i - x}{h})}{\sum_{i=1}^{n} F(\frac{x_i - x}{h})}, \tag{5}$$

gdzie $F(\cdot)$ jest tak zwaną funkcją jądrową o współczynniku h. W programie korzystamy z funkcji gausowskiej, danej wzorem:

$$F(u) = exp(-u^2/(2\alpha^2)),$$

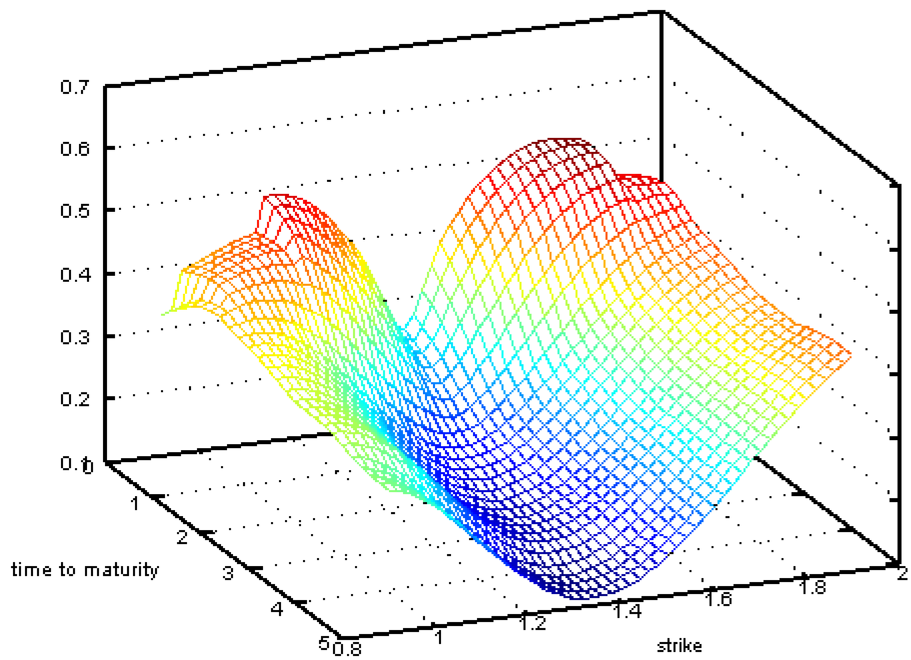
dla parametrów $\alpha = 8$ i $h = 0.1$

4.3 Interpolacja po czasie

W praktyce, aby uzyskać wartość zmienności implikowanej dla danego okresu zapadalności, mając dane krzywe uśmiechu zmienności dla kolejnych momentów czasu, najczęściej stosuje się interpolacje liniową. Jak pokazał Peter Jäckel w [5] taka metoda może prowadzić do problemów, gdy chcemy przeprowadzić symulację Monte Carlo, zgodną z wyceną opcji. Przeprowadzając taką symulację aby przejść od czasu t_1 do t_2 musimy wygenerować zmienną o wariancji $\sigma(t_2)^2 t_2 - \sigma(t_1)^2 t_1$. Metoda liniowej interpolacji funkcji $\sigma(t)$ nie gwarantuje, że powyższa wielkość będzie dodatnia. Problem znika, gdy zamiast bezpośrednio interpolować zmienność, będziemy interpolować funkcję $t \mapsto \sigma(t)^2 t$ za pomocą dowolnej metody interpolacji zachowującej monotoniczność - na przykład interpolacji liniowej. W programie została zaimplementowana zarówno powyższa metoda, jak i zwykła metoda interpolacji liniowej.

5 Wyniki

Zwieńczeniem pracy jest wygenerowanie powierzchni zmienności implikowanej dla cen wykonania i czasu. Poniższy rysunek przedstawia przykładową powierzchnię dla pary walutowej USD/EUR.



6 Dokumentacja

Całość pragramu obejmuje następujące zadania:

1. wczytanie i przetworzenie danych rynkowych,
2. wygenerowanie tablicy zmienności implikowanej
3. stworzenie funkcji, która podaje wartość zmienności implikowanej dla konkretnych danych rynkowych (strike, initial date, expiry date) na podstawie stworzonej tablicy.

6.1 Inicjalizacja

Aby program mógł wykonać swoje zadanie potrzebuje on zewnętrznych informacji definiujących rynek krajowy i zagraniczny, czynników dyskontowych (stóp procentowych) na tych rynkach oraz kursu walutowego spot. Infromacje te wraz z definicjami odpowiednich zmiennych globalnych znajdują się w przykładowym pliku `pre_init_vol.m`. Niezależnie w katalogu, w którym wykonywany jest program musza znajdować się pliki z czynnikami dyskontowymi `DSD_Bid.m`, `DSD_Ask.m`, `DSF_Bid.m`, `DSF_Ask.m` (w skrypcie `pre_init_vol.m` wykorzystano tylko pliki DSD_Bid.m i DSF_Bid.m, ale możliwe jest inne korzystanie z czynników dyskontowych, które będzie wymagało innych z wymienionych plików).

Przed wykonaniem skryptu pre_init_vol.m należy wczytać do pamięci program funkcje_kalendarzowe2, tj. plik funkcje_kalendarzowe2.m oraz program discount_functions, tj. plik discount_functions.m.

6.2 Program główny

W opisie programu będziemy przyjmować dla uproszczenia, że $xx = 25$.

Dane do programu można wczytać przy pomocy przykładowego skryptu VOL_data_in_ver.m, który definiuje potrzebne zmienne globalne oraz wczytuje dane z pliku volatility_data_in.m.

Następnie należy wczytac główny program implied_volatility i wykonać jego pierwszą funkcję market_vol_gen.

Funkcja market_vol_gen generuje plik market_vol.m, korzystając z wprowadzonych danych. Umożliwia to użytkownikowi ingerowanie w dane i wprowadzenie zmian poprzez zmianę zawartości pliku market_vol.m. Funkcja ta wywołuje także funkcje PREM_CURR i delta_convention ustalając Premium Currency oraz konwencje obliczania delty.

```
% output: dane przygotowane do późniejszego uzycia

market_vol_gen()
```

Funkcja create_vol_table wczytuje dane z pliku market_vol.m i przygotowuje tablicę zmienności implikowanych volatility_grid dla każdego z czasów zapadalności podanego w danych oraz dla ustalonej siatki wartości strike (gęstość siatki strike ustala zmienna m_points). Siatka wartości strike jest zapamiętywana w tablicy K_grid a wektor czasów zapadalności w T_grid. Wszystkie te tablice są zapisywane w pliku vol_tables.m.

Wartości zmienności implikowanej na siatce są obliczane z wykorzystaniem jednej z dostępnych w programie metod interpolacji ustalanej przez zmienną smile_interp. W obecnej wersji program dopuszcza dwie metody interpolacji: Vanna–Volga oraz metodę jądrową.

```
% output: dane przygotowane do późniejszego uzycia

create_vol_table()
```

Ukoronowaniem pracy jest funkcja ImpVol, która zwraca watrość zmienności implikowanej dla danych początkowych.

```
% input:
%   strike - cena wykonania
%   start_date - data początkowa
%   expire_date - data zapadalności
%
```

```
% output: wartość zmienności implikowanej

ImpVol(strike, start_date,expire_date)

% przykładowe wywołanie
ImpVol(1.4, "24-aug-2009", "24-sep-2010")
ans = 0.2152
```

W następnych podrozdziałach znajduje się krótki opis wszystkich funkcji pomocniczych, z których korzystano przy implementacji funkcji `ImpVol`. Aby dobrze zrozumieć działanie programu należy dokładnie prześledzić opis teoretyczny.

6.3 Wybór konwencji

Funkcja `PREM_CURR` oblicza walutę premium dla danej pary walutowej.

```
% input:
%   CURR_FOR - waluta obca,
%   CURR_DOM - waluta krajowa
% output: waluta premium

PREM_CURR(CURR_FOR, CURR_DOM)
```

Funkcja `delta_convention` oblicza konwencję wyboru delty dla danej pary walutowej, przekazując jeden z parametrów: SPOT_UN, SPOT_PA, FORWARD_UN, FORWARD_PA.

```
% input:
%   CURR_FOR - waluta obca,
%   CURR_DOM - waluta krajowa,
%   tau - czas do zapadalności,
% output: konwencja delty

delta_convention(CURR_FOR, CURR_DOM, tau)
```

Funkcje `spot_un`, `spot_pa`, `fwd_un`, `fwd_pa` obliczają ceny wykonania K_{ATM}, K_{25C}, K_{25P} odpowiadające σ_{ATM}, σ_{25C}, σ_{25P} odpowiednio, w zależności od wyboru konwencji. Dodatkowo funkcje te obliczają K_{LC} i K_{LP}, minimalną i maksymalna wartość K dla jakiej może być wykonana ekstrapolacja. W przypadku funkcji `spot_un` i `fwd_un` korzystamy z gotowych wzorów analitycznych opisanych rozdziale 2.6. Dla funkcji `spot_pa` i `fwd_pa` niezbędne jest obliczenie miejsca zerowego formuł (1) i (2). W tym celu korzystamy z funkcji pomocniczych `spot_pa_call`, `spot_pa_put`,`fwd_pa_call`, `fwd_pa_put`, a także z wbudowanej w Octave funkcji obliczania miejsc zerowych `fzero`.

```
% input:
%   sigma_deltaC, sigma_deltaP, sigma_atm - zmienności dla delt xxC, xxP, ATM
```

```
%   S_0 - cena początkowa,
%   f = FX_rate * DF_d/DF_f,
%   tau - czas do zapadalności,
%   DF_f, DF_d - czynniki dyskontowe, zagraniczny i krajowy odpowiednio
%   ATM_conv - konwencje wyboru atm,
% output:
%   wektor [K_atm, K_deltaC, K_deltaP, K_LC, K_LP]

spot_un(sigma_deltaC, sigma_deltaP, sigma_atm, S_0, f, tau, DF_f, DF_d, ATM_conv)
spot_pa(sigma_deltaC, sigma_deltaP, sigma_atm, S_0, f, tau, DF_f, DF_d, ATM_conv)
fwd_un(sigma_deltaC, sigma_deltaP, sigma_atm, S_0, f, tau, DF_f, DF_d, ATM_conv)
fwd_pa(sigma_deltaC, sigma_deltaP, sigma_atm, S_0, f, tau, DF_f, DF_d, ATM_conv)

% funkcje pomocnicze
spot_pa_call(K, f, sigma_deltaC, tau, DF_f, DF_d, delta_val)
spot_pa_put(K, f, sigma_deltaP, tau, DF_f, DF_d, delta_val)
fwd_pa_call(K, f, sigma_deltaC, tau, DF_f, DF_d, delta_val)
fwd_pa_put(K, f, sigma_deltaP, tau, DF_f, DF_d, delta_val)
```

Funkcja `VOL` zwraca wartość cen wykonania i zmienności dla delt xxC, ATM, xxP.

```
% input:
%  expiry - data zapadalności,
%   sigma_atm, sigma_rr, sigma_str - dane odczytane z rynku,
%   ATM_conv = {delta_neutral, atm_fwd, atm_spot, atm_value_neutral},
%   conv = {SPOT_UN, SPOT_PA, FORWARD_UN, FORWARD_PA},
%
% output:
%   strikes - wektor cen wykonania [K_xxP, K_atm, K_xxC],
%   sigmas - wektor zmienności [sigma_xxP, sigma_atm, sigma_xxC],
%   DFs - czynniki dyskontowe,
%   tau - czas pozostały do zapadalności,
%   limits - lewa i prawa granica strike dla której wyniki ekstrapolacji jeszcze
%            mają sens (te granice sa ustalane przez podanie granicznej wartości delta
%            w zmiennej limit_delta

function strikes, sigmas, DFs, tau, limits = VOL(expiry, sigma_atm,
        sigma_rr, sigma_str, ATM_conv, conv)
```

6.4 Uzyskanie wartości zmienności 2-vol-BF (butterfly)

Algorytm obliczania wartości $\sigma_{BF25(2vol)}$ realizuje funkcja `BF2vol`, znajdująca miejsce zerowe funkcji `strangle_value_difference`. Korzysta z wbudowanej w Octave funkcji `fzero`.

```
% input:
%   expiry -  data zapadalności
%   sigma_atm, sigma_rr, sigma_BF1vol - odczytane z rynku
%   ATM_conv, conv - przyjęte konwencje
%
% output: wartość sigma_BF2vol

BF2vol(expiry, sigma_atm, sigma_rr, sigma_BF1vol, ATM_conv, conv)
```

Funkcja `strangle_value_difference` jest funkcją parametru `sigma` oraz danych odczytanych z rynku. Funkcja zwraca różnicę między lewą i prawą stroną wyrażenia (3) powstałą z podstawienia danej na wejściu wartości `sigma` jako $\sigma_{BF25(2vol)}$ do wyznaczenia krzywej zmienności. Obliczone zostają kolejno wielkości `sigma_STG`$= \sigma_{STG25(1vol)}$, `K_deltaC_star`$= \sigma(K^*_{xxC})$, `K_deltaP_star`$= \sigma(K^*_{xxP})$ (dwie ostatnie wielkości funkcja oblicza identyczną metodą jak funkcje `spot_un`, `fwd_un`, `spot_pa`, `fwd_pa`). W celu wyznaczenia krzywej zmienności z użyciem parametru `sigma` wywoływana jest funkcja
`VOL(expiry, sigma_atm, sigma_rr, sigma, ATM_conv, conv)`,
której output jest następnie wykorzystywany przy obliczaniu $\sigma(K^*_{xxC})$, $\sigma(K^*_{xxP})$ (co dokonuje się za pomocą funkcji `strike_interpolate`, opis w dalszej części dokumentacji). Funkcja korzysta ponadto z funkcji `C` do obliczenia ceny opcji kupna o zadanych parametrach (opis w dalszej części dokumentacji).

```
% input:
%   sigma - wartość zmienności podstawionej zamiast sigma_BFvol2
%   expiry - data zapadalności
%   sigma_atm, sigma_rr, sigma_BF1vol - odczytane z rynku
%   ATM_conv, conv - przyjęte konwencje
% output:
%   diff - różnica w wartości broker’s strangle

strangle_value_difference(sigma, expiry, sigma_atm, sigma_rr, sigma_BF1vol, ATM_conv, conv)
```

6.5 Interpolacja metodą Vanna–Volga

Funkcja `vanna_volga_interpolate` interpoluje krzywą metodą Vanna Volga dla danych trzech punktów (σ_{25P}, K_{25P}), (σ_{ATM}, K_{ATM}), (σ_{25C}, K_{25C}). Oblicza miejsce zerowe wyrażenia opisanego w rozdziale 4.1, korzystając z funkcji `ffzero`, a także funkcji pomocniczych `vanna_volga`, `vega` i `C`.

```
% input: strikes, sigmas, DFs, tau, strike
%   strikes - wektor cen wykonania [K_25P, K_atm, K_25C],
%   sigmas - wektor zmienności [sigma_25P, sigma_atm, sigma_25C],
%   DFs - czynniki dyskontowe,
%   tau - czas pozostały do zapadalności,
%   strike - cena wykonania, dla której zmienność interpolujemy
% output: interpolowana wartość zmienności dla ustalonego strika
```

```
vanna_volga_interpolate(strikes, sigmas, DFs, tau, strike)
```

Funkcja `vanna_volga` oblicza wartość prawej strony wyrażenia (4). Korzysta z funkcji `vega` i `C`.

```
% input:
%   strikes - wektor cen wykonania [K_25P, K_atm, K_25C],
%   sigmas - wektor zmienności [sigma_25P, sigma_atm, sigma_25C],
%   DFs - czynniki dyskontowe,
%   tau - czas pozostały do zapadalności,
%   K - cena wykonania, dlaktórej interpolujemy wartość sigma
% output: wartość opcji kupna

vanna_volga(strikes, sigmas, DFs, tau, K)
```

Funkcja `C` oblicza lewą stronę wyrażenia (4), czyli wartość opcji kupna.

```
% input:
%   S_0 - cena początkowa,
%   f = FX_rate * DF_d/DF_f,
%   tau - czas do zapadalności,
%   sigma - zmienność,
%   K - cena wykonania,
%   DF_f, DF_d - czynniki dyskontowe, zagraniczny i krajowy odpowiednio
% output: wartość opcji kupna

C(S_0, K, sigma, f, tau, DF_f, DF_d)
```

Natomiast funkcja pomocnicza `vega` oblicza wartości vegi, vanny i volgi, korzystając ze wzorów analitycznych w rozdziale 4.1, parametry wejścia jak wyżej.

```
% output: wartość vega

vega(S_0, f, tau, sigma, K, DF_f, DF_d)
```

6.6 Interpolacja metodą jądrową

Funkcja `kernel_interpolate` oblicza wartość wyrażenia (5), czyli szacuje wartość funkcji dla pewnego argumentu, przy danym zbiorze informacji.

```
% input:
%   X - zbiór informacji - argumenty
%   Y - zbiór informacji - wartości
%   x - argument, dla którego wartość interpolujemy
% output: wartość funkcji w punkcie x
```

```
kernel_interpolate(X, Y, x, h = 0.01, kernel=@gaussian_kernel)
```

```
% input:
%   u - zmienna
%   a - parametry funkcji (domyślnie 8)
% output: wartość funkcji gaussowskiej

gaussian_kernel(u, a = 5)
```

Funkcja `strike_interpolate` wybiera metodę interpolacji na płaszczyźnie zmienności i cen wykonania (Vanna - Volga lub metoda jądrowa).

```
strike_interpolate(strikes, sigmas, DFs, tau, strike, smile_interp)
```

6.7 Odczyt wartości zmienności z siatki

Główna funkcja `ImpVol` odczytuje z siatki zmienności stworzonej przez funkcję `create_vol_table` wartość zmienności implikowanej dla zadanego strike'a oraz przedziału czasowego. Korzysta z funkcji: `VolsInTime` do interpolacji względem strike'a oraz `TimeInterpolate` do interpolacji po czasie.

Funkcja `VolsInTime` zwraca wektor volatilities otrzymanych z krzywej volatility smile dla danego strike'a w poszczególnych momentach czasu w siatce po czasie. Do interpolacji między strikami w siatce zastosowano metodę interpolacji liniowej.

```
% input: strike
% output: vols (wektor volatilities)

VolsInTime(strike)
```

Funkcja `TimeInterpolate` na podstawie wektora otrzymanego jako output funkcji `VolsInTime` zwraca wartość zmienności implikowanej dla wybranego momentu czasu, używając wybranej metody interpolacji po czasie (jak dotąd zaimplementowano metody `"linear"` (domyślna) oraz `"linear_on_variance"`).

```
% input:
%   start_date
%   expire_date
%   vols - wektor volatilities otrzymanych z krzywej volatility smile dla
%   danego strike'a w poszczególnych momentach czasu w T_grid (siatce po
%   czasie)
% output: wartość zmiennosci implikowanej dla okresu między start_date a expire_date

TimeInterpolate(start_date, expire_date, vols)
```

6.8 Funkcje pomocnicze przy obliczaniu miejsc zerowych

Funkcja `ffzero` oblicza miejsce zerowe dla danej funkcji f. Jest modyfikacją wbudowanej w Octave funkcji `fzero`.

```
% input: funkcja f
% output: miejsce zerowe

ffzero(f)
```

Funkcja `fmax` oblicza argument funkcji f, dla którego wartość pochodnej wynosi zero

```
% input:
%   funkcja f,
%   punkt startowy x0,
% output: argument funkcji f, dla którego wartość pochodnej wynosi zero

function result = fmax(f, x0)
```

Funkcja `derivative` oblicza pochodną numeryczną dla funkcji f i ustalonego punktu x.

```
% input:
%   funkcja f,
%   punkt startowy x,
% output: wartość pochodnej numerycznej

derivative(f, x)
```

6.9 Uśmiechy zmienność - rysunki

Funkcja `smile` rysuje krzywą zmienności implikowanej dla danej daty.

```
function smile(fixdate)
```

Funkcja `smile_i` rysuje krzywą zmiennosci implikowanej dla i-tego wiersza danych z rynku.

```
function smile_i(index)
```

Funkcja `vol_surface` rysuje powierzchnię zmienności implikowanej.

```
vol_surface( )
```

Literatura

[1] Dimitri Reiswich, Uwe Wystup *FX Volatility Smile Construction*, CPQF Working Paper Series No. 20, April 2010.

[2] Antonio Castagna, Fabio Mercurio, *Consistent Pricing of FX Options*, Banca IMI, Milan, 2007.

[3] Andrzej Palczewski, *Implied volatility The vanna-volga method and beyond*, Lecture notes.

[4] Frédéric Bossens, Grégory Rayée, Nikos S. Skantzos, Griselda Deelstra, *Vanna-Volga methods applied to FX derivatives: from theory to market practice*, Universite Libre de Bruxelles, Working Papers CEB, 09-016.RS., May 2010

[5] Peter Jäckel, "Monte Carlo Methods in Finance", John Wiley and Sons, Ltd., July 2002