

Dokumentacja

Uśmiech zmienności implikowanej

na rynku walutowym.

Joanna Pieniak

1 Opis problemu

Celem mojej 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 $S_t = \text{FOR}/\text{DOM}$ oznacza ilość jednostek waluty krajowej potrzebnej do zakupu jednej jednostki waluty obcej w chwili t . Zatem jeżeli $S_t = 3.8914 \text{ EUR}/\text{PLN}$, 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 Vanilla 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

$$\begin{aligned} V(S_t, K, \sigma, \omega) &= \omega DF_f(t, T) S_t N(\omega d_+) - DF_d(t, T) K N(\omega d_-)] = \\ &= \omega DF_f(t, T) [f(t, T) N(\omega d_+) - K N(\omega d_-)], \end{aligned}$$

gdzie

$$d_+ = \frac{\ln\left(\frac{f(t,T)}{K}\right) + \frac{1}{2}\sigma^2\tau}{\sigma\sqrt{\tau}},$$

$$d_- = d_+ - \sigma\sqrt{\tau},$$

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 domowej. Równoważnie można zapisać wartość kontraktu w walucie obcej $\frac{v}{S_t}$. Waluta, w której wyrażamy wartość opcji jest nazywana **premium currency**. Przyjmijemy 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

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

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.

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

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:¹

USD > EUR > GBP > AUD > NZD > CAD > CHF > NOK > SEK > DKK
> CZK > PLN > TRY > MXN > 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.

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 $K = f$,

¹dokładniej opisane w pracy Clark, I.J., Foreign Exchange Option Pricing: A Practitioner's Guide, Wiley, forthcoming

- ATM–value–neutral, dla K , takiego że V opcji call = V opcji put,
- ATM– Δ –neutral, dla K , takiego że V opcji call = $-V$ 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 między rodzajami delty a definicjami ATM.

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

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

$$K = f e^{-\omega N^{-1}(\omega\Delta_f)\sigma\sqrt{\tau} + \frac{1}{2}\sigma^2\tau}.$$

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

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

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

nie może być wyrażona wzorem analitycznym. Zatem, aby wskazać K , należy skorzystać z metod numerycznych obliczania miejsc zerowych. Na potrzeby programu wykorzystałam wbudowaną w Octave'a funkcję `fzero`, a także napisałam jej drobną modyfikację. 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. Wyróżniamy:

- 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 RR (risk reversal), zmienność implikowana z rynku dla różnicy delty opcji call o wartości 0.25 i delty put o wartości delty -0.25 .
- zmienność VWB (butterfly), zmienność implikowana dla różnicy połowy sumy delty opcji call o wartości 0.25 i delty put o wartości delty -0.25 oraz zmienności ATM.

Korzystając z powyższych danych rynkowych możemy wyliczyć zmienności implikowane odpowiadające odpowiednio opcji kupna, dla której wartość delty wynosi 0.25 (σ_{25C}) i opcji sprzedaży, dla której wartość delty wynosi -0.25 (σ_{25P}),

Po prostych przekształceniach uzyskujemy następujące wzory:

$$\sigma_{25C} = \sigma_{ATM} + \sigma_{25STR} + 0.5\sigma_{25RR},$$

$$\sigma_{25P} = \sigma_{ATM} + \sigma_{25STR} - 0.5\sigma_{25RR}.$$

Kolejnym etapem jest wyliczenie strików dla danej σ_{ATM} i uzyskanych σ_{25C} , σ_{25P} . 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

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

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_{25C} , σ_{25P} i przyjętych konwencji obliczania delt, stosujemy wzory z poprzedniego podrozdziału (2.6)

$$\Delta_{(\sigma_{25C}, K_{25C})} = 0.25,$$

$$\Delta_{(\sigma_{25P}, K_{25P})} = -0.25,$$

dla obliczenia K_{25C} , K_{25P} .

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_{25P} , K_{ATM} , K_{25C} oraz odpowiadającym im zmiennościach implikowanych σ_{25P} , σ_{ATM} , σ_{25C} przeprowadzamy interpolację między punktową. W tym celu zaimplementowałam metodę Vanna–Volga.

Upraszczając notację przyjmę 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:

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

Wzory Blacka Scholesa pozwalają uzyskać analityczną postać:

$$\begin{aligned}\nu(t, K) &:= \frac{\delta C_{BS}(t; K)}{\delta \sigma} \\ \frac{\delta C_{BS}(t; K)}{\delta \sigma} &= S_t D F_f(t, T) \sqrt{\tau} \phi(d_+(t, K)), \\ \frac{\delta^2 C_{BS}(t; K)}{\delta \sigma^2} &= \frac{\nu(t, K)}{\sigma} d_+(t, K) d_-(t, K), \\ \frac{\delta^2 C_{BS}(t; K)}{\delta \sigma \delta S_t} &= -\frac{\nu(t, K)}{S_t \sigma \sqrt{\tau}} d_-(t, K).\end{aligned}$$

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

$$\begin{aligned}x_1 &= \frac{\nu(t, K) \ln(K_2/K) \ln(K_3/K)}{\nu(t, K_1) \ln(K_2/K_1) \ln(K_3/K_1)}, \\ x_2 &= \frac{\nu(t, K) \ln(K_1/K) \ln(K_3/K)}{\nu(t, K_2) \ln(K_1/K_2) \ln(K_3/K_2)}, \\ x_3 &= \frac{\nu(t, K) \ln(K_1/K) \ln(K_2/K)}{\nu(t, K_3) \ln(K_1/K_3) \ln(K_2/K_3)}.\end{aligned}$$

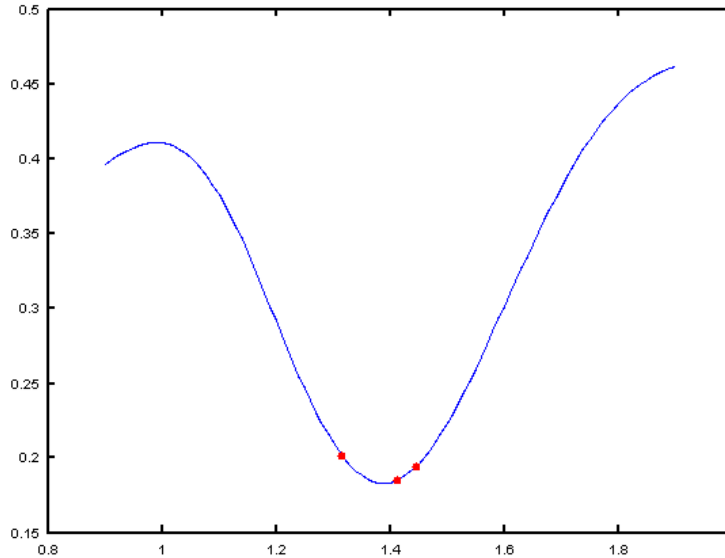
Kwintesencją powyższych rozważań jest alternatywny wzór na cenę opcji dla danego t i K , z który implikuje krzywą zmienności:

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

gdzie $C_{MKT}(t, K_i)$ 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 $\tau = 1Y$ 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 $i = 1, \dots, n$ szacujemy wartość Y , dla danego punktu x korzystamy z następującego wzoru

$$\hat{Y}(x) = \frac{\sum_{i=1}^n Y_i F\left(\frac{x_i - x}{h}\right)}{\sum_{i=1}^n F\left(\frac{x_i - x}{h}\right)}, \quad (4)$$

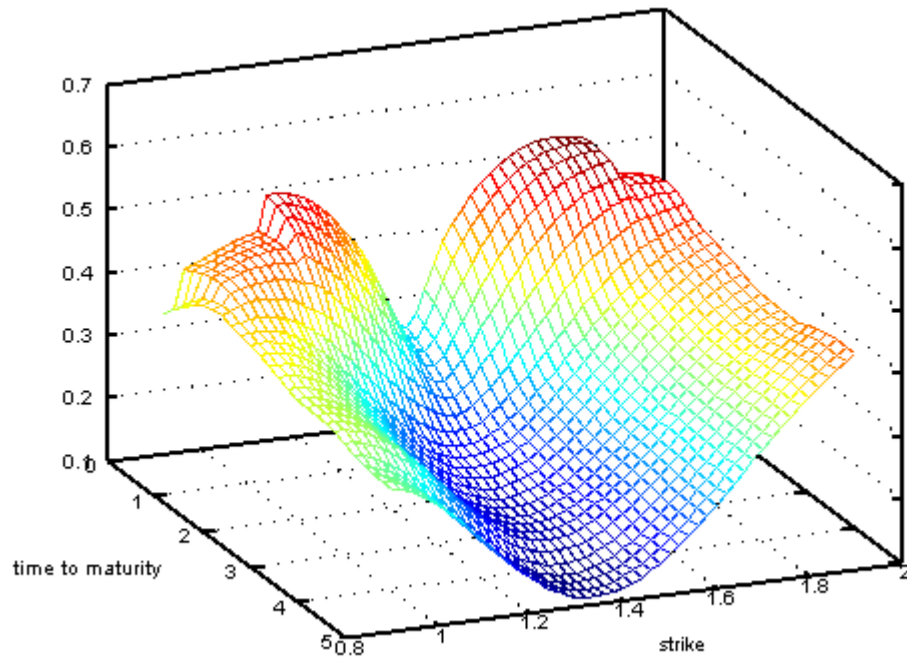
gdzie $F(\cdot)$ jest tak zwaną funkcją jądrową o współczynniku h . W programie korzystam z funkcji gausowskiej, danej wzorem:

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

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

5 Wyniki

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



6 Dokumentacja

Funkcja `gen_fx_market` generuje plik `fx_market.m`, korzystając z danych w pliku `fx_vol.m`, umożliwiając użytkownikowi ingerowanie w dane i wprowadzenie zmian. Należy wcześniej wykonać funkcje `PREM_CURR` i `delta_convention`, opisane poniżej.

Funkcja `get_fx_vol_data` wczytuje dane z pliku `fx_market` lub zwraca, jeżeli wcześniej były wczytane i przygotowuje dane (ceny wykonania, zmienności i współczynniki dyskontowe) do późniejszego użycia.

% output: dane przygotowane do późniejszego użycia

```
get_fx_vol_data()
```

Ukoronowaniem pracy jest funkcja `ImpVol`, która zwraca wartość zmienności implikowanej dla danych początkowych. Korzysta z funkcji pomocniczej `imp_vol`. Pierwsze wywołanie funkcji `ImpVol` zajmuje 1.45s., a każde kolejne 0.07s., dzięki zapamiętywaniu danych.

% input:

```
% strike - cena wykonania
% start_date - data początkowa
% expire_date - data zapadalności
% vol_param - dodatkowe parametry, kolejno:
% metoda interpolacji po cenie, metoda interpolacji po czasie
% (domyślnie vol_param={"vanna-volga", "kernel"})
%
```

% output: wartość zmienności implikowanej

```
ImpVol(strike, start_date, expire_date, vol_param)
```

% przykładowe wywołanie

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

Funkcja `imp_vol` zwraca wartość zmienności implikowanej dla danego strika, czasu i wybranych metod interpolacyjnych. Jako funkcja pomocnicza różni się od `ImpVol` parametrami input.

% input:

```
% strike - cena wykonania
% t - czas do zapadalności (ułamek roku)
% smile_interp - metoda interpolacji po cenie (domyślnie "vanna-volga")
% time_interp - metoda interpolacji po czasie (domyślnie "kernel")
% output: wartość zmienności implikowanej
```

```
imp_vol(strike, t, smile_interp = "vanna-volga", time_interp="kernel")
```

W następujących podrozdziałach krótko opiszę wszystkie funkcje pomocnicze, z których korzystałam przy implementacji funkcji `ImpVol`. Aby dobrze zrozumieć działanie programu należy dokładnie prześledzić opis teoretyczny.

6.1 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)
```

Funkcja `ATM_convention` oblicza konwencję wyboru strike dla σ_{ATM} , przekazując jeden z parametrów: `delta_neutral`, `atm_fwd`, `atm_spot`, `atm_value_neutral` (domyślnie `delta_neutral`).

```
% input:
%   S_0 - cena początkowa,
%   f = FX_rate * DF_d/DF_f,
%   sigma_atm - zmienność atm,
%   tau - czas do zapadalności,
%   K_conv - konwencje wyboru atm,
% output: K_atm
```

```
ATM_convention(S_0, f, sigma_atm, tau, K_conv = "delta_neutral")
```

Funkcje `spot_un`, `spot_pa`, `fwd_un`, `fwd_pa` obliczają ceny wykonania K_{ATM} , K_{25C} , K_{25P} odpowiadającym im σ_{ATM} , σ_{25C} , σ_{25P} odpowiednio, w zależności od wyboru konwencji. W przypadku funkcji `spot_un` i `fwd_un` korzystam z gotowych wzorów analitycznych opisanych w rozdziale 2.6. Dla funkcji `spot_pa` i `fwd_pa` niezbędne jest obliczenie miejsca zerowego formuł (1) i (2). W tym celu korzystam 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_25C, sigma_25P, sigma_atm - zmienności dla delt 25C, 25P, ATM
```

```

% S_0 - cena początkowa,
% f = FX_rate * DF_d/DF_f,
% tau - czas do zapadalności,
% K_conv - konwencje wyboru atm,
% DF_f, DF_d - czynniki dyskontowe, zagraniczny i krajowy odpowiednio
% output:
% wektor [K_atm, K_25C, K_25P]

spot_un(sigma_25C, sigma_25P, sigma_atm, S_0, f, tau, DF_f, DF_d, K_conv)
spot_pa(sigma_25C, sigma_25P, sigma_atm, S_0, f, tau, DF_f, DF_d, K_conv)
fwd_un(sigma_25C, sigma_25P, sigma_atm, S_0, f, tau, DF_f, DF_d, K_conv)
fwd_pa(sigma_25C, sigma_25P, sigma_atm, S_0, f, tau, DF_f, DF_d, K_conv)

% funkcje pomocnicze
spot_pa_call(K, f, sigma_25C, tau, DF_f, DF_d)
spot_pa_put(K, f, sigma_25P, tau, DF_f, DF_d)
fwd_pa_call(K, f, sigma_25C, tau, DF_f, DF_d)
fwd_pa_put(K, f, sigma_25P, tau, DF_f, DF_d)

```

Funkcja VOL zwraca wartość cen wykonania i zmienności dla delt 25C, ATM, 25P.

```

% input:
% expiry - data zapadalności,
% sigma_atm, sigma_25rr, sigma_25str - dane odczytane z rynku,
% K_conv = {delta_neutral, atm_fwd, atm_spot, atm_value_neutral},
% conv = {SPOT_UN, SPOT_PA, FORWARD_UN, FORWARD_PA},
% prem_curr - waluta premium
%
% output:
% 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,

function strikes, sigmas, DFs, tau = VOL(expiry, sigma_atm,
    sigma_25rr, sigma_25str, K_conv = 'delta_neutral', conv, prem_curr)

```

6.2 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 TODO, 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 (3). Korzysta z funkcji `vanna_volga`, `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, dla której interpolujemy wartość sigma
% output: wartość opcji kupna

```

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

Funkcja `C` oblicza lewą stronę wyrażenia (3), 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)
```

6.3 Interpolacja metodą jądrową

Funkcja `kernel_interpolate` oblicza wartość wyrażenia (4), 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)
```

6.4 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)
```

6.5 Uśmiechy zmienność - rysunki

Funkcja `smile` rysuje i zwraca krzywą zmienności implikowanej dla danego τ .

```
function X, Y = smile(tau, smile_interp = "vanna-volga", time_interp = "kernel")
```

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

```
function X, Y = smile_i(index, smile_interp = "vanna-volga")
```

Funkcja `vol_surface` rysuje i zwraca powierzchnię zmienności implikowanej.

```
vol_surface(smile_interp = "vanna-volga", time_interp = "kernel")
```

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 beyond*, Lecture notes.