La volatilidad implícita

Los mercados de opciones han evolucionado bastante desde los años setentas, época en la que fue publicada la fórmula de Black Scholes (BS). Dicha fórmula quedó tan arraigada en la mente de los traders que aún hoy en día sigue siendo ampliamente utilizada a pesar de sus falencias. Una de las más grandes herencias que obtuvo el mercado por el uso de esta fórmula, fue la posibilidad de calcular la volatilidad implícita. Esta medida actúa de una manera muy similar a como funciona el rendimiento hasta el vencimiento de un bono. Cuando se usa el rendimiento hasta el vencimiento, los traders pueden comparar de una manera fácil el valor relativo de bonos con diferentes vencimientos o calidades crediticias sin necesidad de mirar el precio limpio. De la misma forma los operadores de opciones miran la volatilidad implícita para comparar diferentes opciones sin necesidad de mirar las primas a las que se negocian en el mercado. La volatilidad implícita también se utiliza bastante para compararla con la volatilidad realizada del activo subyacente.

Hay diferentes maneras de calcular la volatilidad realizada. Allí hacemos uso de la información histórica del activo subyacente, particularmente de los retornos del mismo. No vamos a tratar el tema en este documento ya que nos ocupáremos esencialmente del cálculo de la volatilidad implícita. Para trabajar este tema es fundamental que estemos familiarizados con la fórmula de BS (Se puede consultar nuestro documento “Lá Fórmula de Black Scholes” en nuestra sección de HERRAMIENTAS).

La volatilidad implícita es aquella que habría que ingresar a la fórmula de BS para obtener un precio observado en el mercado. Sabemos que el precio de una opción bajo el paradigma de Black-Scholes es una función del Spot (S), el Strike (K), la tasa de interés (r), el tiempo al vencimiento (T) y la volatilidad (σ). Esto es

$$CALL = f(S, K, r, T, \sigma)$$

Supongamos que tenemos un mercado de opciones en donde podemos observar las primas para diferentes strikes y vencimientos. Lo único que tenemos que hacer para calcular la volatilidad implícita es mirar el precio spot del activo subyacente (S) y la tasa de interés de ese plazo (r). Con estos datos podemos deducir que valor de la volatilidad σ, me genera una prima como la observada en el mercado. Veamos un ejemplo. Supongamos que tenemos los siguientes datos: S = 100, K=110, r=0.05, T=0.25, CALL=0.5317

Podemos utilizar la función que construimos en Python, “bsimpvol”, para calcular cual sería la volatilidad implícita en este precio. Esto se ilustra en la Figura 1. Allí podemos ver que la volatilidad implícita dados estos valores es del 15% aproximadamente. Esto es consistente con el ejemplo que ilustramos en nuestro documento “La Fórmula de Black Scholes”.

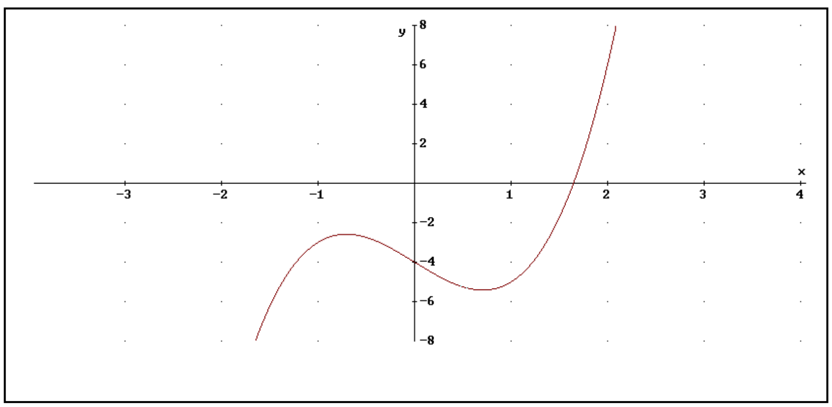
Figura 1

```
>>> bsimpvol('call',100,110,0.05,0.25,0.5317,q=0.0,pricetolerance=0.01,method='bisect')
array([ 0.1 ,  0.14,  0.16,  0.15])

>>> bsimpvol('call',100,110,0.05,0.25,0.5317,q=0.0,pricetolerance=0.01,method='newton')
array([ 0.1        ,  0.18073884,  0.15248155,  0.15001657])
```

Ahora surge la pregunta de como obtuvimos el valor del 15%. Desafortunadamente la fórmula de BS no permite que tengamos una solución analítica. Esto debido a que no podemos despejar explícitamente el valor σ en función de los otros parámetros. Por esta razón, si queremos calcular la volatilidad implícita de una opción, tenemos que utilizar un método numérico. Este problema es el mismo que el de encontrar las raíces de una ecuación del tipo f(x) = 0. Veamos por ejemplo la función $f(x) = 2x^3 - 3x - 4$ en la Figura 2. Allí vemos que esta función tiene una raíz en el intervalo [1,2] y a simple vista pareciera que está alrededor de 1.6 o 1.7.

Figura 2



Los algoritmos más utilizados para solucionar este tipo de problemas cuando x no se puede despejar explícitamente son dos:

a) El método de la Bisección.

b) El método de Newton-Raphson.

Estos algoritmos los explicamos con detalle en el Anexo 1 y 2 al final de este documento. La intuición básica de los mismos es que vamos buscando iterativamente un valor que haga que la función sea igual a un valor objetivo. En el caso de la Figura 2, tenemos que buscar un valor (que creemos está cerca de 1.6 o 1.7 por inspección visual) que haga que la función valga cero.

El algoritmo se puede modificar de una forma simple para que la función tenga un valor específico. En nuestra aplicación puntual de la volatilidad implícita el valor objetivo va a ser la prima de la opción y el valor que vamos a iterar es el de la volatilidad implícita. En EXCEL se puede hacer esto usando la función "Buscar Objetivo". Allí vemos que el sistema itera hasta que nos da un valor de la volatilidad implícita que coincide con el objetivo de prima que introducimos inicialmente. EXCEL utiliza un procedimiento iterativo que no podemos observar explícitamente. Con nuestros archivos de Python aprendemos a generar este algoritmo.

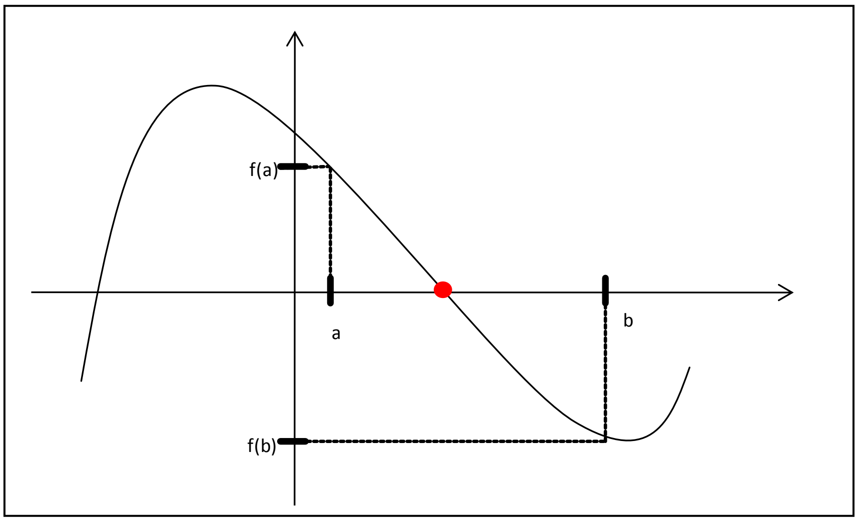
Recordemos que PYTHON es un programa de uso libre por el cual no tenemos que pagar una licencia. Adjuntamos tres archivos de Python que utilizamos para calcular la volatilidad implícita. Los archivos llamados Bisect.py y NewtonRaphson.py contienen los algoritmos numéricos que iteran para hallar el valor de una raíz. El archivo BS.py contiene una función llamada bsimpvol que hace uso de los módulos anteriores para calcular específicamente el valor de la volatilidad implícita utilizando la fórmula de BS.

Mencionamos al principio del documento que la volatilidad implícita es utilizada con fines comparativos. Veamos un ejemplo de cómo comparar la volatilidad implícita y la volatilidad realizada. La volatilidad realizada del S&P500 durante Enero de 2013 fue de aproximadamente 11.5%. A principios de este mes la volatilidad implícita de opciones At The Money (ATM) para el plazo de un mes sobre este índice estaba alrededor del 13%. Un operador de opciones podría haber realizado una estrategia de gamma scalping vendiendo Straddles ATM y cubriendo el delta en el spot para muy seguramente obtener una utilidad de 1.5 volatilidades. Algunos de estos términos como "Gamma Scalping", "Straddles", "Delta", etc, no los hemos explicado en este documento. En nuestro curso **OPCIONES INTERMEDIO** explicamos con detalle esta estrategia que es más conocida como "Arbitraje de Volatilidad".

ANEXO 1: Método de la bisección.

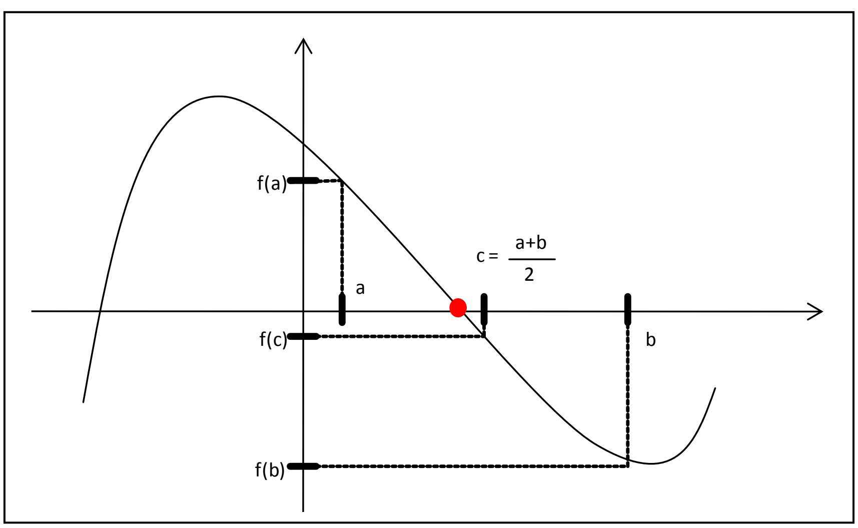
El método de la bisección es el más simple para resolver la ecuación escalar f(x) = 0. Lo único que necesitamos en este método es poder evaluar la función f en cualquier punto. Supongamos que conocemos dos puntos a y b con a<b. Supongamos adicionalmente que la función es continua en el intervalo [a,b] y que f(a)*f(b) < 0. Esta última condición indica que la ecuación tiene una raíz en el intervalo [a,b] ya que la función pasa de positivo a negativo (o viceversa). Una función que cumple las anteriores condiciones puede ser observada en la Figura 3.

Figura 3



El punto rojo ilustra una raíz de la ecuación f(x) = 0. A continuación podemos proceder a reducir el tamaño del intervalo para acercarnos más al valor buscado. El método de la bisección toma como aproximación el punto c, equivalente al punto medio entre a y b. Este se ilustra en la Figura 4.

Figura 3



Allí podemos observar que el intervalo [a,c] es de menor longitud que el intervalo [a,b] y que igualmente contiene la raíz de la ecuación f(x) = 0. Para escoger el intervalo siempre debemos chequear si la raíz está contenida en [a,c] o [c,b], esto es f(a)*f(c) < 0 o f(c)*f(b) < 0. También podría ser que f(c)≈ 0 con un nivel de tolerancia predefinido. En este caso no tendríamos que iterar.

En la siguiente iteración buscaríamos el punto medio del intervalo [a,c] y continuaríamos hasta que alguna de las siguientes condiciones se cumpla:

a) El valor de la función sea muy cercano a cero (menor que una tolerancia ε) -> $|f(c_n)| < \varepsilon$
b) El valor del intervalo $[a_n,b_n]$ sea muy pequeño (menor que una tolerancia δ) -> $a_n-b_n < \delta$
c) Se haya alcanzado un número máximo de iteraciones.

Esta descripción está basada en el libro **Numerical Methods in Finance and Economics** del autor **Paolo Brandimarte**. Allí se pueden encontrar muchos más detalles como la tasa de convergencia o el error esperado del método.

ANEXO 2: Método de Newton Raphson.

A diferencia del método de la bisección, este método necesita mayor información de la función f, particularmente su primera derivada. La deducción del método se puede ilustrar usando la expansión de una serie de Taylor. Recordemos que una función puede ser aproximada usando la serie de Taylor como se ilustra a continuación. En este caso lo hacemos en el entorno del punto x_n

$$f(x) = f(x_n) + \frac{f'(x_n)}{1!}(x - x_n) + \frac{f''(x_n)}{2!}(x - x_n)^2 + \ldots$$

Si tomamos un punto cualquiera x_{n+1} y sólo expandimos los dos primeros términos de la serie obtenemos

$$f(x_{n+1}) = f(x_n) + f'(x_n)(x_{n+1} - x_n)$$

Ahora bien, si el valor x_{n+1} es una raíz de la ecuación f(x) = 0 (o está cerca de ser una raíz) tendríamos que

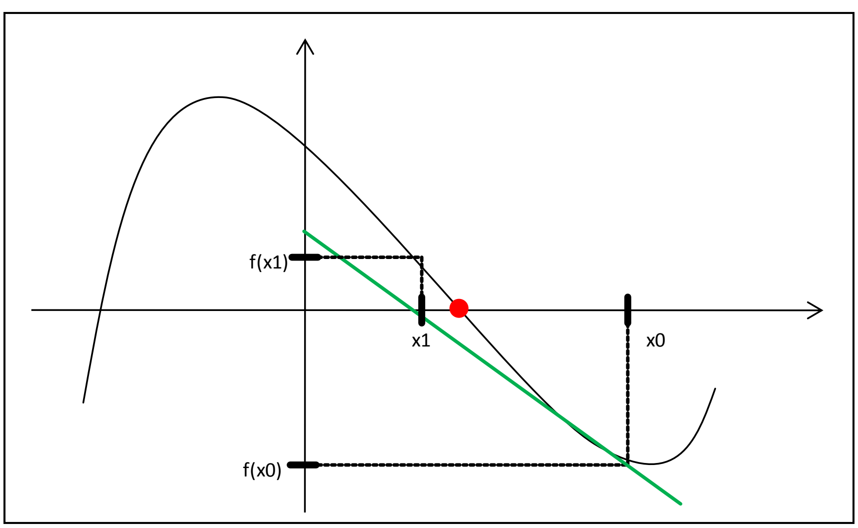
$f(x_{n+1}) \approx 0$ y por lo tanto

$$x_{n+1} \approx x_n - \frac{f(x_n)}{f'(x_n)}$$

Esta ecuación ilustra la forma en la que podemos iterar para llegar a una solución.

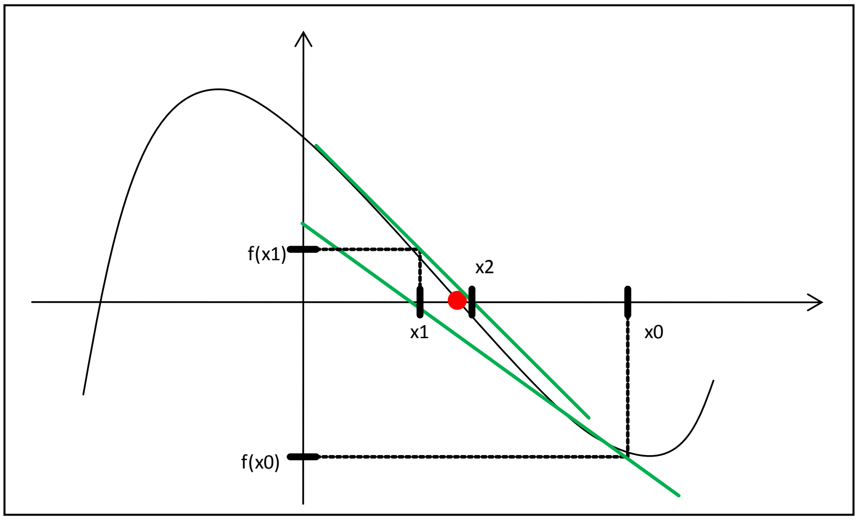
Geométricamente el método se ilustra en las Figuras 3 y 4. En la Figura 3 podemos observar el punto x_0. Allí evaluamos la derivada de la función en este punto $f'(x_0)$ y proyectamos la recta tangente al eje x. El intercepto con el eje x, en este caso x_1 en un valor más cercano a la raíz de la ecuación f(x) = 0.

Figura 3



Si hacemos una nueva iteración podemos observar que x_2 está aún más cerca de ser la raíz.

Figura 4



Iteramos hasta que alguna de las siguientes condiciones se cumpla:

a) El valor de la función sea muy cercano a cero (menor que una tolerancia ε) -> $|f(c_n)| < \varepsilon$
b) Se haya alcanzado un número máximo de iteraciones.