

Facultad de Ciencias Económicas y Empresariales

Doble Grado Economía-Matemáticas y Estadística

Trabajo de Fin de Grado



UNIVERSIDAD
COMPLUTENSE
MADRID

Estimación de las “griegas” de opciones europeas a partir de simulaciones Monte Carlo

Fátima Lastra Incera

Tutora: María del Pilar Grau Carles

Curso académico: 2024-2025

Convocatoria: Junio

Resumen

La estimación de las sensibilidades del precio de una opción financiera, conocidas como las ‘griegas’, es una cuestión fundamental en el campo de las finanzas cuantitativas. Estas sensibilidades se definen como las derivadas parciales del precio de una opción con respecto a parámetros clave, como el precio del activo subyacente, la volatilidad, el tiempo o la tasa de interés del activo libre de riesgo. Este Trabajo de Fin de Grado aborda la comparación y evaluación de diferentes métodos para estimar las sensibilidades en el contexto de opciones europeas. Los métodos que se estudian incluyen: el método de diferencias finitas, el método de trayectoria y el método de razón de verosimilitud. Esta comparación se basa en aplicar los métodos mencionados a una cartera de opciones financieras sobre materias primas y acciones que no distribuyen dividendos. Los resultados obtenidos muestran que, los métodos de diferencias finitas y el de trayectoria presentan estimaciones de las griegas muy cercanas a las obtenidas por el modelo de Black and Scholes [1973]. El método de razón de verosimilitudes presenta buenos resultados, aunque su desempeño es inferior en la estimación de Rho. Los métodos analizados han demostrado ser útiles para obtener las griegas de ciertas opciones, aunque se han identificado áreas de mejora, particularmente cuando se utilizan fechas de vencimiento más largas. De modo que, la elección del método más adecuado depende de la griega que se desea estimar, el tipo de opción y las condiciones del mercado.

Palabras clave: valoración de opciones, griegas, Black-Scholes, Simulación de Monte Carlo, Movimiento Browniano Geométrico, diferencias finitas, trayectorias, razón de verosimilitudes.

Abstract

The estimation of the price sensitivities of a financial option, known as the "Greeks" is fundamental aspect in the field of quantitative finance. These sensitivities are defined as the partial derivatives of the option's price with respect to key parameters, such as the spot price, volatility, time, and risk-free asset interest rate. This undergraduate thesis addresses the comparison and evaluation of different methods to estimate these sensitivities in the context of European options. The methods studied include: the finite differences method, the trajectory method, and the likelihood ratio method. This comparison involves applying the aforementioned methods to a portfolio of real financial options on commodities and non-dividend-paying stocks. The results show that, in general, the finite differences and trajectory methods perform satisfactorily, with estimates very close to those obtained by the model of Black and Scholes [1973]. The likelihood ratio method presents good results, although its performance was inferior in the estimation of Rho. The methods analyzed is useful for obtaining the "Greeks" of certain options, although areas for improvement were identified, particularly when using longer maturities. The choice of method should depend on the Greek to be estimated, the type of option, and market conditions.

Keywords: option pricing, Greeks, Black-Scholes, Monte Carlo Simulation, Geometric Brownian Motion, Finite Differences, trajectories, likelihood ratio.

Índice

1	Introducción	1
1.1	Motivación e hipótesis	3
1.2	Contribuciones	3
2	Marco teórico	4
2.1	Fundamentos de valoración de derivados	4
2.1.1	Conceptos básicos	4
2.1.2	Modelo Black-Scholes	9
2.1.3	Las griegas	13
2.2	Fundamentos de Monte Carlo	16
2.2.1	Simulación Monte Carlo	16
2.2.2	Generación de variables aleatorias	21
3	Metodología	24
3.1	Método de las diferencias finitas	24
3.2	Método de trayectoria	29
3.3	Método de razón de verosimilitudes	34
4	Datos	40
4.1	Cartera de opciones	40
4.2	Fuentes de datos	41
5	Resultados	44
6	Discusión	48
7	Conclusión	50
	Referencias	I
	Apéndice A: Cálculo estocástico	II
	Apéndice B: Código de Python	v

Capítulo 1

Introducción

La valoración de opciones financieras y, en particular, el cálculo preciso de las sensibilidades al precio de las mismas, conocidas como las “griegas” es un tema central de investigación en el campo de las finanzas cuantitativas. Estas métricas son utilizadas especialmente por los *traders* tanto para cuantificar riesgos asociados a las posiciones asociadas a las opciones financieras como para realizar estrategias de cobertura que minimicen el riesgo al que se exponen al tomar dichas posiciones. De modo que, resulta esencial contar con métodos que permitan obtener su valor de manera precisa.

Las opciones financieras son contratos financieros derivados que otorgan a su titular el derecho, pero no la obligación, de comprar o vender un activo subyacente a un precio previamente acordado, conocido como el precio de ejercicio, dentro de un plazo determinado. Existen dos tipos principales de opciones: las **opciones de compra**, que otorgan el derecho de comprar el activo subyacente, y las **opciones de venta**, que permiten venderlo. Los participantes del mercado incluyen compradores y vendedores de ambos tipos de opciones. Los compradores pagan una prima, conocida como el precio de la opción, a cambio del derecho a ejercer la opción en el futuro, mientras que los vendedores asumen la obligación de cumplir con los términos del contrato a cambio de recibir esa prima.

El problema fundamental en este contexto es determinar un **precio justo** de las distintas opciones financieras existentes, lo que se conoce como el problema de la *valoración de opciones*, que es clave para garantizar el equilibrio en los mercados financieros.

Los modelos exactos para la determinación del valor de las opciones financieras y sus griegas asociadas hacen suposiciones muy restrictivas, que no reflejan la realidad de los mercados financieros y no son útiles cuando las opciones financieras son más complejas. El modelo que más impacto ha tenido en este ámbito fue el propuesto por Black and Scholes [1973], los cuales desarrollaron fórmulas cerradas para la valoración de opciones europeas. No obstante, este modelo supone, entre otras cosas, que la volatilidad del activo es constante a lo largo del tiempo, que los mercados son libres de fricciones o que los activos no pagan dividendos. En la literatura se han propuesto modelos de solución cerrada que tratan de cubrir estas limitaciones, pero cuando se trata de opciones financieras más complejas es necesario recurrir a métodos numéricos.

En este marco, el principal problema de investigación abordado en este trabajo se centra en el cálculo de las griegas de opciones europeas, que se definen como las derivadas del precio de un derivado con respecto a varios parámetros del modelo, tales como el precio del activo subyacente, la volatilidad, el tiempo hasta el vencimiento y la tasa de interés del activo libre de riesgo. Las griegas de las opciones europeas tienen fórmulas derivadas de la solución cerrada propuesta por Black and Scholes. Sin embargo, como hemos mencionado previamente en el caso de otros tipos de opciones más complejas es necesario recurrir a métodos numéricos para valorarlas y, por ende, también tendremos que calcular las griegas de forma aproximada.

En la literatura existen diversas familias de métodos numéricos destinados a abordar el problema de estimar las sensibilidades de opciones financieras. Entre las técnicas más utilizadas destaca la simulación Monte Carlo. Boyle [1977] fue pionero en la aplicación de técnicas de simulación de Monte Carlo para valorar opciones europeas y, por tanto, también para obtener el valor de las griegas. Glasserman [2010] propone dos categorías fundamentales de métodos de estimación de las griegas que utilizan simulación Monte Carlo: (1) los métodos que requieren simular en dos o más valores del parámetro de diferenciación, como el *método de diferencias finitas*, y (2) los que no requieren esta simulación múltiple, como el *método de trayectoria* y el *método de razón de verosimilitud*. En ambos casos, se emplea la simulación Monte Carlo para calcular el precio de las opciones, permitiendo así estimar las griegas de manera aproximada en escenarios donde las soluciones analíticas no son posibles.

En este trabajo nos centramos en la estimación de las griegas Delta, Gamma, Theta, Vega y Rho, dentro del marco del modelo de Black and Scholes [1973] para opciones europeas. El objetivo principal es comparar distintos métodos numéricos de estimación de las sensibilidades de opciones europeas con las fórmulas analíticas derivadas de dicho modelo, con el fin de evaluar la eficacia y precisión de cada enfoque en el cálculo de las sensibilidades. La métrica de error escogida es la raíz del error cuadrático medio (RMSE), considerando como valor de referencia el obtenido con las fórmulas de Black and Scholes.

Los tres métodos de estimación se han aplicado a una cartera compuesta por 20 opciones europeas sobre materias primas y 10 opciones europeas sobre acciones que no pagan dividendos, tanto de compra como de venta. La fecha de valoración considerada es el 1 de diciembre de 2016, y se han analizado cinco vencimientos distintos a 3, 6, 9, 12 y 24 meses, lo que permite evaluar el comportamiento de los métodos en distintos horizontes temporales y en un contexto representativo de condiciones de mercado reales. Los datos se han obtenido de la plataforma *Yahoo Finance* mediante la librería *yfinance* de *Python*, y toda la implementación de los métodos, las simulaciones y la generación de gráficos se ha desarrollado en el lenguaje de programación *Python* en el entorno de *Spyder*.

Los resultados obtenidos muestran que los métodos de diferencias finitas y de trayectoria presentan, en general, una buena precisión en la estimación de las griegas, ofreciendo valores muy cercanos a los obtenidos mediante las fórmulas analíticas del modelo de Black and Scholes. En cambio, el método de razón de verosimilitudes muestra una menor precisión, especialmente en la estimación de la griega Rho, donde los errores aumentan considerablemente con la fecha de vencimiento. Esto nos permite concluir que es importante seleccionar el método de estimación más adecuado según la griega que se desea calcular, el horizonte temporal de la opción y las condiciones del mercado.

1.1. Motivación e hipótesis

El objetivo principal de este trabajo es la evaluación y comparación de tres métodos de estimación de las griegas de opciones financieras europeas: (1) el método de diferencias finitas, (2) el método de la trayectoria y (3) el método de razón de verosimilitudes, todos ellos implementados mediante simulación Monte Carlo. Para ello, se considera una cartera compuesta por opciones europeas de compra y venta sobre 20 materias primas y 10 acciones sin dividendos para cinco vencimientos diferentes y niveles de moneyness de 0.9 y 1.1, con el objetivo de responder a las siguientes preguntas de investigación:

- P1** ¿Cuál de los tres métodos de estimación de las griegas mediante simulación Monte Carlo proporciona mayor precisión en el caso de opciones europeas?
- P2** ¿Cómo varía el desempeño de los métodos en función del tipo de activo subyacente y las características de la cartera analizada?
- P3** ¿Qué método ofrece un mejor desempeño en la estimación de las griegas para opciones de largo vencimiento en comparación con opciones de corto vencimiento?

1.2. Contribuciones

Nuestras principales contribuciones son las siguientes. Primero, se presentan los diferentes métodos de estimación que se van a emplear en el análisis, junto con las expresiones explícitas para el cálculo de cada una de las griegas. Estas expresiones están deducidas y demostradas rigurosamente, mostrando paso a paso su obtención para cada uno de los métodos. Segundo, se presenta un análisis comparativo de tres métodos de simulación para la estimación de las griegas de opciones europeas, evaluando su precisión en diferentes condiciones de mercado. Por último, se proponen áreas de mejora, como la optimización del número de trayectorias simuladas y el uso de métodos avanzados para mejorar la precisión en escenarios de alta volatilidad o de largo vencimiento.

Capítulo 2

Marco teórico

En este capítulo presentamos los conceptos fundamentales que se emplean en el desarrollo de este trabajo. En particular, abordamos la teoría acerca de la valoración de derivados financieros (apartado 2.1) y los fundamentos del método de simulación de Monte Carlo (apartado 2.2). Estos elementos son esenciales para comprender los métodos empleados en la estimación de las sensibilidades del precio de los derivados financieros comúnmente denominadas griegas.

2.1. Fundamentos de valoración de derivados

El contenido de este apartado se basa en los manuales de Glasserman [2010], Hull and White [1987], McDonald [2012] y Clark [2014], que constituyen referencias clave en el estudio de la valoración de derivados y sus sensibilidades.

2.1.1. Conceptos básicos

En las últimas décadas, el mundo financiero ha experimentado una transformación drástica, en la que los derivados han desempeñado un papel central como uno de los instrumentos más relevantes de esta evolución.

Un *derivado financiero* es un instrumento financiero cuyo valor depende de la evolución de los precios de otros activos denominados subyacentes, que pueden ser un índice bursátil, acciones, valores de renta fija, divisas, materias primas, deuda, tipos de interés, etc. Los derivados tienen su origen en un contrato entre dos partes, ya sea de carácter obligatorio u opcional, a partir del cual se llega a un compromiso de entrega de un activo a cambio del pago de un precio. No obstante, el intercambio se produce en una fecha posterior a la formalización del contrato.

Los derivados se intercambian en dos tipos de mercados principales. Por un lado, encontramos los *mercados de valores organizados*, que son mercados centralizados y regulados donde se negocian productos derivados estandarizados, es decir, las partes no pueden modificar su contenido. El primer mercado que sentó las bases de los actuales mercados de derivados fue el Chicago Board of Trade (CBOT), constituido en 1848. Este mercado fue pionero en la creación de mercados de futuros, inicialmente diseñados para

estandarizar el comercio de granos de cereales tras una escalada de precios causada por una inundación masiva que afectó la producción. A partir de entonces, surgieron muchos otros mercados organizados. En España, un ejemplo destacado es el Mercado Oficial de Opciones y Futuros Financieros (MEFF).

Por otro lado, están los *mercados over-the-counter* (OTC), que funcionan como acuerdos privados entre dos partes, generalmente instituciones financieras o grandes empresas. A diferencia de los mercados organizados, las transacciones en los mercados OTC no están estandarizadas, lo que permite una mayor flexibilidad para personalizar los contratos según las necesidades de las partes. Sin embargo, esta flexibilidad también implica un mayor riesgo de crédito y una menor transparencia en las operaciones.

Los derivados se han convertido en una herramienta esencial en la gestión de riesgos, siendo ampliamente utilizados para protegerse contra fluctuaciones de precios, tasas de interés, o movimientos en los tipos de cambio. A pesar que algunas pérdidas notorias relacionadas con derivados han atraído atención mediática, entre ellas la crisis financiera del 2008, la realidad es que estos instrumentos son de uso común en las finanzas modernas.

Los derivados pueden clasificarse en diversos tipos según la naturaleza del contrato, como futuros, opciones, swaps y forwards, entre otros. No obstante, en este trabajo nos enfocamos en las **opciones financieras**, específicamente en las **opciones europeas**. A continuación, se define el concepto de opción financiera.

Definición 2.1.1 (Opción financiera). Una **opción financiera** es un contrato financiero derivado que otorga a su propietario (el titular) el derecho, pero no la obligación, de comprar o vender una cantidad específica de un activo subyacente a un precio establecido de antemano (conocido como **precio de ejercicio** o strike price) en una fecha futura conocida (la **fecha de vencimiento** o madurez), ya sea en una fecha específica (en el caso de una **opción europea**) o en cualquier momento dentro de un período determinado (en el caso de una **opción americana**). El comprador de la opción paga una prima, conocida como **precio de la opción**, al vendedor de la misma a cambio del derecho que obtiene.

Las opciones se clasifican en dos tipos, una **opción de compra** (call), que otorga al titular el derecho de adquirir el activo subyacente a un precio de ejercicio antes de una fecha determinada, y la **opción de venta** (put), que otorga al titular el derecho de vender el activo subyacente a un precio de ejercicio antes de una fecha determinada.

En los mercados de opciones, existen cuatro tipos de participantes: compradores de opciones de compra, vendedores de opciones de compra, compradores de opciones de venta y vendedores de opciones de venta, donde los compradores tienen **posiciones largas** y los vendedores tienen **posiciones cortas**. La diferencia entre ambos es que el comprador

tiene el derecho (a comprar o vender), mientras que el vendedor tiene la obligación (de vender o comprar). De este modo, la posición corta cobra una prima, que es el **precio de la opción**, como compensación por asumir la obligación de cumplir con el contrato, mientras que la posición larga paga la prima a cambio de la flexibilidad de decidir si ejercer o no la opción en el futuro. Además, el vendedor de la opción debe depositar una garantía, conocida como margen, debido a que tiene incentivos a no cumplir con el contrato cuando no le sea favorable. Esto se debe a que los emisores de opciones pueden enfrentarse a pérdidas ilimitadas, mientras que el comprador de la opción enfrenta una pérdida limitada, que está restringida al monto de la prima pagada por la opción, y por lo tanto no necesita proporcionar garantías adicionales.

$$\text{Prima} = \text{Valor Intrínseco} + \text{Valor Temporal o Extrínseco} \quad (2.1)$$

El **valor intrínseco** de una opción representa el valor que tendría la opción si se ejerciera en el momento actual, es decir, el valor real que se obtendría si se decidiera ejercer la opción inmediatamente. Mientras que, el **valor temporal** refleja el valor adicional de la opción debido a factores como el tiempo hasta su vencimiento y la volatilidad del activo subyacente. Formalmente, definimos el valor intrínseco de la siguiente manera.

Definición 2.1.2 (Valor intrínseco de una opción europea). El valor intrínseco (VI) de una opción se define como el máximo entre cero y el valor que tendría la opción si se ejerciera inmediatamente. Sea $S(t)$ el precio del activo subyacente en la fecha de vencimiento, y K el precio de ejercicio, entonces:

- El VI de una **posición larga** en una **opción de compra** europea es:

$$\max(S(t) - K, 0) \quad (2.2)$$

- El VI de una **posición corta** en una **opción de compra** europea es:

$$-\max(S(t) - K, 0) = \min(K - S(t), 0) \quad (2.3)$$

- El VI de una **posición larga** en una **opción de venta** europea es:

$$\max(K - S(t), 0) \quad (2.4)$$

- El VI de una **posición corta** en una **opción de venta** europea es:

$$-\max(K - S(t), 0) = \min(S(t) - K, 0) \quad (2.5)$$

No debemos confundir el valor intrínseco de una opción financiera con su beneficio. El valor intrínseco representa el valor actual de una opción si se ejerciera inmediatamente, mientras que el beneficio de una opción considera no solo el valor intrínseco, sino también la prima pagada o recibida.

Definición 2.1.3 (Beneficio de una opción europea). Sea $S(t)$ el precio del activo subyacente en la fecha de vencimiento, K el precio de ejercicio, c la prima de una opción de compra y v la prima de una opción de venta, entonces el beneficio en diferentes posiciones es el siguiente:

- El Beneficio de una **posición larga** en una **opción de compra** europea es:

$$B = S(t) - K - c \quad (2.6)$$

- El Beneficio de una **posición corta** en una **opción de compra** europea es:

$$B = K - S(t) + c \quad (2.7)$$

- El Beneficio de una **posición larga** en una **opción de venta** europea es:

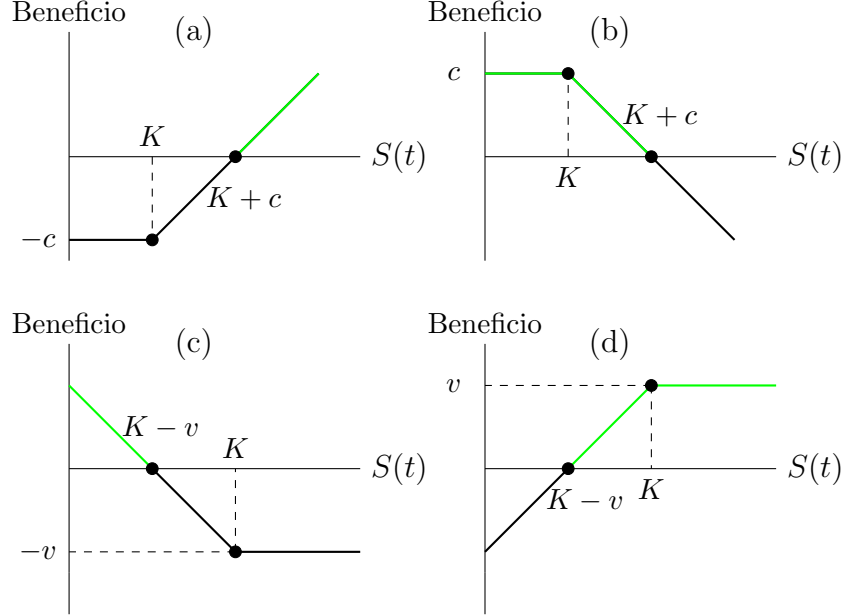
$$B = S(t) - K - v \quad (2.8)$$

- El Beneficio de una **posición corta** en una **opción de venta** europea es:

$$B = K - S(t) + v \quad (2.9)$$

A continuación, en la Figura 2.1 están representados los beneficios asociados a las distintas posiciones que se pueden tomar en el caso de opciones europeas. Los gráficos de la Figura 2.1 ilustran cómo varían los valores de las opciones de compra y de venta dependiendo del precio del activo subyacente $S(t)$ en el momento del vencimiento, tanto para posiciones largas como cortas. Además, en el gráfico (a) de la la Figura 2.1 podemos observar que el propietario de una opción de compra tiene sus pérdidas limitadas a la prima pagada, donde la función no puede descender por debajo del valor de la prima. De manera análoga, el propietario de una opción de venta se beneficia cuando el precio del activo subyacente baja, pero su riesgo ante una subida también está limitado a la prima pagada, lo cual se puede ver claramente en el gráfico (c) de la la Figura 2.1. Por otro lado, en los gráficos (b) y (d) de la la Figura 2.1, que representan las posiciones cortas de opciones de compra y de venta, respectivamente, muestran que están expuestos a pérdidas potencialmente ilimitadas, lo que pone de relieve el riesgo asociado a estas posiciones.

Figura 2.1: Beneficio de opciones europeas ante las cuatro posiciones posibles: (a) Posición larga de una opción de compra, (b) Posición corta de una opción de compra, (c) Posición larga de una opción de venta, (d) Posición corta en opción de venta. En los gráficos, S_T es el precio del activo subyacente en la fecha de vencimiento, K es el precio de ejercicio, c es la prima de una opción de compra y p es la prima de una opción de venta. Fuente: elaboración propia.



En función del valor intrínseco, las opciones pueden clasificarse de la siguiente manera.

Definición 2.1.4 (Grado del dinero o Moneyness). Las opciones se pueden clasificar en las siguientes categorías: “in the money” (ITM), “at the money” (ATM), o “out of the money” (OTM). Sea $S(t)$ el precio del activo subyacente en la fecha de vencimiento y K es el precio de ejercicio, una opción de compra está ITM cuando $S(t) > K$, ATM cuando $S(t) = K$, y OTM cuando $S(t) < K$. Por otro lado, una opción de venta está ITM cuando $S(t) < K$, ATM cuando $S(t) = K$, y OTM cuando $S(t) > K$. En resumen:

$$\text{Compra}(S(t), K) : \begin{cases} \text{ITM,} & \text{si } S(t) > K \\ \text{ATM,} & \text{si } S(t) = K \\ \text{OTM,} & \text{si } S(t) < K \end{cases} \quad \text{Venta}(S(t), K) : \begin{cases} \text{ITM,} & \text{si } S(t) < K \\ \text{ATM,} & \text{si } S(t) = K \\ \text{OTM,} & \text{si } S(t) > K \end{cases}$$

Esta terminología que relaciona el precio del activo subyacente y el precio de ejercicio de la opción es importante para entender cuando conviene ejercer una opción, lo cual ocurre cuando esté ITM, ya que en tal caso el propietario obtiene una ganancia inmediata. Estos conceptos pueden parecer irrelevantes para una opción europea, ya que solo se puede ejercer en el vencimiento, pero nos dan información sobre el valor de la opción antes de esa fecha. Aunque no se pueda ejercer, la opción puede ser vendida en cualquier momento, y la información sobre si está ITM adquiere relevancia para determinar su VI.

Por último, cabe destacar cinco factores que afectan el precio de una opción europea sobre acciones. (1) El **precio actual del activo subyacente** (S_0) afecta directamente a las opciones de compra y venta, las primeras se vuelven más valiosas cuando el precio sube y las segundas cuando baja. (2) El **precio de ejercicio** (K) aumenta el valor de las opciones de venta a medida que sube ($K - S(t)$), pero disminuye el valor de las opciones de compra ($S(t) - K$). (3) El **tiempo hasta el vencimiento** (T) puede aumentar o disminuir el valor de la opción, por lo que su efecto sobre el precio de la opción es incierto. (4) La **volatilidad** (σ), que mide la incertidumbre del movimiento del precio, incrementa el valor tanto de las opciones de compra como de las de venta, dado que aumenta la probabilidad de que el precio se mueva significativamente en una dirección favorable o desfavorable. Finalmente, (5) la **tasa de interés del activo libre de riesgo** (r) influye positivamente en el valor de las opciones de compra y negativamente en las de venta, debido a que, si sube, los pagos futuros se descuentan a una tasa más alta, ya que el dinero en el futuro tiene un valor menor hoy.

En resumen, cada uno de estos factores afecta el valor de una opción al influir en las probabilidades de que esta termine ITM en el momento de vencimiento. Su efecto sobre el precio de las opciones europeas de compra y venta se resume en el Cuadro 2.1.

Cuadro 2.1: Factores que afectan al precio de las opciones europeas de compra (call) y venta (put) sobre acciones. Los signos indican la dirección del impacto esperado: + (incrementa el valor), − (disminuye el valor) y ? (efecto ambiguo). Fuente: Hull and White [1987].

Variable	Call europea	Put europea
(1) S_0 , precio actual del activo subyacente	+	−
(2) K , precio de ejercicio	−	+
(3) T , tiempo hasta el vencimiento	?	?
(4) σ , volatilidad del precio del activo subyacente	+	+
(5) r , tasa de interés libre de riesgo	+	−

2.1.2. Modelo Black-Scholes

Un aspecto fundamental en finanzas cuantitativas radica en determinar cuál es el precio justo de una opción financiera en el momento de su venta. En 1973, Black and Scholes [1973] introdujeron la primera solución en forma cerrada para la valoración de opciones de compra y de venta europeas. Este modelo, conocido como el **modelo Black-Scholes**, permite calcular el precio teórico de una opción de manera precisa, basándose en factores como el precio actual de la acción, el precio de ejercicio de la opción, el tiempo hasta el vencimiento y la volatilidad del activo subyacente. La contribución de Black and Scholes, junto con el trabajo relacionado de Merton [1976], marcó un punto de inflexión tanto en la teoría como en la práctica en las finanzas modernas.

Black and Scholes fundamentaron su modelo en una serie de supuestos clave que permiten la aplicación de este enfoque en la valoración de opciones, estos se detallan en el Cuadro 2.2. Bajo estos supuestos, el valor de una opción dependerá únicamente del precio del activo subyacente, del tiempo, y de variables consideradas constantes. Además, Black and Scholes [1973] enuncian que es posible crear una cobertura, consistente en una posición larga en la opción y una posición corta en la opción, cuyo valor no dependerá del precio del activo subyacente, sino únicamente del tiempo y de los valores de constantes conocidas. Esta es la idea principal detrás de la ecuación de valoración de Black and Scholes [1973], la cual se apoya en la condición de que se puede cubrir perfectamente la opción, comprando y vendiendo el activo subyacente de forma que se elimine el riesgo.

Cuadro 2.2: Supuestos sobre los activos y el mercado en el modelo de Black and Scholes [1973].
Fuente: Elaboración propia.

Supuestos	
Sobre los activos	- Tasa libre de riesgo: la tasa de interés a corto plazo es conocida y constante en el tiempo.
	- Volatilidad constante: la volatilidad del activo subyacente es conocida y constante.
	- Paseo aleatorio: el precio del activo sigue un paseo aleatorio en tiempo continuo, con una tasa de varianza proporcional al cuadrado del precio del activo. La distribución de los precios futuros es lognormal, con μ conocida y constante.
	- Sin dividendos: la opción no paga dividendos durante su vida.
Sobre el mercado	- Opción europea: la opción solo puede ejercerse al vencimiento.
	- Mercado sin fricciones: no hay costes de transacción para comprar o vender el activo subyacente o la opción.
	- Posibilidad de endeudamiento: se puede pedir prestado cualquier fracción del precio de un valor de un activo a la tasa de interés a corto plazo.
	- Ausencia de arbitraje: no existen oportunidades de arbitraje, es decir, no existen discrepancias de precios entre dos o más mercados o instrumentos que puedan asegurar un beneficio sin riesgo.
	- Sin penalizaciones por ventas en corto: no hay penalizaciones por ventas en corto, y la liquidación ocurre en una fecha futura con el precio de mercado.

Sea $S(t)$ el precio del activo subyacente en el tiempo t . Consideremos una opción que otorga al titular el derecho de comprar el activo subyacente a un precio fijo K en un tiempo fijo T en el futuro y el tiempo actual es $t = 0$.

En el caso de una opción de compra europea, si en el tiempo T el precio del activo subyacente $S(T)$ excede el precio de ejercicio K , el titular ejerce la opción con un VI de $S(T) - K$. Por otro lado, si $S(T) \leq K$, entonces la opción caduca sin valor. De modo que, en este caso, el pago al titular de la opción en el tiempo T es:

$$(S(T) - K)^+ = \text{máx}\{0, S(T) - K\} \quad (2.10)$$

Por el contrario, en una opción de venta europea, si en el momento T el precio del activo subyacente $S(T)$ es inferior al precio de ejercicio K , el titular ejerce la opción, obteniendo un valor de $K - S(T)$. En caso contrario la opción expira sin valor. Así, el pago al titular de la opción en el tiempo T es:

$$(S(T) - K)^- = \text{máx}\{0, K - S(T)\} \quad (2.11)$$

El valor presente de este pago lo obtenemos si multiplicamos por un **factor de descuento** e^{-rT} , con r como la tasa de interés compuesto continuo ¹. De forma que, el valor presente esperado viene dado por $E[e^{-rT}(S(T) - K)^+]$ y $E[e^{-rT}(S(T) - K)^-]$. Para poder conocer el valor de estas expresiones, debemos especificar la distribución de la variable aleatoria $S(T)$.

El modelo de Black and Scholes describe la evolución del precio del activo subyacente, $S(T)$, a través de la Ecuación Diferencial Estocástica (EDE) ²:

$$\frac{dS(t)}{S(t)} = r dt + \sigma dW(t) \quad (2.12)$$

donde $W(t)$ es un movimiento Browniano estándar ³. La ecuación 2.12 puede interpretarse como la modelización de los cambios porcentuales $\frac{dS}{S}$ en el precio del activo subyacente como los incrementos de un movimiento Browniano. En otras palabras, en el contexto del movimiento browniano los incrementos corresponden a variaciones aleatorias que ocurren a lo largo del tiempo, por tanto estamos modelando las fluctuaciones del precio del activo subyacente mediante este proceso. En la ecuación 2.12, el parámetro σ es la volatilidad del precio del activo subyacente y el coeficiente de dt es la tasa media de retorno.

¹Cuando los intereses se componen una vez al año, el valor final de una inversión V después de n años con una tasa de interés r es $V(1 + R)^n$. Cuando los intereses se componen m veces al año, entonces $V(1 + \frac{R}{m})^{mn}$. Cuando m tiende a infinito:

$$\lim_{m \rightarrow \infty} \left(1 + \frac{r}{m}\right)^{mn} = e^{rn}$$

²Ver apéndice para la definición de EDE.

³Ver apéndice para la definición de movimiento browniano estándar.

La solución de la EDE (2.12) es la siguiente:

$$S(T) = S(0)e^{\left(r - \frac{\sigma^2}{2}\right)T + \sigma W(T)} \quad (2.13)$$

Dado que $S(0)$ es el precio actual del activo subyacente, podemos suponer que es conocido. Además, como la variable aleatoria $W(T)$ está distribuida normalmente con media 0 y varianza T , entonces es también la distribución de $\sqrt{T}Z$ si Z es una variable aleatoria normal estándar. En consecuencia, podemos representar el precio del activo subyacente al vencimiento de la siguiente manera:

$$S(T) = S(0)e^{\left(r - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{T}Z} \quad (2.14)$$

De esta expresión, se deduce que el logaritmo del precio del activo subyacente se distribuye como una normal y el precio del activo subyacente en sí sigue una **distribución lognormal**, debido a que $S(T)$ es igual al exponente de una variable normal.

De todo lo anterior, llegamos a que las expectativas $E[e^{-rT}(S(T) - K)^+]$ y $E[e^{-rT}(S(T) - K)^-]$ son integrales con respecto a la densidad lognormal de $S(T)$. De manera que, evaluando esta integral en términos de la función de distribución acumulada normal estándar, obtenemos las fórmulas de Black and Scholes para los precios de las opciones europeas de compra, $V_c(S, t)$, y venta, $V_p(S, t)$:

$$V_c(S, t) = S_0N(d_1) - Ke^{-rt}N(d_2) \quad (2.15)$$

$$V_p(S, t) = Ke^{-rt}N(-d_2) - S_0N(-d_1) \quad (2.16)$$

con

$$d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{t}} \quad (2.17)$$

$$d_2 = \frac{\ln(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{t} \quad (2.18)$$

donde S_0 es el precio del activo subyacente en el momento inicial, K es el precio de ejercicio de la opción, r es la tasa libre de riesgo continua, σ representa la volatilidad del precio del activo subyacente, t es el tiempo hasta el vencimiento de la opción y $N(x)$ es la función de distribución acumulada de la normal estándar.

El modelo clásico de Black and Scholes tiene algunas limitaciones, ya que los supuestos son poco realistas al simplificar las condiciones del mercado. Una restricción significativa es que, en realidad, los precios no siguen una distribución lognormal de forma estricta, ni se conoce la tasa de interés libre de riesgo y, al igual que la volatilidad, se presupone constante en el modelo, lo que no refleja completamente la dinámica real del mercado.

2.1.3. Las griegas

En este contexto, las griegas surgen como herramientas importantes para analizar el comportamiento de las opciones y la gestión de riesgos, ya que cada una de ellas mide la sensibilidad del precio de una opción ante un pequeño cambio en ciertos parámetros de los que depende. Las griegas más comunes son las derivadas de primer orden del valor de una opción: Delta (Δ), Vega (\mathcal{V}), Theta (Θ) y Rho (ρ), así como Gamma (Γ), que es una derivada de segundo orden del valor de una opción. La idea es que las griegas sean cercanas a cero, lo que indica que el valor de la opción no se ve fuertemente alterado por cambios en los parámetros de los que se derivan las griegas. A continuación, definimos cada una de las griegas que buscamos estimar en este trabajo.

Definición 2.1.5 (Delta). El **Delta** (Δ) de una opción se define como la tasa de cambio del precio de la opción con respecto al **precio del activo subyacente**. En otras palabras, el Delta es la pendiente de la curva que relaciona el precio de la opción con el precio del activo subyacente y se expresa como:

$$\Delta = \frac{\partial V(S, t)}{\partial S} \quad \text{tal que} \quad \Delta \in [-1, 1] \quad (2.19)$$

Definición 2.1.6 (Theta). El **Theta** (Θ) de una opción es la tasa de cambio del valor de la opción al paso del **tiempo**, manteniéndose todo lo demás constante. A veces, el Theta se conoce como la “decaída temporal”, ya que, a medida que el tiempo avanza y se acerca la fecha de vencimiento, el valor de una opción disminuye, lo que explica que la derivada sea negativa. Su expresión es la siguiente:

$$\Theta = -\frac{\partial V(S, t)}{\partial t} \quad (2.20)$$

Definición 2.1.7 (Gamma). La **Gamma** (Γ) de una opción sobre un activo subyacente es la **tasa de cambio del Delta** de la opción con respecto al precio del activo subyacente, y se expresa como:

$$\Gamma = \frac{\partial \Theta}{\partial S} = \frac{\partial^2 V(S, t)}{\partial S^2} \quad \text{tal que} \quad \Delta \in [0, 1] \quad (2.21)$$

Definición 2.1.8 (Vega). La **Vega** (\mathcal{V}) de una opción es la tasa de cambio del valor de la opción con respecto a la **volatilidad del activo subyacente**. La expresión matemática de Vega es:

$$\mathcal{V} = \frac{\partial V(S, t)}{\partial \sigma} \quad (2.22)$$

Definición 2.1.9 (Rho). El **Rho** (ρ) de una opción es la tasa de cambio del valor de la opción con respecto a la **tasa de interés**. Su expresión es la siguiente:

$$\rho = \frac{\partial V(S, t)}{\partial r} \quad (2.23)$$

En el Cuadro 2.3 se presenta un resumen de las griegas de una opción, incluyendo las expresiones y las descripciones de cómo cada una mide la sensibilidad del valor de la opción frente a cambios en los parámetros correspondientes.

Cuadro 2.3: Expresiones y definiciones de las principales griegas asociadas al precio de opciones financieras. Fuente: Elaboración propia.

Griega	Expresión	Descripción
Delta (Δ_t)	$\frac{\partial V(S,t)}{\partial S}$	Mide la sensibilidad del precio de la opción al precio del activo subyacente.
Gamma (Γ_t)	$\frac{\partial^2 V(S,t)}{\partial S^2}$	Mide la sensibilidad de Delta respecto al precio del activo subyacente.
Theta (Θ_t)	$-\frac{\partial V(S,t)}{\partial t}$	Mide la sensibilidad del precio de la opción al tiempo restante hasta el vencimiento.
Vega (ν_t)	$\frac{\partial V(S,t)}{\partial \sigma}$	Mide la sensibilidad del precio de la opción a la volatilidad.
Rho (ρ_t)	$\frac{\partial V(S,t)}{\partial r}$	Mide la sensibilidad del precio de la opción a la tasa de interés.

A continuación, a partir de los resultados obtenidos del modelo de Black and Scholes para calcular el valor de una opción, dados por las ecuaciones 2.15 y 2.16, podemos obtener de forma directa las expresiones de las griegas. En el Cuadro 2.4 se presentan estas expresiones para opciones europeas de compra y venta.

Cuadro 2.4: Este cuadro presenta las expresiones de las griegas asociadas a opciones europeas de compra y venta bajo el modelo de Black and Scholes [1973]. En las fórmulas, S_0 es el precio del activo subyacente, K es el precio de ejercicio, r es la tasa libre de riesgo, T es el tiempo hasta el vencimiento y σ es la volatilidad. Además, $N(x)$ es la función de distribución acumulada de la normal estándar y $N'(x)$ es su función de densidad. Fuente: Hull and White [1987].

Griega	Opción de compra	Opción de venta
Delta (Δ)	$N(d_1)$	$[N(d_1) - 1]$
Gamma (Γ)	$\frac{N'(d_1)}{S_0 \sigma \sqrt{T}}$	
Theta (Θ)	$\frac{S_0 N'(d_1) \sigma}{2\sqrt{T}} - r K e^{-rT} N(d_2)$	$-\frac{S_0 N'(d_1) \sigma}{2\sqrt{T}} + r K e^{-rT} N(-d_2)$
Vega (ν)	$S_0 \sqrt{T} N'(d_1)$	
Rho (ρ)	$K T e^{-rT} N(d_2)$	$-K T e^{-rT} N(-d_2)$

En resumen, las griegas son medidas que describen la sensibilidad de los precios de las opciones ante diversos factores, que no solo se utilizan para analizar el comportamiento de las opciones, sino que también para la gestión del riesgo mediante la **cobertura de posiciones**. De modo que, veamos cómo se emplean en este contexto.

Hull and White explican que las instituciones financieras ofrecen productos basados en opciones financieras, cuyo activo subyacente no se negocia en las bolsas organizadas, lo que hace que sea complicado cubrir las posiciones, siendo este el escenario en el que las griegas juegan un papel fundamental.

Una estrategia de cobertura conocida es el **Delta hedging**, que consiste en crear una posición cuyo Delta sea igual a cero, es decir, una posición que no tenga exposición al movimiento del precio del activo subyacente. De modo que, como el Delta de una opción varía con el tiempo, la posición en el activo subyacente debe ajustarse frecuentemente.

Una vez que se ha logrado neutralizar el Delta de una opción, se analiza el Gamma, que mide la tasa de cambio de su Delta con respecto al precio del activo subyacente. Una posición **Gamma-neutral** permite mitigar este efecto, y generalmente consiste en comprar o vender una opción con un Gamma que compense el de la posición original.

La cobertura de Delta y Gamma se basa en la suposición de que la volatilidad del activo subyacente es constante, pero en la práctica esta cambia con el tiempo. De modo que, para cubrir una opción frente a cambios en la volatilidad podemos tomar una posición **Vega-neutral**, lo cual se logra tomando una posición contraria. Es decir, si la opción inicial tiene una Vega positiva (se beneficia de un aumento en la volatilidad), se toma una posición con Vega negativa para contrarrestar el efecto de cambios en la volatilidad.

Finalmente, una estrategia alternativa de cobertura consiste en crear una **opción sintética**. Esta estrategia busca replicar el comportamiento de una opción utilizando combinaciones del activo subyacente y diferentes opciones. Una opción sintética de compra se construye comprando el activo correspondiente y una opción de venta con el mismo precio de ejercicio y vencimiento, mientras que una de venta se forma vendiendo el activo y comprando una opción de compra equivalente.

2.2. Fundamentos de Monte Carlo

Los métodos de simulación Monte Carlo son muy útiles para abordar el problema de valoración de derivados cuando no es posible resolver de forma analítica las Ecuaciones Diferenciales Estocásticas de los modelos clásicos. Al realizar simulaciones de múltiples trayectorias de los precios, se obtiene una aproximación del precio de los derivados financieros, en nuestro caso de las opciones financieras.

En la teoría de valoración de derivados, se tiene que, bajo ciertas condiciones, el precio de un instrumento derivado puede expresarse como un valor esperado. En consecuencia, la valoración de derivados se reduce al cálculo de expectativas, que podemos expresar como una integral, la cual puede tener una dimensión elevada. En este contexto, los métodos de simulación de Monte Carlo resultan especialmente útiles.

En esta sección, discutimos los principales algoritmos de simulación de Monte Carlo para generar variables aleatorias, centrándonos en el caso de la **distribución normal estándar**, ya que es la que necesitamos para simular trayectorias de precios de activos financieros mediante el **Movimiento Browniano Geométrico** (GBM, en sus siglas en inglés). Esto se debe a que los precios de los activos siguen una distribución lognormal debido a la naturaleza exponencial del GBM. De modo que, profundizamos en los métodos para simular trayectorias del movimiento browniano en una y múltiples dimensiones, así como la extensión de estos métodos al GBM.

Además, estudiamos diferentes métodos para generar variables aleatorias con distribución normal, como el algoritmo propuesto por Box and Muller [1958], Marsaglia and Bray [1964], y el algoritmo Ziggurat de Marsaglia and Tsang [2000], siendo este último el implementado en *Python* mediante la librería *numpy*. Todo el contenido de este apartado se basa en el manual de Glasserman [2010], en el cual se presentan los conceptos acerca de la simulación Monte Carlo para poder entender los métodos de estimación de las griegas.

2.2.1. Simulación Monte Carlo

En este apartado presentamos nuestro Algoritmo de simulación de las trayectorias de los precios de los activos subyacentes asociados a una opción financiera, los cuales siguen un GBM y en consecuencia, obtener el valor de dichas opciones. Es conveniente definir una serie de conceptos que resultan fundamentales para comprender el proceso de simulación basado en los modelos estocásticos que describen el comportamiento de los activos subyacentes. La base teórica empleada acerca del cálculo estocástico en este punto está definida en detalle en el Apéndice A.

- Movimiento Browniano

La simulación del movimiento browniano requiere generar los valores $(W(t_1), \dots, W(t_n))$ o $(X(t_1), \dots, X(t_n))$ en un conjunto fijo de puntos $0 \leq t_1 < \dots < t_n$. Dado que el movimiento browniano tiene incrementos independientes e idénticamente distribuidos normalmente, simular los valores $W(t_i)$ o $X(t_i)$ a partir de sus incrementos es directo.

Sean Z_1, \dots, Z_n variables aleatorias normales estándar independientes e idénticamente distribuidas. Supongamos que tenemos un movimiento browniano estándar con $W(0) = 0$, entonces los valores posteriores se pueden generar de la siguiente manera. Para $X \sim BM(\mu, \sigma^2)$ con μ constante y σ , y dado que $X(0) = 0$, tenemos:

$$X(t_{i+1}) = X(t_i) + \mu(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}Z_{i+1}, \quad i = 0, \dots, n-1 \quad (2.24)$$

$$W(t_{i+1}) = W(t_i) + \sqrt{t_{i+1} - t_i}Z_{i+1}, \quad i = 0, \dots, n-1 \quad (2.25)$$

El vector $(W(t_1), \dots, W(t_n))$, que representa los valores del movimiento browniano en distintos instantes de tiempo, es una transformación lineal del vector de incrementos $(W(t_1), W(t_2) - W(t_1), \dots, W(t_n) - W(t_{n-1}))$. Dado que estos incrementos son independientes y siguen una distribuidos normal, entonces $(W(t_1), \dots, W(t_n))$ sigue una distribución normal multivariada. De esta forma, simular dicho vector se basa en métodos de generación de **vectores normales multivariados**.

En la práctica generaremos los incrementos paso a paso, obteniendo en cada instante vectores normales unidimensionales. Para ello, utilizaremos la expresión equivalente a la ecuación 2.25:

$$dW_{i+1} = \sqrt{dt_{i+1}} \cdot Z_{i+1}, \quad Z_{i+1} \sim N(0, 1) \quad (2.26)$$

donde dW_{i+1} representa el incremento del movimiento browniano o incremento de Wiener en el intervalo $[t_i, t_{i+1}]$, y Z_{i+1} es una variable aleatoria normal estándar independiente que que simularemos en cada paso temporal con la librería *Numpy*.

- Movimiento Browniano Geométrico

Un proceso estocástico $S(t)$ es un movimiento browniano geométrico (GBM) si $\log S(t)$ es un movimiento browniano con valor inicial $\log S(0)$, es decir, un GBM es simplemente un movimiento browniano estándar elevado a la exponencial. En consecuencia, todos los métodos para simular el movimiento browniano se convierten en métodos para simular el movimiento browniano geométrico mediante exponenciación.

En general, si $u < t$, entonces se tiene el siguiente resultado⁴:

$$S(t) = S(u) \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) (t - u) + \sigma(W(t) - W(u)) \right) \quad (2.27)$$

Dado que los incrementos de W son independientes y distribuidos normalmente, esto proporciona un procedimiento recursivo simple para simular los valores de S en los instantes $0 = t_0 < t_1 < \dots < t_n$ para $i = 0, 1, \dots, n - 1$:

$$S(t_{i+1}) = S(t_i) \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) (t_{i+1} - t_i) + \sigma(W(t_{i+1}) - W(t_i)) \right) \quad (2.28)$$

donde Z_1, Z_2, \dots, Z_n son normales estándar independientes. Este método es una fórmula exacta para el GBM, ya que las trayectorias $(S(t_0), \dots, S(t_n))$ generadas tienen la distribución conjunta del proceso $S \sim \text{GBM}(\mu, \sigma^2)$ en los instantes t_1, \dots, t_n .

La expresión 2.29 puede reescribirse de forma equivalente como:

$$S(t_{i+1}) = S(t_i) \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) dt_{i+1} + \sigma dW_{i+1} \right) \quad (2.29)$$

donde $dW_{i+1} = \sqrt{dt_{i+1}} \cdot Z_{i+1}$.

Esta formulación permite simular el proceso paso a paso, generando en cada intervalo un incremento aleatorio dW a partir de una variable normal estándar, obteniendo así distintas trayectorias del precio del activo subyacente, que nos permitirán obtener el precio de la opción asociada a dicho activo subyacente con la ayuda del método Monte Carlo.

- Distribución Lognormal

El GBM está definido por la siguiente expresión⁵:

$$S(t) = S(0) \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W(t) \right) \quad (2.30)$$

donde $S(t)$ es el precio del activo subyacente en el tiempo t , $S(0)$ es el precio inicial del activo subyacente, μ es la tasa de crecimiento esperada (media), σ es la volatilidad del activo subyacente y $W(t)$ es un proceso de Wiener o Browniano estándar.

En la ecuación 2.30, observamos que si $S(t) \sim \text{GBM}(\mu, \sigma^2)$, entonces la distribución de $S(t)$ es la de la exponencial de una variable aleatoria normal, que se denomina **distribución lognormal**. Esta distribución se expresa como, $S(t) \sim \text{LN}(\mu, \sigma^2)$ si la variable aleatoria $S(t)$ tiene la distribución de $\exp(\mu + \sigma Z)$, donde $Z \sim N(0, 1)$.

⁴Ver apéndice para más detalles.

⁵Ver apéndice para más detalles.

En consecuencia, si $Y = S(t)$, entonces su distribución está dada por:

$$P(Y \leq y) = P\left(Z \leq \frac{\log(y) - \mu}{\sigma}\right), \quad (2.31)$$

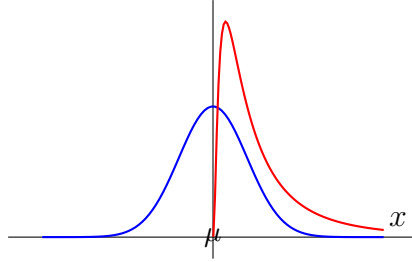
y su densidad es

$$f_Y(y) = \frac{1}{y\sigma\sqrt{2\pi}} \exp\left(-\frac{(\log(y) - \mu)^2}{2\sigma^2}\right) \quad y > 0 \quad (2.32)$$

En la Figura 2.2, se presenta una comparación visual entre la distribución normal y la lognormal, representando sus respectivas funciones de densidad. Por un lado, en la Figura 2.2 observamos que la distribución normal es simétrica, mientras que la lognormal es asimétrica, con una cola que se extiende hacia la derecha, lo que refleja la mayor probabilidad de obtener valores grandes.

Por otro lado, en la Figura 2.2 la distribución lognormal está definida solo para valores positivos y, a diferencia de la normal, no tiene la misma media, mediana y moda. Esto se debe a que la distribución normal es simétrica, por lo que la media, la mediana y la moda coinciden en el mismo valor, pero la distribución lognormal, al no ser simétrica, tiene una media ($\exp\left(\mu + \frac{\sigma^2}{2}\right)$) mayor que su mediana ($\exp(\mu)$) y una moda ($\exp(\mu - \sigma^2)$) menor que la media.

Figura 2.2: Comparación entre las distribuciones normal (azul) y lognormal (rojo), mostrando sus respectivas densidades de probabilidad. Fuente: Elaboración propia.



Los momentos de una variable aleatoria lognormal se pueden calcular utilizando la función generadora de momentos de una normal estándar $E[e^{aZ}] = e^{a^2/2}$. En consecuencia, para una variable aleatoria $Y \sim \text{LN}(\mu, \sigma^2)$, se tiene que los momentos son los siguientes:

$$E[Y] = e^{\mu + \frac{\sigma^2}{2}} \quad \text{y} \quad \text{Var}[Y] = e^{2\mu + \sigma^2} (e^{\sigma^2} - 1) \quad (2.33)$$

De esta forma, dado que $S(t) \sim \text{GBM}(\mu, \sigma^2)$, entonces de la ecuación 2.30, deducimos que la variable $S(t)$ sigue una distribución lognormal con parámetros específicos:

$$\frac{S(t)}{S(0)} \sim \text{LN}\left(\left(\mu - \frac{\sigma^2}{2}\right)t, \sigma^2 t\right) \quad (2.34)$$

Esto implica que para $S(t)$ los momentos vienen dados por:

$$E[S(t)] = e^{\mu t} S(0) \quad \text{y} \quad \text{Var}[S(t)] = e^{2\mu t} S^2(0) (e^{\sigma^2 t} - 1) \quad (2.35)$$

- Simulación Monte Carlo del precio de una opción

Una vez que hemos definido los conceptos clave, podemos proceder a explicar el algoritmo utilizado para la simulación de las trayectorias de precios y la estimación del valor de la opción que empleamos en este trabajo. Este algoritmo emplea la técnica de Monte Carlo para generar las trayectorias del precio del activo subyacente correspondiente y, posteriormente, calcular el valor estimado de la opción europea en función de las variables de entrada. Los pasos detallados se presentan en el Algoritmo 1.

Algoritmo 1: Simulación Monte Carlo del precio de una opción europea.

Datos: S_0 (precio inicial), K (precio de ejercicio), r (tasa libre de riesgo), σ (volatilidad), T (tiempo hasta vencimiento), N (número de trayectorias), pasos de tiempo.

Resultado: Precio estimado de la opción europea.

Algoritmo:

1. Generar $Z \sim N(0, 1)$, una variable aleatoria normal estándar para cada incremento temporal y cada trayectoria.
2. Calcular los incrementos $dW_t = \sqrt{\Delta t} \cdot Z$, donde $\Delta t = T/\text{pasos}$.
3. Sustituir los incrementos en la fórmula del GBM para simular el precio subyacente S_T :

$$S(T) = S(0) \cdot \exp \left(\left(r - \frac{\sigma^2}{2} \right) T + \sigma \cdot W_t \right)$$

4. Evaluar el VI de la opción:

$$\text{Opción de compra: } \max(S_T - K, 0), \quad \text{Opción de venta: } \max(K - S_T, 0)$$

5. Descontar los beneficios a valor presente:

$$\text{Beneficio descontado} = e^{-rT} \cdot \text{beneficio}$$

6. Monte Carlo: promediar los beneficios descontados sobre todas las trayectorias:

$$\text{Precio de la opción} = \frac{1}{N} \sum_{i=1}^N \text{beneficio descontado}$$

2.2.2. Generación de variables aleatorias

En este apartado presentamos los métodos de generación de variables aleatorias de la distribución normal estándar $N(0,1)$, fundamentales para simular las trayectorias del GBM, el cual modela el comportamiento del precio de los activos subyacentes durante la vida de una opción.

El método **Ziggurat** es el que implementamos mediante la librería *NumPy* de *Python*. Este algoritmo se basa en contribuciones previas como el método de Box and Muller [1958] y su mejora propuesta por Marsaglia and Bray [1964]. Por ello, antes de introducir el Ziggurat, presentamos brevemente estos enfoques para contextualizar su evolución.

- Método Box-Muller

El método propuesto por Box and Muller es un algoritmo que genera una muestra de una distribución normal bidimensional estándar, donde cada componente es una variable aleatoria normal unidimensional estándar, apoyándose en el siguiente Teorema.

Teorema 2.2.1. *Sean U_1, U_2 variables aleatorias independientes e idénticamente distribuidas $U(0,1)$. Entonces, las variables aleatorias definidas mediante las transformaciones:*

$$X = \sqrt{-2 \log(U_1)} \cos(2\pi U_2) \quad (2.36)$$

$$Y = \sqrt{-2 \log(U_1)} \sin(2\pi U_2) \quad (2.37)$$

son independientes e idénticamente distribuidas $N(0,1)$.

El algoritmo se basa en las siguientes dos propiedades de la distribución normal bidimensional estándar $Z \sim N(0, I_2)$:

1. $R = X^2 + Y^2$ sigue una distribución exponencial con media 2, es decir:

$$P(R \leq x) = 1 - e^{-x/2} \quad \forall x > 0 \quad (2.38)$$

2. Dado R , el punto (X, Y) está distribuido uniformemente sobre el círculo de radio \sqrt{R} centrado en el origen.

De modo que, para generar (X, Y) , primero generamos R y luego seleccionamos un punto uniformemente en el círculo de radio \sqrt{R} . Para generar R , tenemos que generar una muestra de la distribución exponencial, que se obtiene mediante:

$$R = -2 \log(U_1), \quad \text{donde } U_1 \sim \text{Unif}[0, 1] \quad (2.39)$$

Después, para seleccionar un punto aleatorio en el círculo, se genera un ángulo uniforme entre 0 y 2π , que se puede calcular como:

$$\Theta = 2\pi U_2, \quad U_2 \sim U(0, 1). \quad (2.40)$$

Por último, las coordenadas correspondientes del punto en el círculo son:

$$\left(\sqrt{R} \cos(\Theta), \sqrt{R} \sin(\Theta)\right) = (X, Y) \quad (2.41)$$

En la Figura 2.3, se muestra la representación de un punto en coordenadas polares empleadas en el método de Box and Muller, que transforma variables uniformes en variables normales estándar con un cambio a estas coordenadas.

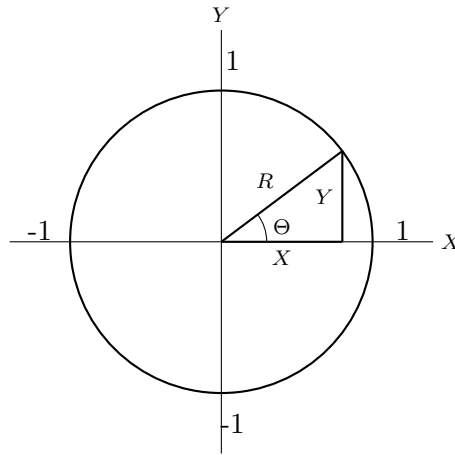


Figura 2.3: Representación de un punto en coordenadas polares dentro del círculo unitario. Fuente: Elaboración propia.

El algoritmo completo del método Box-Muller para generar variables aleatorias de una distribución normal estándar está detallado en el Algoritmo 2.

Algoritmo 2: Algoritmo de Box-Muller para generar variables $N(0,1)$.

Datos: Dos variables uniformes independientes e idénticamente distribuidas

$$U_1, U_2 \sim U(0, 1)$$

Resultado: $X, Y \sim N(0, 1)$

1. Generar $R = -2\log(U_1)$.
 2. Generar $\Theta = 2\pi U_2$.
 3. Calcular $X = \sqrt{R} \cos(\Theta)$.
 4. Calcular $Y = \sqrt{R} \sin(\Theta)$.
 5. Obtener X, Y .
-

- Método Marsaglia

El algoritmo de Marsaglia and Bray es una modificación del método de Box and Muller que mejora la eficiencia evitando las funciones seno y coseno. Marsaglia and Bray utilizan el enfoque de aceptación-rechazo para generar puntos uniformemente distribuidos en un disco unitario y luego los transforma en variables normales. Este método se resume en el Algoritmo 3.

Algoritmo 3: Algoritmo Marsaglia-Bray para generar variables $N(0,1)$.

Datos: $U_1, U_2 \sim U[0, 1]$

Resultado: $X, Y \sim N(0, 1)$

Si $X > 1$ Generar $U_1, U_2 \sim U[0, 1]$

$U_1 \leftarrow 2U_1 - 1$

$U_2 \leftarrow 2U_2 - 1$

$R \leftarrow U_1^2 + U_2^2$

Calcular $V = \sqrt{-2\log(R)/R}$

Calcular $X = U_1V$

Calcular $Y = U_2V$

Obtener X, Y

- Método Zigurat

Marsaglia and Tsang [2000] propusieron un método para generar variables aleatorias, alternativo y más eficiente al anterior, conocido como el método Zigurat. Supongamos que C es el conjunto de puntos (x, y) localizados bajo la curva definida como $y = f(x)$, con un área finita, y definamos Z como un conjunto que contiene a C . El método de Marsaglia and Tsang, consiste en generar los puntos (x, y) de manera uniforme dentro del conjunto Z hasta que uno de ellos esté dentro de C . Cuando esto ocurre, el valor de x es aceptado como una muestra válida. La probabilidad de aceptar un valor x tiene una densidad $c \cdot f(x)$, siendo c una constante que normaliza la densidad $f(x)$.

Este es el método que utilizaremos en este trabajo, el cual se emplea comúnmente en Python mediante la librería *Numpy*. No obstante, no entramos en más detalle, ya que la base teórica se encuentra fuera de los objetivos de este trabajo.

Capítulo 3

Metodología

En este capítulo, presentamos los métodos de estimación de sensibilidades de expectativas, es decir, las derivadas de las expectativas. El objetivo es estimar las griegas presentadas previamente, que son precisamente las derivadas de los precios de las opciones respecto a diferentes parámetros. El contenido de este capítulo se apoya en los métodos de estimación de las griegas presentados en el manual de Glasserman [2010].

Glasserman [2010] utiliza dos enfoques principales para estimar estas sensibilidades: (1) **método de diferencias finitas**, que requiere realizar simulaciones para dos o más valores del parámetro de interés, y (2) los métodos que no necesitan simulaciones adicionales, como el **método de la trayectoria** y el **método de la razón de verosimilitud**. Por un lado, aunque el método de diferencias finitas es sencillo de implementar, presenta el inconveniente de introducir un **sesgo** en las estimaciones, es decir, la esperanza matemática del estimador no coincide con el valor del parámetro que se desea estimar. Por otro lado, los métodos de la segunda categoría permiten obtener estimaciones *insesgadas*, ya que utilizan la simulación del proceso estocástico para calcular derivadas de forma exacta, ya sea diferenciando cada trayectoria simulada (método de la trayectoria) o la densidad de probabilidad asociada (método de la razón de verosimilitud).

Por último, cabe mencionar que además de presentar las bases teóricas y las estimaciones generales propuestas por estos métodos, también deducimos las estimaciones para cada una de las griegas: Delta (Δ), Gamma (Γ), Vega (ν), Theta (Θ) y Rho (ρ).

3.1. Método de las diferencias finitas

Los métodos de diferencias finitas son técnicas numéricas utilizadas para resolver ecuaciones diferenciales, en las cuales las derivadas se estiman o se aproximan mediante diferencias finitas, en nuestro caso las griegas.

Consideremos un modelo que depende de un parámetro $\Theta \in \mathbb{R}$ y supongamos que, para cada valor de Θ , contamos con un método para generar una variable aleatoria $Y(\Theta)$, la cual representa el resultado del modelo para el valor específico de Θ . En el contexto de simulaciones, $Y(\Theta)$ representa el resultado de un modelo estocástico cuando se establece el parámetro Θ . Este modelo genera un conjunto de variables aleatorias $Y(\Theta)$ para diferentes valores de Θ , las cuales dependen de un proceso aleatorio.

La derivada de $Y(\Theta)$, se interpreta como la derivada del paseo aleatorio de la simulación, manteniendo constantes los números aleatorios utilizados. Aunque se supone que la derivada existe en la mayoría de los casos, hay situaciones excepcionales donde esta derivada puede no estar definida, dependiendo del valor específico de Θ .

Definamos la esperanza de la variable aleatoria $Y(\Theta)$ de la siguiente manera:

$$\alpha(\Theta) = \mathbb{E}[Y(\Theta)] \quad (3.1)$$

El objetivo de este método es estimar la derivada de $\alpha(\Theta)$ respecto al parámetro Θ , es decir, $\alpha'(\Theta) = \frac{d}{d\Theta} \mathbb{E}[Y(\Theta)]$.

En la aplicación a la teoría de valoración de opciones, $Y(\Theta)$ es el **beneficio descontado de una opción**, $\alpha(\Theta)$ es el **precio de la opción** y Θ podría ser cualquier **parámetro** del modelo o del mercado que influye en el precio. En nuestro problema de estimación de las sensibilidades, dependiendo del parámetro Θ que seleccionemos, obtendremos una estimación diferente para cada griega. Por ejemplo, cuando Θ es el precio inicial del activo subyacente, entonces $\alpha'(\Theta)$ es la Delta de la opción respecto a ese activo.

A continuación, describimos dos estimadores que podemos utilizar para calcular las derivadas de $\alpha(\Theta)$ con este enfoque. Si bien existen tres estimadores en este método, el de *diferencia hacia adelante*, el de *diferencia centrada* y el de *diferencia hacia atrás*, nosotros nos centraremos en los dos primeros. Esto se debe a que nuestro objetivo principal es emplear el estimador de *diferencia central*, y el de *diferencia hacia adelante* será útil para justificar dicha elección, pero con el de *diferencia hacia atrás* el razonamiento es análogo.

- Estimador de diferencia hacia adelante

El estimador de *diferencia hacia adelante*, permite aproximar la derivada de $\alpha(\Theta)$ utilizando simulaciones en dos puntos cercanos, Θ y $\Theta + h$, donde $h > 0$ es un valor pequeño. Para construir este estimador, primero se generan n simulaciones independientes del modelo en el parámetro Θ , denotadas como $Y_1(\Theta), \dots, Y_n(\Theta)$. De manera similar, se generan otras n simulaciones en el parámetro $\Theta + h$, denotadas como $Y_1(\Theta + h), \dots, Y_n(\Theta + h)$. Después, a partir de estas simulaciones, se calculan los promedios correspondientes de cada conjunto de replicaciones de la siguiente manera:

$$\bar{Y}_n(\Theta) = \frac{1}{n} \sum_{i=1}^n Y_i(\Theta) \quad \text{y} \quad \bar{Y}_n(\Theta + h) = \frac{1}{n} \sum_{i=1}^n Y_i(\Theta + h)$$

El estimador de *diferencia hacia adelante* se define entonces como:

$$\hat{\Delta}_F = \hat{\Delta}_F(n, h) = \frac{\bar{Y}_n(\Theta + h) - \bar{Y}_n(\Theta)}{h} \quad (3.2)$$

Si la función α es dos veces diferenciable en Θ , entonces tenemos que la serie de Taylor para $\alpha(\Theta + h)$ viene dada por:

$$\alpha(\Theta + h) = \alpha(\Theta) + \alpha'(\Theta)h + \frac{1}{2}\alpha''(\Theta)h^2 + o(h^2) \quad (3.3)$$

Sustituyendo la serie de Taylor 3.3 en la expresión 3.2, tenemos que la **esperanza** del estimador de *diferencia hacia adelante* es:

$$\mathbb{E}[\hat{\Delta}_F] = \frac{\alpha(\Theta + h) - \alpha(\Theta)}{h} = \alpha'(\Theta) + \frac{1}{2}\alpha''(\Theta)h + o(h) \quad (3.4)$$

En consecuencia, el **sesgo** del estimador de *diferencia hacia adelante* es el siguiente:

$$\text{Sesgo}(\hat{\Delta}_F) = \mathbb{E}[\hat{\Delta}_F - \alpha'(\Theta)] = \frac{1}{2}\alpha''(\Theta)h + o(h) \quad (3.5)$$

- Estimador de diferencia central

Una alternativa al estimador de diferencia hacia adelante es el estimador de *diferencia central*. En este caso, se evalúa la función en dos puntos simétricos alrededor de Θ , que son $\Theta - h$ y $\Theta + h$, donde $h > 0$ representa un pequeño incremento. Para aplicar este método, se realizan n simulaciones en dos puntos: $\Theta - h$ y $\Theta + h$, obteniendo dos conjuntos de replicaciones independientes del modelo: $Y_1(\Theta - h), \dots, Y_n(\Theta - h)$ y $Y_1(\Theta + h), \dots, Y_n(\Theta + h)$, para algún valor de $h > 0$. A continuación, se calcula el promedio de cada conjunto de replicaciones de forma análoga a como se hizo en el caso del estimador anterior. De modo que, obtenemos el estimador de *diferencia central*:

$$\hat{\Delta}_C = \hat{\Delta}_C(n, h) = \frac{\bar{Y}_n(\Theta + h) - \bar{Y}_n(\Theta - h)}{2h} \quad (3.6)$$

Si la función α es al menos dos veces diferenciable en Θ , entonces se pueden obtener las expansiones en serie de Taylor alrededor de ese punto. La serie de Taylor para $\alpha(\Theta + h)$ ya fue presentada en la ecuación 3.3 y, de forma análoga, para $\alpha(\Theta - h)$ se tiene:

$$\alpha(\Theta - h) = \alpha(\Theta) - \alpha'(\Theta)h + \frac{1}{2}\alpha''(\Theta)h^2 + o(h^2) \quad (3.7)$$

Sustituyendo las series de Taylor 3.3 y 3.7 en la expresión 3.6, la **esperanza** del estimador de *diferencia central* viene dada por:

$$\mathbb{E}[\hat{\Delta}_C] = \frac{\alpha(\Theta + h) - \alpha(\Theta - h)}{2h} = \frac{2\alpha'(\Theta)h + o(h^2)}{2h} = \alpha'(\Theta) + o(h) \quad (3.8)$$

En consecuencia, el **sesgo** del estimador de *diferencia central* es el siguiente:

$$\text{Sesgo}(\hat{\Delta}_C) = \mathbb{E}[\hat{\Delta}_C - \alpha'(\Theta)] = o(h) \quad (3.9)$$

En general, el estimador de *diferencia central* es **más costoso computacionalmente** que el estimador de *diferencia hacia adelante* por la siguiente razón. Si estamos interesados en estimar $\alpha(\Theta)$ y $\alpha'(\Theta)$, normalmente simularíamos en Θ para estimar $\alpha(\Theta)$, lo que significa que el estimador de *diferencia hacia adelante* solo requiere realizar una simulación adicional en $\Theta + h$. Por otro lado, el estimador de *diferencia central* requiere realizar simulaciones en dos puntos adicionales, $\Theta + h$ y $\Theta - h$. Sin embargo, el estimador de *diferencia central* tiene una ventaja importante en términos de la **tasa de convergencia del sesgo**. Como se observa en las ecuaciones (3.5) y (3.9), el sesgo asociado al estimador de *diferencia central* es de orden $o(h)$, menor que el de *diferencia hacia adelante*.

Este comportamiento implica que, aunque el uso del estimador de *diferencia central* requiere un mayor costo computacional debido a la necesidad de realizar simulaciones en dos puntos, este incremento en el esfuerzo computacional se ve justificado por la mejora en la precisión de las estimaciones. En particular, su menor sesgo permite obtener derivadas más precisas, por lo tanto, en este trabajo utilizaremos el estimador de *diferencia central*.

El estimador de *diferencia central* puede extenderse para calcular derivadas de segundo orden. En este caso, la expresión del estimador viene dada por:

$$\hat{\Delta}_C^{(2)} = \hat{\Delta}_C(n, h)^{(2)} = \frac{\bar{Y}_n(\Theta + h) - 2\bar{Y}_n(\Theta) + \bar{Y}_n(\Theta - h)}{h^2} \quad (3.10)$$

La **esperanza** del estimador de *diferencia central* para derivadas de segundo orden, considerando las series de Taylor presentadas en las expresiones 3.3 y 3.7, es la siguiente:

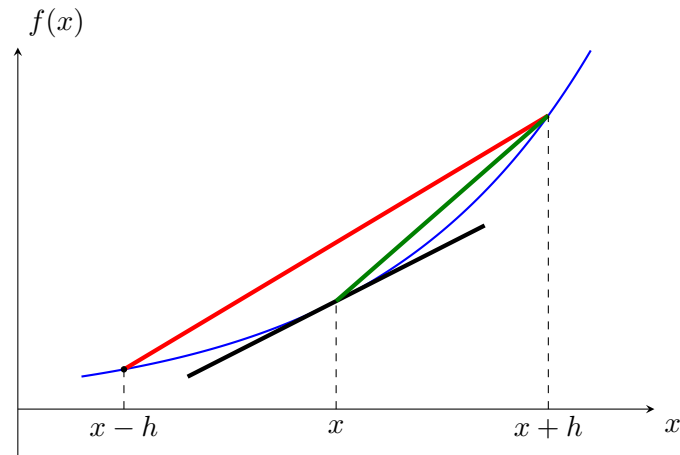
$$\mathbb{E}[\hat{\Delta}_C] = \frac{\alpha(\Theta + h) - 2\alpha(\Theta) + \alpha(\Theta - h)}{h^2} = \alpha''(\Theta) + o(h^2) \quad (3.11)$$

En consecuencia, el **sesgo** del estimador de *diferencia central* para segundas derivadas es el siguiente:

$$\text{Sesgo}(\hat{\Delta}_C^{(2)}) = \mathbb{E}[\hat{\Delta}_C^{(2)} - \alpha''(\Theta)] = o(h^2) \quad (3.12)$$

En la Figura 3.1 presentamos un gráfico que ilustra la aproximación de derivadas mediante los estimadores de *diferencia hacia adelante* y *diferencia centrada*. En la Figura 3.1 se observa que este último proporciona la mejor aproximación de la derivada de la función en el punto x , evidenciando su mayor precisión.

Figura 3.1: Gráfico de la aproximación de derivadas mediante el estimador de diferencia hacia adelante (verde) y el estimador central (rojo), siendo la recta la que representa la derivada exacta en el punto de interés. Fuente: Elaboración propia.



A continuación, en el Cuadro 3.1 se resumen las estimaciones de las griegas mediante el método de diferencias finitas a partir del estimador de *diferencia central*.

Cuadro 3.1: Estimación de las griegas de opciones europeas utilizando el estimador de *diferencia central*. En estas expresiones, S es el precio actual del activo subyacente, r es la tasa libre de riesgo, T es el tiempo hasta el vencimiento de la opción, σ es la volatilidad del activo subyacente y h representa un pequeño incremento. Fuente: Elaboración propia.

Griega	Estimación
Delta (Δ)	$\hat{\Delta} = \frac{V(S+h)-V(S-h)}{2h}$
Gamma (Γ)	$\hat{\Gamma} = \frac{V(S+h)-2V(S)+V(S-h)}{h^2}$
Vega (ν)	$\hat{\nu} = \frac{V(\sigma+h)-V(\sigma-h)}{2h}$
Theta (Θ)	$\hat{\Theta} = \frac{V(T+h)-V(T-h)}{2h}$
Rho (ρ)	$\hat{\rho} = \frac{V(r+h)-V(r-h)}{2h}$

3.2. Método de trayectoria

El método de trayectoria se presenta como una alternativa al método de diferencias finitas descrito previamente, ya que permite estimar las derivadas directamente sin la necesidad de simular múltiples valores del parámetro. La principal diferencia con el método anterior es que, estimamos la derivada de $\alpha(\Theta) = \mathbb{E}[Y(\Theta)]$ directamente con la definición:

$$Y'(\Theta) = \lim_{h \rightarrow 0} \frac{Y(\Theta + h) - Y(\Theta)}{h} \quad (3.13)$$

Este estimador tiene como **esperanza** $\mathbb{E}[Y'(\Theta)]$ y es un estimador **insesgado** de $\alpha'(\Theta)$ si se verifica que:

$$\mathbb{E} \left[\frac{dY(\Theta)}{d\Theta} \right] = \frac{d}{d\Theta} \mathbb{E}[Y(\Theta)] \quad (3.14)$$

es decir, si se cumplen una serie de condiciones que justifican el intercambio de la operación de diferenciación con la de esperanza. No obstante, este tema está fuera del alcance de este trabajo, por lo que no se explicarán dichas condiciones. Por último, cabe destacar que este método no es aplicable para estimar derivadas de segundo orden, lo cual se discutirá más adelante cuando tratemos de obtener una estimación para la Gamma.

A continuación, deduciremos las estimaciones por el método de trayectoria para cada una de las griegas apoyándonos en la expresión 3.14. Comencemos definiendo el beneficio descontado asociado a una opción financiera. Sea Y_c el beneficio descontado de una opción de **compra** y, Y_v en el caso de una opción de **venta**, estas cantidades se expresan como:

$$Y_c = e^{-rT} \max(0, S(T) - K) \quad (3.15)$$

$$Y_v = e^{-rT} \max(0, K - S(T)) \quad (3.16)$$

donde $S(T) = S(0)e^{(r - \frac{1}{2}\sigma^2)T + \sigma\sqrt{T}Z}$, con $Z \sim N(0, 1)$ y r, σ, T y K constantes positivas.

- Delta

El Delta mide la sensibilidad del valor de la opción con respecto al precio del activo subyacente y su estimación con este método se obtiene aplicando la *Regla de la cadena*.

$$\frac{dY_c}{dS(0)} = \frac{dY}{dS(T)} \cdot \frac{dS(T)}{dS(0)} \quad (3.17)$$

Para el primer factor, observamos que:

$$\frac{d}{dS(T)} \max(0, S(T) - K) = \begin{cases} 0, & \text{si } S(T) < K \\ 1, & \text{si } S(T) > K \end{cases} \quad (3.18)$$

Esta derivada no existe en $S(T) = K$, ya que hay una discontinuidad en este punto al no coincidir los límites laterales. Sin embargo, dado que el evento $\{S(T) = K\}$ tiene probabilidad 0 por seguir una distribución continua (normal estándar), Y es diferenciable casi seguramente con respecto a $S(T)$ y su derivada es:

$$\frac{dY_c}{dS(T)} = e^{-rT} \mathbf{1}\{S(T) > K\} \quad (3.19)$$

donde $\mathbf{1}\{S(T) > K\}$ representa la función indicatriz.

Para el segundo factor, obtenemos que:

$$\frac{dS(T)}{dS(0)} = \frac{e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z}}{S(0)} = \frac{S(T)}{S(0)} \quad (3.20)$$

Sustituyendo los resultados de las expresiones (3.19) y (3.20) en la ecuación (3.17), obtenemos el estimador para Δ :

$$\frac{dY_c}{dS(0)} = e^{-rT} \frac{S(T)}{S(0)} \mathbf{1}\{S(T) > K\} \quad (3.21)$$

Este estimador es *insesgado*, ya que su valor esperado es el Delta de Black and Scholes.

$$\Delta_c = \mathbb{E} \left[\frac{dY_c}{dS(0)} \right] = \mathbb{E} \left[e^{-rT} \frac{S(T)}{S(0)} \mathbf{1}\{S(T) > K\} \right] \quad (3.22)$$

Análogamente, en el caso de una opción de **venta**, para el primer factor, tenemos que:

$$\frac{d}{dS(T)} \max(0, K - S(T)) = \begin{cases} -1, & \text{si } S(T) < K \\ 0, & \text{si } S(T) > K \end{cases} \quad (3.23)$$

En consecuencia, obtenemos que el estimador también es *insesgado* y es el siguiente:

$$\frac{dY_v}{dS(0)} = -e^{-rT} \frac{S(T)}{S(0)} \mathbf{1}\{S(T) < K\} \quad (3.24)$$

- Gamma

El método de la Trayectoria no es adecuado para obtener derivadas de segundo orden. La Gamma mide la sensibilidad de la Delta al precio del activo subyacente. Si intentamos derivar la expresión 3.19, observamos que el beneficio descontado Y_c es diferenciable excepto en $S(T) = K$ y dado que Y_c es constante en el resto de valores de $S(T)$, la derivada es 0 en todos los puntos donde existe, por tanto no podemos obtener un valor de la Gamma.

$$\mathbb{E} \left[\frac{dY_c}{dS(0)} \right] = 0 \quad (3.25)$$

- Vega

La **Vega** mide la sensibilidad del valor de la opción con respecto a la volatilidad del activo subyacente. Debemos tener en cuenta que la Vega es la misma tanto para opciones de compra como de venta, lo cual se presentó en el Cuadro 2.4. Su estimación mediante el método de la trayectoria se obtiene aplicando la *Regla de la cadena*.

$$\frac{dY_c}{d\sigma} = \frac{dY_c}{dS(T)} \cdot \frac{dS(T)}{d\sigma} \quad (3.26)$$

El primer factor es el mismo que obtuvimos en la ecuación (3.19), y el segundo viene dado por la siguiente expresión:

$$\frac{dS(T)}{d\sigma} = S(0)e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z} \left(-\sigma T + \sqrt{T}Z\right) = S(T) \left(-\sigma T + \sqrt{T}Z\right) \quad (3.27)$$

Combinando las expresiones (3.19) y (3.27), obtenemos el estimador para \mathcal{V} :

$$\frac{dY_c}{d\sigma} = e^{-rT} \left(-\sigma T + \sqrt{T}Z\right) S(T) \mathbf{1}\{S(T) > K\} \quad (3.28)$$

donde, usando la ecuación (2.14), podemos expresar la Z de la siguiente manera:

$$Z = \frac{\ln\left(\frac{S(T)}{S(0)}\right) - \left(r - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} \quad (3.29)$$

Este estimador es *insesgado*, ya que su valor esperado es el Vega de Black and Scholes.

$$\mathcal{V} = \mathbb{E} \left[\frac{dY_c}{d\sigma} \right] \quad (3.30)$$

- Theta

La Theta mide la sensibilidad del precio de la opción al tiempo restante hasta el vencimiento. El estimador del método de la Trayectoria para el *Theta* de Black and Scholes, utilizando la *Regla de la Cadena*, viene dado por la siguiente derivada:

$$\frac{dY_c}{dT} = - \left(\frac{d}{dT} (e^{-rT}) \max(0, S(T) - K) + e^{-rT} \frac{d}{dT} \max(0, S(T) - K) \right) \quad (3.31)$$

El primer término de esta expresión se deriva fácilmente como:

$$\frac{d}{dT} (e^{-rT}) \max(0, S(T) - K) = -re^{-rT} \max(0, S(T) - K) \quad (3.32)$$

Para el segundo término, utilizando el resultado de la ecuación (3.19), obtenemos la siguiente expresión:

$$\frac{d}{dT} (\max(0, S(T) - K)) = \mathbf{1}\{S(T) > K\} \frac{dS(T)}{dT} \quad (3.33)$$

donde

$$\frac{dS(T)}{dT} = S(T) \left[\left(r - \frac{1}{2}\sigma^2 \right) + \frac{\sigma Z}{2\sqrt{T}} \right] \quad (3.34)$$

Sustituyendo las ecuaciones (3.32) y (3.33) en la derivada (3.31), obtenemos el estimador para Θ :

$$\frac{dY_c}{dT} = e^{-rT} (r \max(0, S(T) - K) - \mathbf{1}\{S(T) > K\} \left[\left(r - \frac{1}{2}\sigma^2 \right) + \frac{\sigma Z}{2\sqrt{T}} \right] S(T)) \quad (3.35)$$

Este estimador es *insesgado*, pues su valor esperado es el Theta de Black and Scholes.

$$\Theta = \mathbb{E} \left[\frac{dY_c}{dT} \right] \quad (3.36)$$

Análogamente, para una opción de **venta** el estimador correspondiente al Theta de una opción de venta se obtiene siguiendo el mismo procedimiento.

$$\frac{dY_v}{dT} = e^{-rT} (r \max(0, K - S(T)) + \mathbf{1}\{S(T) > K\} \left[\left(r - \frac{1}{2}\sigma^2 \right) + \frac{\sigma Z}{2\sqrt{T}} \right] S(T)) \quad (3.37)$$

Al igual que en el caso de la opción de compra, el valor esperado de este estimador coincide con el Theta de Black and Scholes, asegurando que también es un estimador *insesgado*.

- Rho

El Rho mide la sensibilidad del precio de la opción a la tasa de interés. El estimador por el método de la Trayectoria para el *Rho*, haciendo uso de la *Regla de la Cadena*, viene dado por la derivada:

$$\frac{dY_c}{dr} = \left(\frac{d}{dr} (e^{-rT}) \max(0, S(T) - K) + e^{-rT} \frac{d}{dr} \max(0, S(T) - K) \right) \quad (3.38)$$

El primer término de esta expresión es:

$$\frac{d}{dr} (e^{-rT}) \max(0, S(T) - K) = -T e^{-rT} \max(0, S(T) - K) \quad (3.39)$$

Para el segundo término, utilizando la ecuación (3.19), obtenemos:

$$\frac{d}{dr} (\max(0, S(T) - K)) = \mathbf{1}\{S(T) > K\} \frac{dS(T)}{dr} \quad (3.40)$$

donde

$$\frac{dS(T)}{dr} = S(0)e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z} \cdot T = S(T) \cdot T \quad (3.41)$$

Sustituyendo las expresiones (3.39) y (3.40) en la derivada (3.38), obtenemos el estimador para ρ :

$$\frac{dY_c}{dr} = e^{-rT}(-T \text{máx}(0, S(T) - K) + \mathbf{1}\{S(T) > K\}S(T)T) \quad (3.42)$$

Este estimador es *insesgado*, debido a que su valor esperado es el Rho de Black and Scholes.

$$\rho = \mathbb{E} \left[\frac{dY_c}{dr} \right] \quad (3.43)$$

Análogamente, para una opción de **venta** tendremos que el estimador viene dado por:

$$\frac{dY_v}{dr} = e^{-rT}(-T \text{máx}(0, S(T) - K) + \mathbf{1}\{S(T) > K\}S(T)T) \quad (3.44)$$

De manera similar a la opción de compra, el valor esperado de este estimador equivale al Rho de Black and Scholes, lo que garantiza que se trata de un estimador *insesgado*.

El Cuadro 3.2 resume las estimaciones que hemos obtenido para las griegas de opciones europeas de compra y venta utilizando el método de trayectoria. Los estimadores se obtienen mediante simulaciones Monte Carlo del precio de $S(T)$, ya que sus trayectorias dependen de las realizaciones aleatorias de Z .

Cuadro 3.2: Estimación de las griegas de opciones europeas de compra utilizando el método de trayectoria. $S(T)$ es el precio final del activo subyacente, S_0 es el precio inicial del activo subyacente, K es el precio de ejercicio de la opción, r es la tasa libre de riesgo, T es el tiempo hasta el vencimiento de la opción, y σ es la volatilidad del activo subyacente. La función indicatriz $\mathbf{1}$ representa tanto opciones de compra como de venta, donde para opciones de compra es $\mathbf{1}\{S(T) > K\}$ y para opciones de venta $-\mathbf{1}\{S(T) < K\}$. Finalmente, Y denota el beneficio descontado de la opción correspondiente. Fuente: Elaboración propia.

Griega	Estimación
Delta (Δ)	$e^{-rT} \mathbb{E} \left[\frac{S(T)}{S(0)} \mathbf{1} \right]$
Vega (ν)	$e^{-rT} \mathbb{E} \left[(-\sigma T + \sqrt{T}Z) S(T) \mathbf{1} \right]$
Theta (Θ)	$\mathbb{E} \left[rY + e^{-rT} \left[\left(r - \frac{1}{2}\sigma^2 \right) + \frac{\sigma Z}{2\sqrt{T}} \right] S(T) \mathbf{1} \right]$
Rho (ρ)	$\mathbb{E} \left[-TY + e^{-rT} S(T) T \mathbf{1} \right]$

3.3. Método de razón de verosimilitudes

El método de razón de verosimilitudes es una técnica alternativa para estimar derivadas, que se presenta como una solución a algunas de las limitaciones del método de trayectorias. A diferencia de este último, el método de razón de verosimilitudes no requiere que los pagos descontados, como función del parámetro de diferenciación, sean continuos y puede aplicarse fácilmente para estimar segundas derivadas.

Consideremos un **pago descontado** Y expresado como una función f de un vector aleatorio $X = (X_1, \dots, X_m)$. Suponemos que X tiene una densidad de probabilidad g_Θ , siendo Θ un parámetro de esta densidad. La **esperanza** del pago descontado, $\mathbb{E}_\Theta[Y]$, se puede expresar como:

$$\mathbb{E}_\Theta[Y] = \mathbb{E}_\Theta[f(X_1, \dots, X_m)] = \int_{\mathbb{R}^m} f(x) g_\Theta(x) dx \quad (3.45)$$

Para obtener un estimador de la derivada de esta esperanza, suponemos que se cumplen las hipótesis del *teorema de derivación bajo el signo integral*¹, es decir, el orden de diferenciación e integración puede ser intercambiado, lo que nos lleva a:

$$\frac{d}{d\Theta} \mathbb{E}_\Theta[Y] = \int_{\mathbb{R}^m} f(x) \frac{d}{d\Theta} g_\Theta(x) dx \quad (3.46)$$

En consecuencia, multiplicando y dividiendo el integrando por $g_\Theta(x)$ obtenemos:

$$\frac{d}{d\Theta} \mathbb{E}_\Theta[Y] = \int_{\mathbb{R}^m} f(x) \frac{g'_\Theta(x)}{g_\Theta(x)} g_\Theta(x) dx = \mathbb{E}_\Theta \left[f(X) \frac{g'_\Theta(X)}{g_\Theta(X)} \right] \quad (3.47)$$

donde $g'_\Theta(x)$ denota la derivada de $g_\Theta(x)$ con respecto a Θ y nos referimos al término $\frac{g'_\Theta(x)}{g_\Theta(x)}$ como el *score*². De este modo, a partir de la expresión 3.47, deducimos que el estimador *insesgado* de la derivada de $\mathbb{E}_\Theta[Y]$, es decir, el estimador del método de razón de verosimilitudes, viene dado por la siguiente expresión:

$$f(X) \frac{g'_\Theta(X)}{g_\Theta(X)} \quad (3.48)$$

¹**Teorema:** Sea U un subconjunto compacto de $\mathbb{R}^n \times \mathbb{R}^m$, y sea $f : U \rightarrow \mathbb{R}$ una función continua tal que la derivada existe y es continua en todo U . Si A y B son subconjuntos con volumen de \mathbb{R}^n y \mathbb{R}^m respectivamente, tales que B es abierto y $A \times B \subseteq U$, entonces la función $F : B \rightarrow \mathbb{R}$ definida por $F(y) = \int_A f(x, y) dx$ es diferenciable en B , y $F'(y) = \int_A \frac{\partial f}{\partial y}(x, y) dx \quad \forall y \in B$.

²Aunque el logaritmo no aparece de forma explícita, este cociente equivale a la derivada del logaritmo de la densidad:

$$\frac{g'_\Theta(x)}{g_\Theta(x)} = \frac{d}{d\Theta} \log g_\Theta(x),$$

por lo que en la literatura estadística se le conoce como función de puntuación o *score function*.

A continuación, tratamos de estimar las griegas de Black and Scholes utilizando el método de razón de verosimilitudes. Para ello, necesitamos considerar $S(0)$ como un parámetro de la densidad de $S(T)$. La densidad lognormal de $S(T)$ viene dada por:

$$g(S(T)) = \frac{1}{S(T)\sigma\sqrt{T}}\phi(\zeta(S(T))) \quad (3.49)$$

donde ϕ es la densidad normal estándar y $\zeta(S(T))$ está dada por la siguiente expresión:

$$\zeta(S(T)) = \frac{\ln\left(\frac{S(T)}{S(0)}\right) - \left(r - \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \quad (3.50)$$

- Delta

El Delta de la opción mide la sensibilidad del valor de la opción con respecto al precio del activo subyacente, por lo que, el *score* asociado a Delta se define de la siguiente manera:

$$\frac{\frac{dg(S(T))}{dS(0)}}{g(S(T))} \quad (3.51)$$

Comencemos analizando el numerador de esta expresión:

$$\frac{dg(S(T))}{dS(0)} = \frac{1}{S(T)\sigma\sqrt{T}} \cdot \phi'(\zeta(S(T))) \frac{d\zeta(S(T))}{dS(0)} \quad (3.52)$$

Por un lado, la función de densidad normal estándar $\phi(z) = \frac{1}{\sqrt{2\pi}}e^{-z^2/2}$ tiene derivada $\phi'(z) = -z\phi(z)$. Por lo tanto, la derivada de $\phi(\zeta(S(T)))$ viene dada por:

$$\phi'(\zeta(S(T))) = -\zeta(S(T))\phi(\zeta(S(T))) \quad (3.53)$$

Por otro lado, hallamos la derivada de $\zeta(S(T))$ con respecto al precio inicial $S(0)$.

$$\frac{d\zeta(S(T))}{dS(0)} = -\frac{1}{S(0)\sigma\sqrt{T}} \quad (3.54)$$

Sustituyendo las derivadas anteriores (3.53) y (3.54) en la expresión (3.52) obtenemos:

$$\frac{dg(S(T))}{dS(0)} = \frac{-\zeta(S(T))\phi(\zeta(S(T)))}{S(T)\sigma\sqrt{T}} \cdot \left(-\frac{1}{S(0)\sigma\sqrt{T}}\right) = \frac{\zeta(S(T))\phi(\zeta(S(T)))}{S(T)S(0)\sigma^2T} \quad (3.55)$$

En consecuencia, la expresión general del *score* de Delta sería:

$$\frac{\frac{dg(S(T))}{dS(0)}}{g(S(T))} = \frac{\frac{\zeta(S(T))\phi(\zeta(S(T)))}{S(T)S(0)\sigma^2T}}{\frac{1}{S(T)\sigma\sqrt{T}}\phi(\zeta(S(T)))} = \frac{\zeta(S(T))}{S(0)\sigma\sqrt{T}} = \frac{\log\left(\frac{S(T)}{S(0)}\right) - \left(r - \frac{\sigma^2}{2}\right)T}{S(0)\sigma^2T} \quad (3.56)$$

El estimador *insesgado* de Delta que obtenemos es:

$$e^{-rT} \max(0, S(T) - K) \frac{\log\left(\frac{S(T)}{S(0)}\right) - \left(r - \frac{\sigma^2}{2}\right) T}{S(0)\sigma^2 T} \quad (3.57)$$

donde $e^{-rT} \max(0, S(T) - K)$ es el valor descontado de la opción de compra al tiempo T .

Finalmente, si $S(T)$ es generado a partir de $S(0)$ usando una variable aleatoria normal estándar Z dada por la expresión (2.14), entonces $\zeta(S(T)) = Z$, y el estimador se simplifica a la siguiente expresión:

$$e^{-rT} \max(0, S(T) - K) \frac{Z}{S(0)\sigma\sqrt{T}} \quad (3.58)$$

- Gamma

La Gamma mide la sensibilidad de la Delta al precio del activo subyacente. De forma que, en este caso, el *score* es el siguiente cociente:

$$\frac{\frac{d^2 g(S(T))}{dS^2(0)}}{g(S(T))} \quad (3.59)$$

La derivada de primer orden ha sido calculada para obtener el estimador de Delta, que se muestra en la expresión (3.55). De modo que, procedemos a derivar nuevamente aplicando la *Regla del Cociente* y utilizando el resultado (3.54) para la derivada de $\zeta(S(T))$:

$$\begin{aligned} \frac{d^2 g(S(T))}{dS^2(0)} &= \frac{[\zeta'(S(T))\phi(\zeta(S(T))) + \zeta(S(T))\phi'(\zeta(S(T)))\zeta'(S(T))] S(T)S(0)\sigma^2 T}{(S(T)S(0)\sigma^2 T)^2} \\ &\quad - \frac{\zeta(S(T))\phi(\zeta(S(T)))S(T)\sigma^2\sqrt{T}}{(S(T)S(0)\sigma^2 T)^2} \\ &= \frac{S(T)\sigma^2 T \{[\zeta'(S(T))\phi(\zeta(S(T))) + \zeta(S(T))\phi'(\zeta(S(T)))\zeta'(S(T))] S(0) - \zeta(S(T))\phi(\zeta(S(T)))\}}{(S(T)S(0)\sigma^2 T)^2} \\ &= \frac{[\zeta'(S(T))\phi(\zeta(S(T))) + \zeta(S(T))\phi'(\zeta(S(T)))\zeta'(S(T))] S(0) - \zeta(S(T))\phi(\zeta(S(T)))}{S(T)S(0)^2\sigma^2 T} \\ &= \frac{\left[-\frac{1}{S(0)\sigma\sqrt{T}}\phi(\zeta(S(T))) + \zeta(S(T))(-\zeta(S(T))\phi(\zeta(S(T))))\left(-\frac{1}{S(0)\sigma\sqrt{T}}\right)\right] S(0)}{S(T)S(0)^2\sigma^2 T} \\ &\quad - \frac{-\zeta(S(T))\phi(\zeta(S(T)))}{S(T)S(0)^2\sigma^2 T} \\ &= \frac{\phi(\zeta(S(T))) \left\{ \frac{1}{S(0)\sigma\sqrt{T}} [\zeta^2(S(T)) - 1] S(0) - \zeta(S(T)) \right\}}{S(T)S(0)^2\sigma^2 T} \end{aligned}$$

En consecuencia, la expresión del estimador de Gamma que obtenemos es la siguiente:

$$\frac{\frac{d^2 g(S(T))}{dS^2(0)}}{g(S(T))} = \frac{\frac{\phi(\zeta(S(T))) \left\{ \frac{1}{S(0)\sigma\sqrt{T}} [\zeta^2(S(T)) - 1] S(0) - \zeta(S(T)) \right\}}{S(T)S(0)^2\sigma^2 T}}{\frac{1}{S(T)\sigma\sqrt{T}} \phi(\zeta(S(T)))} = \frac{\left[\frac{1}{S(0)\sigma\sqrt{T}} (\zeta^2(S(T)) - 1) S(0) - \zeta(S(T)) \right]}{S(0)^2\sigma\sqrt{T}}$$

Simplificando nos queda la siguiente expresión para el *score*:

$$\frac{\frac{d^2 g(S(T))}{dS^2(0)}}{g(S(T))} = \frac{\zeta(S(T))^2 - 1}{S(0)^2\sigma^2 T} - \frac{\zeta(S(T))}{S(0)^2\sigma\sqrt{T}} \quad (3.60)$$

Si $S(T)$ es generado a partir de $S(0)$ usando una variable aleatoria normal estándar Z , es decir, $\zeta(S(T)) = Z$, el estimador de la segunda derivada se simplifica a:

$$e^{-rT} \max(0, S(T) - K) \left(\frac{Z^2 - 1}{S(0)^2\sigma^2 T} - \frac{Z}{S(0)^2\sigma\sqrt{T}} \right) \quad (3.61)$$

- Vega

La **Vega** mide la sensibilidad del valor de la opción con respecto a la volatilidad del activo subyacente. En consecuencia, para estimar Vega, la función *score* es la siguiente:

$$\frac{\frac{dg(S(T))}{d\sigma}}{g(S(T))} \quad (3.62)$$

Para obtener esta expresión, primero examinaremos el numerador:

$$\frac{dg(S(T))}{d\sigma} = \frac{\phi'(\zeta(S(T))) \frac{d\zeta(S(T))}{d\sigma} S(T)\sigma\sqrt{T} - \phi(\zeta(S(T))) S(T)\sqrt{T}}{(S(T)\sigma\sqrt{T})^2} \quad (3.63)$$

donde recordemos que la derivada de la densidad normal estándar está dada por la ecuación (3.53) y la derivada de $\zeta(S(T))$ con respecto a la volatilidad σ es:

$$\frac{d\zeta(S(T))}{d\sigma} = \frac{\log(S(0)/S(T)) + (r + \frac{\sigma^2}{2})T}{\sigma^2\sqrt{T}} \quad (3.64)$$

El *score* de Vega sustituyendo las derivadas anteriores vendrá dado por:

$$\frac{\frac{dg(S(T))}{d\sigma}}{g(S(T))} = -\frac{1}{\sigma} - \zeta(S(T)) \frac{d\zeta(S(T))}{d\sigma} = -\frac{1}{\sigma} - \zeta(S(T)) \frac{\log(S(0)/S(T)) + (r + \frac{\sigma^2}{2})T}{\sigma^2\sqrt{T}} \quad (3.65)$$

Finalmente, si $\zeta(S(T)) = Z$, el estimador de razón de verosimilitud de Vega puede expresarse como:

$$e^{-rT} \max(0, S(T) - K) \left(\frac{Z^2 - 1}{\sigma} - Z\sqrt{T} \right) \quad (3.66)$$

- Theta

La Theta mide la sensibilidad del precio de la opción al tiempo restante hasta el vencimiento, entonces la función *score* para estimar Theta se define como:

$$\frac{\frac{dg(S(T))}{dT}}{g(S(T))} \quad (3.67)$$

En primer lugar, obtengamos la derivada del numerador mediante la *Regla del cociente*:

$$\begin{aligned} \frac{dg(S(T))}{dT} &= \frac{\phi'(\zeta(S(T)))\zeta'(S(T))S(T)\sigma\sqrt{T} - \phi(\zeta(S(T)))S(T)\sigma\frac{1}{2\sqrt{T}}}{(S(T)\sigma\sqrt{T})^2} \\ &= \frac{\phi'(\zeta(S(T)))\zeta'(S(T))}{S(T)\sigma\sqrt{T}} - \frac{\phi(\zeta(S(T)))}{2S(T)\sigma T\sqrt{T}} \\ &= \frac{-\phi(\zeta(S(T)))}{S(T)\sigma\sqrt{T}} \left[\zeta(S(T))\zeta'(S(T)) + \frac{\zeta(S(T))}{2T} \right] \end{aligned}$$

Ahora, dividiendo por $g(S(T))$, obtenemos la expresión del *score*:

$$\frac{\frac{dg(S(T))}{dT}}{g(S(T))} = \frac{d\zeta(S(T))}{dT} \cdot (-\zeta(S(T))) - \frac{1}{2T}\zeta(S(T)) \quad (3.68)$$

donde se tiene que:

$$\begin{aligned} \frac{d\zeta(S(T))}{dT} &= \frac{-(r - \frac{\sigma^2}{2})\sigma\sqrt{T} - (\ln\left(\frac{S(T)}{S(0)}\right) - \left(r - \frac{\sigma^2}{2}\right)T)\frac{\sigma}{2\sqrt{T}}}{\sigma^2 T} \\ &= \frac{-(\ln\left(\frac{S(T)}{S(0)}\right) - \left(r - \frac{\sigma^2}{2}\right)T)}{2\sigma T\sqrt{T}} - \frac{r - \frac{\sigma^2}{2}}{\sigma\sqrt{T}} \end{aligned} \quad (3.69)$$

Si $\zeta(S(T)) = Z$, el estimador resultante para Θ es:

$$-e^{-rT} \max(0, S(T) - K) \left[\frac{1}{2T} (Z^2 - 1) + \frac{r - \frac{\sigma^2}{2}}{\sigma\sqrt{T}} Z \right] \quad (3.70)$$

- Rho

El Rho mide la sensibilidad del precio de la opción a la tasa de interés, por lo que la función *score* viene dada por:

$$\frac{\frac{dg(S(T))}{dr}}{g(S(T))} \quad (3.71)$$

Estudiemos primero el numerador:

$$\frac{dg(S(T))}{dr} = \frac{1}{S(T)\sigma\sqrt{T}} \cdot \phi'(\zeta(S(T))) \frac{d\zeta(S(T))}{dr} \quad (3.72)$$

donde sabemos que la derivada de $\phi(\zeta(S(T)))$ viene dada por la ecuación (3.53) y la derivada de $\zeta(S(T))$ respecto r es la siguiente:

$$\frac{d\zeta(S(T))}{dr} = -\frac{\sqrt{T}}{\sigma} \quad (3.73)$$

De esta forma, a partir de las derivadas (3.53) y (3.73) el *score* de Rho es el siguiente:

$$\frac{\frac{dg(S(T))}{dr}}{g(S(T))} = \frac{\frac{-\zeta(S(T))\phi(\zeta(S(T)))}{S(T)\sigma\sqrt{T}} \cdot \left(-\frac{\sqrt{T}}{\sigma}\right)}{\frac{1}{S(T)\sigma\sqrt{T}}\phi(\zeta(S(T)))} = \zeta(S(T))\frac{\sqrt{T}}{\sigma} \quad (3.74)$$

Finalmente, si $\zeta(S(T)) = Z$, el estimador resultante para ρ se define como:

$$e^{-rT} \max(0, S(T) - K) \frac{Z\sqrt{T}}{\sigma} \quad (3.75)$$

Análogamente, para opciones de venta, simplemente sustituiremos el beneficio descontado de una opción de compra, dado por $Y_c = e^{-rT} \max(0, S(T) - K)$, por el beneficio descontado de una opción de venta, que se expresa como $Y_v = e^{-rT} \max(0, K - S(T))$.

El Cuadro 3.3 resume las estimaciones de las griegas para opciones europeas de compra y venta que hemos obtenido y demostrado previamente con el método de razón de verosimilitudes.

Cuadro 3.3: Estimación de las griegas de opciones europeas utilizando el método de razón de verosimilitudes. En estas expresiones, tenemos que $S(0)$ es el precio inicial, σ es la volatilidad, T es el tiempo hasta vencimiento, y r es la tasa libre de riesgo. Además, $Z = \zeta(S(T))$ e Y denota el beneficio descontado de una opción de compra o venta. Fuente: Elaboración propia.

Griega	Estimación
Delta (Δ)	$\mathbb{E} \left[Y \frac{1}{S(0)\sigma\sqrt{T}} Z \right]$
Gamma (Γ)	$\mathbb{E} \left[Y \frac{1}{S(0)^2\sigma^2 T} \left(Z^2 - 1 - \sigma\sqrt{T}Z \right) \right]$
Vega (ν)	$\mathbb{E} \left[Y \left(\frac{1}{\sigma} (Z^2 - 1) - \sqrt{T}Z \right) \right]$
Theta (Θ)	$\mathbb{E} \left[Y \left(\frac{1}{2T} (Z^2 - 1) + \frac{r - \frac{\sigma^2}{2}}{\sigma\sqrt{T}} Z \right) \right]$
Rho (ρ)	$\mathbb{E} \left[Y \frac{Z\sqrt{T}}{\sigma} \right]$

Capítulo 4

Datos

En este capítulo se presenta el conjunto de datos utilizado para valorar una cartera compuesta por 30 opciones europeas, de las cuales 20 son sobre materias primas y 10 sobre acciones sin dividendos. Se incluyen tanto opciones de compra como de venta para 5 vencimientos diferentes y para niveles de *moneyness* de 0.9 y 1.1. Este conjunto de datos permite analizar el rendimiento de los tres métodos seleccionados para la estimación de las griegas.

4.1. Cartera de opciones

Nuestra cartera se compone de un conjunto de opciones sobre activos subyacentes que no distribuyen dividendos con el objetivo de comparar los resultados de la estimación de las griegas con los resultados teóricos del modelo de Black and Scholes, ya que asume estas condiciones. Por este motivo, hemos escogido opciones sobre materias primas¹ y sobre acciones sin dividendos, que se presentan en el Cuadro 4.1.

Cuadro 4.1: Conjunto de commodities y acciones que no pagan dividendos junto con sus respectivos tickers en *Yahoo Finance*. Fuente: Elaboración propia.

Categoría	Commodities
Metales	Oro (GC=F), Plata (SI=F), Cobre (HG=F), Platino (PL=F), Paladio (PA=F).
Energía	Petróleo Crudo WTI (CL=F), Petróleo Crudo Brent (BZ=F), Gas Natural (NG=F), Gasolina RBOB (RB=F), Fuel Oil (HO=F).
Agricultura	Maíz (ZC=F), Trigo (ZW=F), Café (KC=F), Azúcar (SB=F), Algodón (CT=F), Soja (ZS=F), Avena (ZO=F), Arroz (ZR=F), Cacao (CC=F), Zumo de Naranja (OJ=F).

Empresa	Ticker	Empresa	Ticker
Google	GOOGL	Adobe	ADBE
Amazon	AMZN	Shopify	SHOP
Tesla	TSLA	Atlassian	TEAM
Netflix	NFLX	Salesforce	CRM
Meta	META	ServiceNow	NOW

¹Los mercados de materias primas o de “commodities” son los mercados financieros más antiguos. Una materia prima es un bien físico no diferenciado que satisface una determinada necesidad, siendo fundamentales para el comercio mundial y cuya negociación tiene un impacto significativo en la economía global. Los índices más conocidos que las agrupan incluyen: (1) el Índice Thomson Reuters/Jefferies CRB, (2) el S&P GSCI, (3) el Índice Dow Jones-UBS y (4) el Rogers International Commodity Index, cada uno con su propia composición y ponderación según la importancia de cada sector.

4.2. Fuentes de datos

En este apartado se describen las distintas fuentes utilizadas, así como las distintas características de los datos para cada opción. Esto incluye la selección de fechas de vencimiento, los precios de los activos subyacentes, los precios de ejercicio, las volatilidades y la tasa de interés libre de riesgo.

En primer lugar, la **fecha de valoración** y las **fechas de vencimiento** de las opciones han sido establecidas de la siguiente manera. Por un lado, la fecha de valoración se ha fijado como el *1 de diciembre de 2016* con el objetivo de evitar distorsiones en el análisis provocadas por los eventos macroeconómicos y geopolíticos recientes, como la pandemia COVID-19, la posterior salida de la crisis, los altos niveles de inflación global y los conflictos bélicos. De este modo, evaluamos los métodos de estimación de las griegas en un entorno económico representativo de condiciones de mercado relativamente normales.

Por otro lado, hemos considerado vencimientos a 3, 6, 9, 12 y 24 meses, con la finalidad de abarcar diversos horizontes temporales, tanto de corto como de medio plazo y poder analizar cómo varía el comportamiento de las griegas según el tiempo restante hasta el vencimiento. Además, estos representan plazos de vencimiento habituales de las opciones financieras, lo que hace que los resultados obtenidos permitan extraer conclusiones aplicables a la realidad. El tiempo restante hasta el vencimiento (T) se ha obtenido como la diferencia entre la fecha de valoración y la de vencimiento en número de años.

En segundo lugar, el **precio del subyacente** para cada opción en la fecha de valoración, es decir, el precio inicial del activo subyacente, se ha obtenido a través de la librería *yfinance*, la cual permite acceder a datos históricos proporcionados por Yahoo Finance.

En tercer lugar, el **precio de ejercicio** se define a partir del nivel de *moneyness*, el cual se ha fijado en dos valores: 0,9 y 1,1, que representan aquellas situaciones en las que resulta rentable ejercer la opción, pues están cerca de estar ATM. En particular, el **precio de ejercicio** se calcula utilizando la siguiente fórmula:

$$\text{Moneyness} = \frac{\text{Precio del subyacente}}{\text{Precio de ejercicio}} \quad (4.1)$$

En cuarto lugar, la **volatilidad** que hemos considerado es la *volatilidad histórica*, la cual hemos obtenido siguiendo la metodología descrita por Hull and White [1987]. A continuación, se detalla dicho procedimiento para estimar la volatilidad a partir de los precios históricos del activo subyacente.

Dado un conjunto de precios de cierre S_0, S_1, \dots, S_n , donde S_i corresponde al precio del activo subyacente al final del i -ésimo intervalo, entonces el retorno logarítmico en i se define como:

$$u_i = \ln \left(\frac{S_i}{S_{i-1}} \right) \quad (4.2)$$

Después, obtenemos la desviación estándar de los u_i para estimar la volatilidad histórica a partir de estos retornos logarítmicos:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (u_i - \bar{u})^2} \quad (4.3)$$

donde \bar{u} es el promedio de los retornos logarítmicos y n es el número de observaciones. En consecuencia, la estimación de la volatilidad diaria es:

$$\hat{\sigma}_{\text{diaria}} = \frac{s}{\sqrt{\tau}} \quad (4.4)$$

donde τ es la longitud del tiempo en años. Por último, la volatilidad anualizada se obtiene multiplicando la desviación estándar diaria por la raíz cuadrada del número de días de negociación al año, generalmente 252:

$$\hat{\sigma}_{\text{anualizada}} = \hat{\sigma}_{\text{diaria}} \times \sqrt{252} \quad (4.5)$$

Finalmente, en cuanto a la **tasa libre de riesgo**, hemos optado por utilizar los **bonos del gobierno de EE.UU. a 10 años**. Aunque el horizonte temporal de las opciones analizadas es de un máximo de dos años, se justifica la elección de dicho activo libre de riesgo siguiendo la recomendación de Bruner et al. [1998], quienes argumentan que en la práctica es el más adecuado. Dado que las opciones se valoran en fecha 1 de diciembre de 2016, se emplea el rendimiento correspondiente al 30 de noviembre de 2016, ya que, en la fecha de valoración, la única información disponible corresponde al cierre del día anterior. Al igual que los datos previos, este ha sido obtenido mediante la librería *yfinance*.

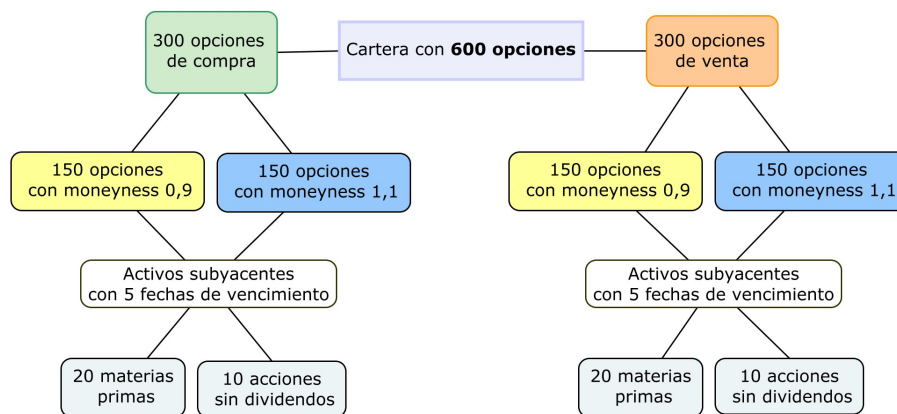
A continuación, en el Cuadro 4.2 se detallan las principales variables que componen cada opción financiera evaluada y las fuentes de datos de donde se han obtenido.

Cuadro 4.2: Resumen de las variables de cada opción financiera evaluada y su fuente, que es Yahoo Finance. Los datos de esta fuente han sido obtenidos a través de la librería `yfinance` en Python.

Variables	Fuente
Fecha de valoración: 12.01.2016	-
Fecha de vencimiento	Yahoo Finance
3 meses: 01.03.2017	Yahoo Finance
6 meses: 30.05.2017	Yahoo Finance
9 meses: 28.08.2017	Yahoo Finance
12 meses: 26.11.2017	Yahoo Finance
24 meses: 21.11.2018	Yahoo Finance
Precio del subyacente	Yahoo Finance
Precio ejercicio = Moneyness · Precio subyacente	-
Volatilidad anual = Volatilidad diaria $\cdot \sqrt{252}$	-
Tasa de interés libre de riesgo	Yahoo Finance

La Figura 4.1 muestra un esquema visual del conjunto de datos utilizado en este trabajo. La cartera total está compuesta por 600 opciones financieras, divididas en 300 opciones de compra y 300 opciones de venta. A su vez, cada grupo de opciones se subdivide según su moneyness, considerando 150 opciones con moneyness de 0,9 y 150 con moneyness de 1,1. Posteriormente, todas las opciones están asociadas a activos subyacentes, cada uno con cinco fechas de vencimiento diferentes. Dentro de este conjunto, los activos subyacentes incluyen 20 materias primas y 10 acciones que no pagan dividendos.

Figura 4.1: Esquema de la cartera de opciones utilizadas en el análisis. La cartera total consta de 600 opciones financieras, divididas en 300 opciones de compra y 300 opciones de venta. Cada grupo se descompone en opciones con moneyness de 0,9 y 1,1. Todas las opciones están vinculadas a activos subyacentes, que incluyen 20 materias primas y 10 acciones sin dividendos, con cinco fechas de vencimiento diferentes. Fuente: Elaboración propia.



Capítulo 5

Resultados

En este capítulo presentamos los resultados obtenidos a partir de la implementación en *python* de los métodos descritos en el Capítulo 3, aplicados al conjunto de datos seleccionado. Los resultados relativos a la estimación de las griegas, utilizando el modelo de Black and Scholes y los métodos de estimación (diferencias finitas, trayectorias y razón de verosimilitudes), pueden descargarse en un archivo en formato Excel al ejecutar el código del Apéndice C.

En primer lugar, se presentan los errores asociados a la estimación de las griegas para cada método, en comparación con los valores obtenidos mediante el modelo de Black and Scholes, el cual consideramos como la solución exacta. El objetivo es evaluar la precisión de cada método al aproximar las griegas en relación con los valores de las mismas obtenidos con el modelo de Black and Scholes. Para ello, utilizaremos la **raíz del error cuadrático medio** (RMSE, por sus siglas en inglés), ya que a diferencia de otras métricas de error permite obtener un resultado en las unidades originales de las variables estimadas. El RMSE nos permite cuantificar la desviación promedio entre los valores estimados y los valores reales, en nuestro caso dados por el modelo de Black and Scholes. De forma que, un valor de RMSE más bajo indica que las estimaciones están más cerca de los valores reales, lo que indica un mejor ajuste del modelo. Esta métrica de error se define de la siguiente manera:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5.1)$$

donde y_i representa los valores reales (los del modelo de Black and Scholes [1973]), \hat{y}_i representa los valores estimados y n es el número total de observaciones.

El Cuadro 5.1 presenta los valores del RMSE para cada una de las griegas calculadas con los distintos métodos de estimación. Se incluyen los RMSE tanto para opciones de compra como de venta, así como el RMSE general de ambos tipos de opciones, en todos los casos con un moneyness de 0.9. Por otro lado, el Cuadro 5.2 presenta los valores del RMSE para cada una de las griegas, calculadas mediante los distintos métodos de estimación. Se incluyen los RMSE tanto para opciones de compra como de venta, así como el RMSE general considerando ambos tipos de opciones, para un moneyness de 1.1.

Cuadro 5.1: RMSE para cada griega según el método utilizado para las opciones de compra, venta y en general con moneyness 0.9. En este contexto, DF representa el método de diferencias finitas, TY corresponde al método de Trayectoria, y RV hace referencia al método de razón de verosimilitudes. El subíndice c indica el caso de una opción de compra, el subíndice v corresponde a una opción de venta, y la ausencia de subíndice representa el valor total considerando ambas.

Método	$\hat{\delta}_c$	$\hat{\gamma}_c$	$\hat{\nu}_c$	$\hat{\theta}_c$	$\hat{\rho}_c$
DF	0.0056	3.7508	0.1942	0.8369	2.1763
TY	0.006	-	3.753	0.836	2.171
RV	0.008	0.003	1.727	2.276	123.095

Método	$\hat{\delta}_v$	$\hat{\gamma}_v$	$\hat{\nu}_v$	$\hat{\theta}_v$	$\hat{\rho}_v$
DF	0.0035	1.1078	0.1942	0.1932	2.1763
TY	0.004	-	1.546	0.192	2.58187
RV	0.002	0.003	2.866	0.434	44.274

Método	$\hat{\delta}$	$\hat{\gamma}$	$\hat{\nu}$	$\hat{\theta}$	$\hat{\rho}$
DF	0.0046	2.7655	0.1942	0.3753	2.1763
TY	0.003	-	1.106	0.192	2.171
RV	0.006	0.003	2.366	1.639	92.5

Cuadro 5.2: RMSE para cada griega según el método utilizado en opciones de compra, venta y en general con moneyness 1.1. En este contexto, DF representa el método de diferencias finitas, TY corresponde al método de Trayectoria, y RV hace referencia al método de razón de verosimilitudes. El subíndice c indica el caso de una opción de compra, el subíndice v corresponde a una opción de venta, y la ausencia de subíndice representa el valor total considerando ambas.

Método	$\hat{\delta}_c$	$\hat{\gamma}_c$	$\hat{\nu}_c$	$\hat{\theta}_c$	$\hat{\rho}_c$
DF	0.0064	2.4849	0.1677	0.4721	3.5929
TY	0.006	-	2.485	0.472	3.608
RV	0.004	0.002	1.306	1.091	66.966

Método	$\hat{\delta}_v$	$\hat{\gamma}_v$	$\hat{\nu}_v$	$\hat{\theta}_v$	$\hat{\rho}_v$
DF	0.0042	1.5465	0.1677	0.2427	3.5929
TY	0.004	-	1.546	0.242	3.608
RV	0.002	0.009	6.784	1.15	106.75

Método	$\hat{\delta}$	$\hat{\gamma}$	$\hat{\nu}$	$\hat{\theta}$	$\hat{\rho}$
DF	0.0054	2.0696	0.1677	0.3753	3.5929
TY	0.005	-	2.07	0.375	3.608
RV	0.003	0.006	4.885	1.121	89.107

En segundo lugar, presentamos los gráficos 5.1, 5.2, 5.3 y 5.4, donde se muestra la estimación de las griegas de opciones financieras para dos activos subyacentes de nuestra cartera, la materia prima Oro y la acción sin dividendos de Google utilizando el modelo de Black and Scholes, el método de diferencias finitas, el método de trayectorias y el método de razón de verosimilitudes.

Figura 5.1: Estimación de Delta para opciones sobre oro y Google con el modelo de Black and Scholes, mediante los métodos de diferencias finitas, trayectorias y razón de verosimilitudes, en función de la fecha de vencimiento.

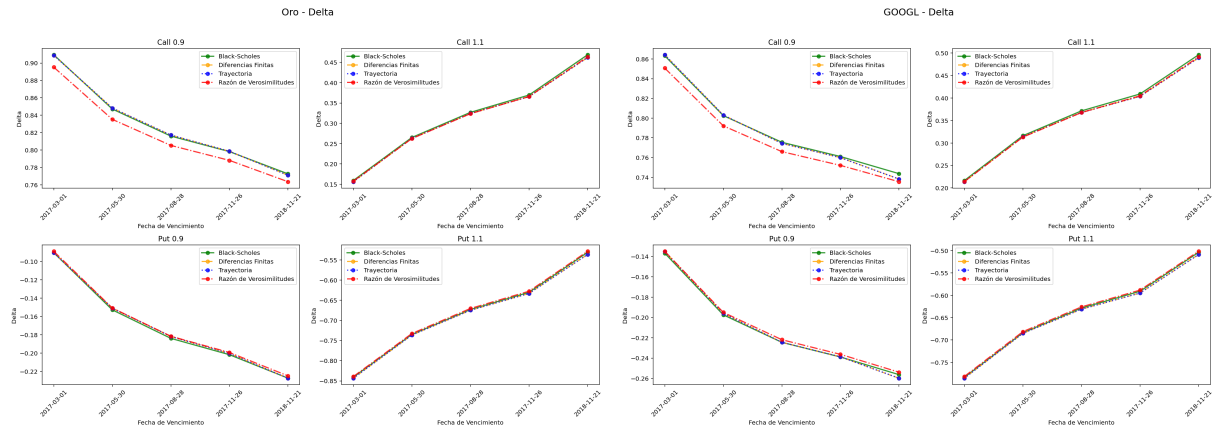


Figura 5.2: Estimación de Vega para opciones sobre oro y Google con el modelo de Black and Scholes, mediante los métodos de diferencias finitas, trayectorias y razón de verosimilitudes, en función de la fecha de vencimiento.

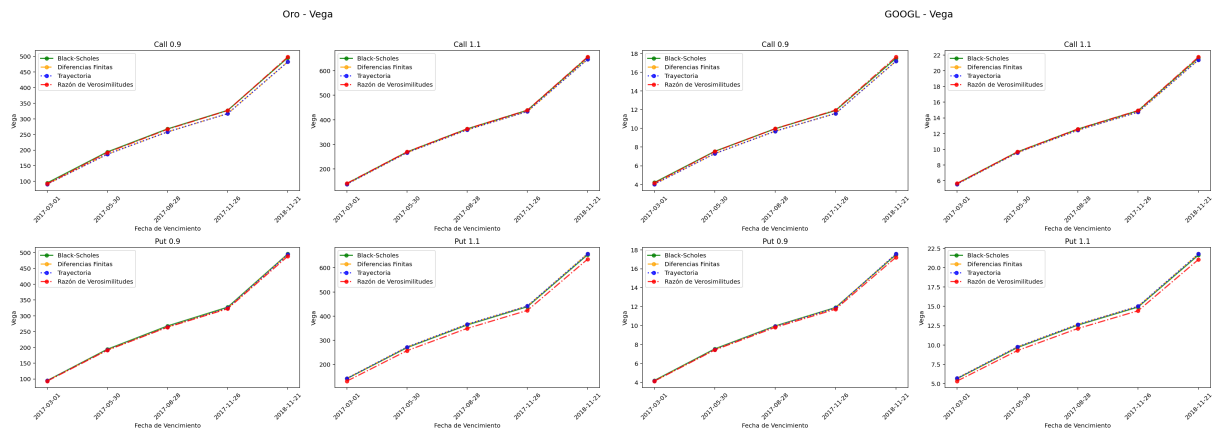


Figura 5.3: Estimación de Theta para opciones sobre oro y Google con el modelo de Black and Scholes, mediante los métodos de diferencias finitas, trayectorias y razón de verosimilitudes, en función de la fecha de vencimiento.

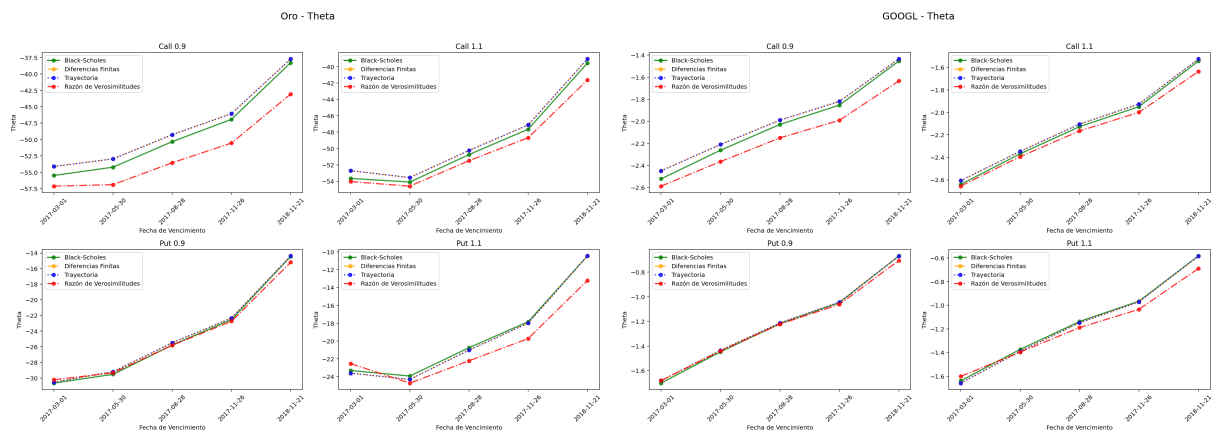
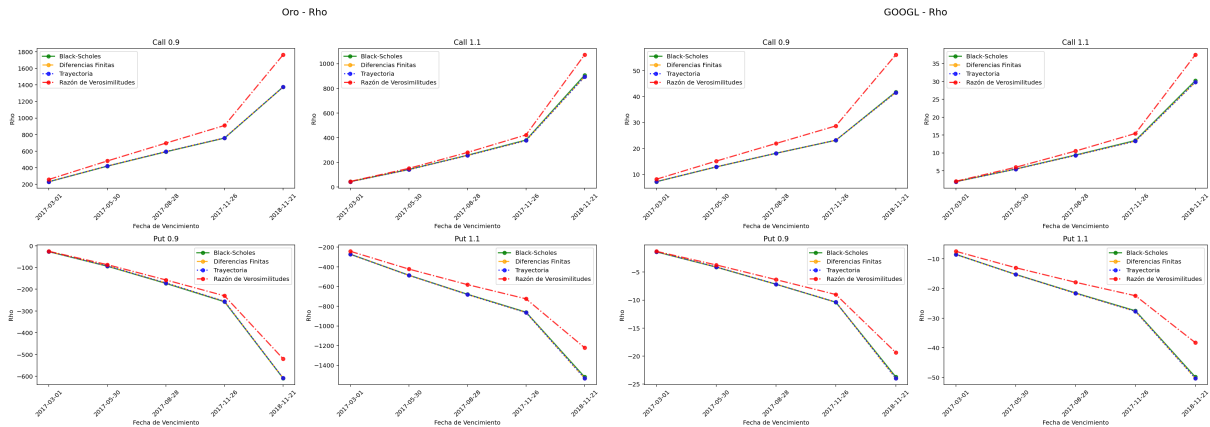


Figura 5.4: Estimación de Rho para opciones sobre oro y Google con el modelo de Black and Scholes, mediante los métodos de diferencias finitas, trayectorias y razón de verosimilitudes, en función de la fecha de vencimiento.



A continuación, se presentan las Figuras 5.5, 5.6, 5.7 y 5.8, que muestran cómo evoluciona el RMSE total para las diferentes griegas a medida que varía la fecha de vencimiento. El RMSE total se refiere al promedio del error combinado entre opciones de compra y venta, considerando ambos niveles de moneyness.

Figura 5.5: Evolución del RMSE total para la griega Delta según la fecha de vencimiento utilizando los métodos de diferencias finitas, Trayectoria y razón de verosimilitudes.

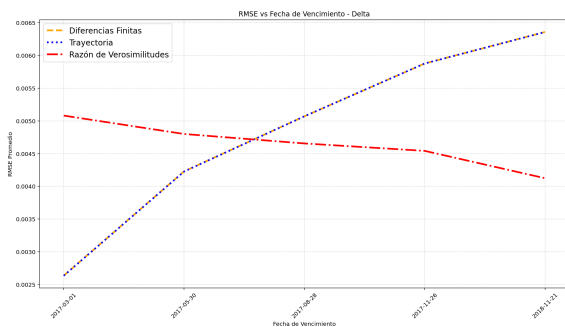


Figura 5.7: Evolución del RMSE total para la griega Theta según la fecha de vencimiento utilizando los métodos de diferencias finitas, trayectoria y razón de verosimilitudes.

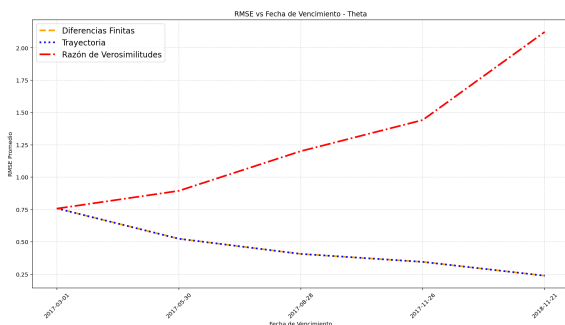


Figura 5.6: Evolución del RMSE total para la griega Vega según la fecha de vencimiento utilizando los métodos de diferencias finitas, trayectoria y razón de verosimilitudes.

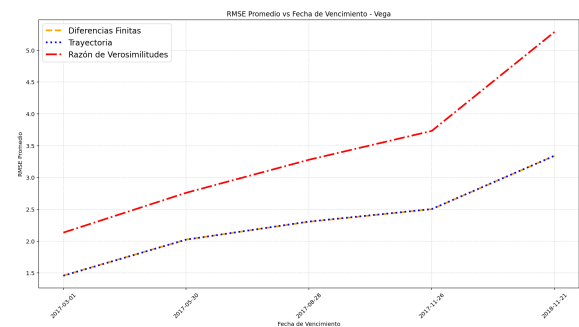
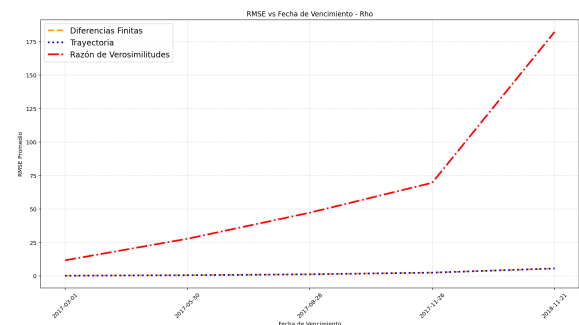


Figura 5.8: Evolución del RMSE total para la griega Rho según la fecha de vencimiento utilizando los métodos de diferencias finitas, trayectoria y razón de verosimilitudes.



Capítulo 6

Discusión

En este capítulo presentamos una discusión detallada sobre los resultados obtenidos al aplicar los métodos al conjunto de datos considerado, así como las diferencias de cada uno de los enfoques utilizados para estimar las griegas de las opciones de nuestra cartera.

En primer lugar, en las Figuras 5.1, 5.2, 5.3 y 5.4, observamos que el método de diferencias finitas y el de la trayectoria presentan estimaciones muy similares, casi indistinguibles, acercándose mucho a los valores obtenidos por el modelo de Black and Scholes. Sin embargo, el método de razón de verosimilitudes se muestra ligeramente más alejado en las estimaciones, tanto para la materia prima Oro como para las acciones de Google. Esta diferencia se mantiene a lo largo de todas las griegas estimadas. No obstante, cabe destacar que en la estimación de Rho, el método de razón de verosimilitudes presenta un RMSE mucho mayor en comparación con las otras dos metodologías, evidenciando una menor precisión en este caso particular.

En segundo lugar, al observar los valores del RMSE presentados en las Tablas 5.1 y 5.2, se puede concluir que, en términos generales, los métodos de diferencias finitas y trayectorias presentan errores relativamente bajos, lo que indica una buena aproximación de las griegas, especialmente cuando se utilizan en el contexto de opciones con moneyness cercano a 1, tanto para opciones de compra como de venta.

Por otro lado, en las Tablas 5.1 y 5.2 se observa nuevamente que el método de la razón de verosimilitudes muestra resultados con errores significativamente más altos en el valor de Rho. Este comportamiento podría estar relacionado con las limitaciones inherentes a la aproximación utilizada por este método, lo que indica que, aunque el método de razón de verosimilitudes puede ser útil en ciertos contextos, sus resultados no alcanzan el nivel de precisión ofrecido por el método de las diferencias finitas o el método de la trayectoria.

En tercer lugar, al analizar la evolución del RMSE según la fecha de vencimiento en las Figuras 5.5, 5.6, 5.7 y 5.8, se observan patrones interesantes. En el caso de Delta, el RMSE aumenta con la fecha de vencimiento para los métodos de trayectoria y diferencias finitas, mientras que en el caso de la razón de verosimilitudes, este disminuye, aunque los valores del aumento son muy pequeños, casi estadísticamente insignificantes.

En cuanto a la griega Vega destaca que el RMSE evoluciona de forma parecida para los tres métodos, aumenta el RMSE cuando la fecha de vencimiento es más lejana. Este comportamiento podría deberse a que, conforme pasa el tiempo, la volatilidad aumenta, lo que impacta la estimación de la Vega.

Por otro lado, para la Theta destaca un aumento considerable del RMSE con la razón de verosimilitudes, mientras que los métodos de trayectoria y diferencias finitas se mantienen estables, siendo estos mejores en este caso.

Por último, como ya se había observado en el análisis previo, la griega Rho presenta una diferencia notable, ya que el RMSE de la razón de verosimilitudes aumenta significativamente con la fecha de vencimiento, mientras que los métodos de trayectoria y diferencias finitas presentan un RMSE que a penas aumenta conforme avanza el tiempo. está estrechamente relacionado con la tasa de interés libre de riesgo. Al valorar las opciones, estamos considerando únicamente la tasa de interés libre de riesgo en la fecha de valoración, lo que puede influir en los resultados obtenidos para las estimaciones a largo plazo. En este análisis, al valorar las opciones, se ha utilizado únicamente la tasa libre de riesgo vigente en la fecha de valoración. Si bien esta aproximación es razonable para opciones de corto plazo, puede no ser adecuada cuando el horizonte temporal es mayor. En esos casos, sería más preciso emplear modelos que permitan anticipar la evolución futura de los tipos de interés.

Finalmente, los resultados obtenidos sugieren que, en general, los métodos empleados son adecuados para la estimación de las griegas, pero se debe prestar atención al tipo de opción, al tiempo hasta vencimiento o a la griega que se desee estimar, así como a las condiciones del mercado para elegir el método de estimación más adecuado.

Capítulo 7

Conclusión

En este trabajo se ha realizado un análisis comparativo entre diferentes métodos para la estimación de las sensibilidades de las opciones financieras, conocidas como las griegas, para el caso de opciones europeas. A lo largo del trabajo, se han evaluado tres métodos: el método de diferencias finitas, el método de trayectoria y el método de razón de verosimilitudes, aplicándolos a un conjunto de opciones reales y contrastando los resultados con el modelo de solución exacta de Black and Scholes.

El conjunto de datos utilizado en este análisis está compuesto por un total de 600 opciones financieras, distribuidas entre opciones de compra y de venta. De las 300 opciones de compra, 150 corresponden a un moneyness de 0,9 y las otras 150 a un moneyness de 1,1. De manera similar, también se dispone de 300 opciones de venta, igualmente distribuidas entre los moneyness 0,9 y 1,1. Este conjunto de datos incluye tanto opciones sobre materias primas como sobre acciones que no pagan dividendos. En concreto, para valorar una cartera compuesta por 20 opciones europeas sobre materias primas (incluyendo metales, energía y agricultura) y 10 opciones europeas sobre acciones sin dividendos, se consideran tanto las opciones de compra como las de venta para 5 vencimientos diferentes.

En este trabajo nos hemos centrado en el uso del método de Monte Carlo para simular los precios de los activos subyacentes. De forma que, a partir de las simulaciones obtenidas, hemos calculado los valores de las opciones financieras de la cartera. Posteriormente, utilizando estos resultados, hemos estimado las griegas correspondientes a cada opción mediante los métodos de estimación seleccionados.

La evaluación y comparación de los resultados obtenidos se ha apoyado en diversas herramientas, como gráficos de los valores de las griegas de cada activo subyacente por fecha de vencimiento para cada método de estimación, valores del RMSE según el grado de moneyness y gráficos de la evolución del RMSE del conjunto de activos subyacentes de cada griega según la fecha de vencimiento.

En general, los resultados obtenidos muestran que los métodos de diferencias finitas, y el de trayectoria presentan un rendimiento satisfactorio, con estimaciones muy cercanas a las obtenidas por el modelo de Black and Scholes. El método de razón de verosimilitudes también presentó buenos resultados, aunque su desempeño fue inferior en la estimación de Rho, deduciendo que este no es el más adecuado en este caso.

Los métodos analizados han demostrado ser útiles para obtener las griegas de ciertas opciones, pero también se han identificados áreas de mejora. Concretamente, se ha observado que los errores en las estimaciones tienden a incrementarse cuando se utilizan fechas de vencimiento más largas. Se observó que el RMSE para las estimaciones de Delta evolucionó de forma relativamente estable entre los métodos, mientras que para Vega, Theta y Rho, el método de razón de verosimilitudes presentó un aumento considerable en el RMSE, particularmente para Rho a medida que se acercaba la fecha de vencimiento. Esto sugiere una vez más que, aunque la razón de verosimilitudes puede ser efectiva en ciertos contextos, no es bueno para estimaciones precisas de Rho. En consecuencia, la elección de cada método debe depender de la griega que se desea estimar, el tipo de opción, así como el tiempo hasta su vencimiento y las condiciones del mercado.

Finalmente, una posible línea de investigación futura sería el refinamiento de los métodos de estimación y la exploración de nuevas aproximaciones que incrementen la precisión de los resultados. Un aspecto clave para mejorar los resultados sería aumentar el número de trayectorias utilizadas en las simulaciones mediante Monte Carlo de los precios de los activos subyacentes, lo cual podría reducir la variabilidad de las estimaciones. Asimismo, se podría investigar la mejora en la estimación de la volatilidad, especialmente mediante el uso de volatilidad implícita o modelos avanzados de volatilidad estocástica, que podrían contribuir a obtener estimaciones más precisas y alineadas con las condiciones reales del mercado. Las aplicaciones futuras de estos métodos podrían centrarse no solo en el análisis de opciones europeas, sino también en la extensión de los mismos a otros tipos de derivados financieros más complejos, como las opciones exóticas o las opciones en mercados altamente volátiles.

En conclusión, la estimación precisa de las griegas es fundamental en finanzas, ya que influye directamente en la gestión del riesgo, la toma de decisiones estratégicas y la valoración de opciones financieras. Estas permiten realizar coberturas eficientes, optimizar carteras y valorar correctamente las posiciones en derivados, lo que es esencial para la protección ante movimientos inesperados del mercado. De este modo, una mala estimación puede conducir a pérdidas significativas y errores en la implementación de estrategias de inversión. Por ello, es crucial seguir perfeccionando los métodos de estimación de las griegas, lo cual permitirá garantizar una mayor precisión en la gestión del riesgo y una mejor aproximación a las condiciones reales del mercado.

Referencias

- F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Polit. Econ.*, 81(3):637–654, May 1973.
- G. E. P. Box and M. E. Muller. A note on the generation of random normal deviates. *Ann. Math. Stat.*, 29(2):610–611, June 1958.
- P. P. Boyle. Options: A monte carlo approach. *J. Financ. Econ.*, 1977.
- R. F. Bruner, K. M. Eades, R. S. Harris, and R. C. Higgins. Best practices in estimating the cost of capital: survey and synthesis. *Financial practice and education*, 8:13–28, 1998.
- I. J. Clark. *Commodity option pricing: A practitioner's guide*. Wiley, Jan. 2014.
- P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Stochastic Modelling and Applied Probability. Springer, New York, NY, Oct. 2010.
- J. Hull and A. White. The pricing of options on assets with stochastic volatilities. *J. Finance*, 42(2):281–300, June 1987.
- G. Marsaglia and T. A. Bray. A convenient method for generating normal variables. *SIAM Rev. Soc. Ind. Appl. Math.*, 6(3):260–264, July 1964.
- G. Marsaglia and W. W. Tsang. The ziggurat method for generating random variables. *Journal of Statistical Software*, 5:1–7, Oct. 2000.
- B. McDonald. *Derivatives markets*. Pearson, Upper Saddle River, NJ, 3 edition, Dec. 2012.
- R. C. Merton. Option pricing when underlying stock returns are discontinuous. *J. Financ. Econ.*, 3(1-2):125–144, Jan. 1976.
- S. E. Shreve. *Stochastic calculus for finance II: Continuous-time models*. Springer Finance. Springer, New York, NY, 1 edition, May 2004.

A. Cálculo estocástico

En esta sección tratamos una serie de conceptos básicos de procesos estocásticos importantes para nuestro trabajo. El desarrollo de esta sección se ha llevado a cabo siguiendo los libros de Shreve [2004] y Glasserman [2010].

Definición 7.0.1 (Proceso de Itô). Un proceso $\{X(t), 0 \leq t \leq T\}$ con valores en \mathbb{R}^d es un proceso de Itô si puede ser representado como

$$X(t) = X(0) + \int_0^t a(u) du + \int_0^t b(u) dW(u), \quad 0 \leq t \leq T, \quad (7.1)$$

donde $X(0)$ es \mathcal{F}_0 -medible, a es un proceso adaptado con valores en \mathbb{R}^d que satisface

$$P\left(\int_0^T |a_i(t)| dt < \infty\right) = 1, \quad i = 1, \dots, d, \quad (7.2)$$

y b es un proceso adaptado con valores en $\mathbb{R}^{d \times k}$ que satisface

$$P\left(\int_0^T \|b(u)\|^2 dt < \infty\right) = 1 \quad (7.3)$$

Podemos reescribir la ecuación 7.1 como:

$$dX(t) = a(t) dt + b(t) dW(t) \quad (7.4)$$

Definición 7.0.2 (Ecuación Diferencial Estocástica). Una Ecuación Diferencial Estocástica (EDE) es una ecuación de la forma

$$dX(t) = a(X(t), t)dt + b(X(t), t)dW(t), \quad X(0) = X_0 \quad (7.5)$$

donde W es un movimiento browniano de k dimensiones, a es un mapeo de $\mathbb{R}^d \times [0, \infty)$ a \mathbb{R}^d (drift o tendencia), b es un mapeo de $\mathbb{R}^d \times [0, \infty)$ a $\mathbb{R}^{d \times k}$ (diffusion), y X_0 es un vector aleatorio de dimensión d independiente de W .

Una solución a (7.5) en un intervalo $[0, T]$ es un proceso de Itô $\{X(t), 0 \leq t \leq T\}$ para el cual $P(X(0) = X_0) = 1$ y

$$X(t) = X(0) + \int_0^t a(X(u), u)du + \int_0^t b(X(u), u)dW(u), \quad 0 \leq t \leq T \quad (7.6)$$

El requisito de que X sea un proceso de Itô impone condiciones 7.2 y 7.3 sobre los procesos $\{a(X(t), t), 0 \leq t \leq T\}$ y $\{b(X(t), t), 0 \leq t \leq T\}$. Ahora enunciamos el resultado principal sobre soluciones fuertes a las EDEs.

Definición 7.0.3 (Movimiento browniano o proceso de Wiener). Un movimiento browniano estándar unidimensional en $[0, T]$ es un proceso estocástico $\{W(t), 0 \leq t \leq T\}$ que satisface las siguientes propiedades:

- i. $W(0) = 0$.
- ii. La función $t \mapsto W(t)$ es, con probabilidad 1, continua en $[0, T]$.
- iii. Los incrementos $\{W(t_1) - W(t_0), W(t_2) - W(t_1), \dots, W(t_k) - W(t_{k-1})\}$ son independientes para cualquier k y cualquier elección de $0 \leq t_0 < t_1 < \dots < t_k \leq T$.
- iv. Para cualquier $0 \leq s < t \leq T$, $W(t) - W(s) \sim N(0, t - s)$, es decir, los incrementos son normalmente distribuidos con media 0 y varianza $t - s$.

Como consecuencia de (i) y (iv), se tiene que:

$$W(t) \sim N(0, t) \quad 0 < t \leq T \quad (7.7)$$

Para constantes $\mu, \sigma > 0$, llamamos a un proceso $X(t)$ un movimiento browniano con *tendencia* μ y *coeficiente de difusión* σ^2 , es decir, $X \sim \text{BM}(\mu, \sigma^2)$ si se tiene que:

$$\frac{X(t) - \mu t}{\sigma} \quad (7.8)$$

es un movimiento browniano estándar. Así, podemos construir X a partir de un movimiento browniano estándar W mediante la fórmula:

$$X(t) = \mu t + \sigma W(t) \quad (7.9)$$

Por lo tanto, se tiene que $X(t) \sim N(\mu t, \sigma^2 t)$ y que X resuelve la EDE:

$$dX(t) = \mu dt + \sigma dW(t) \quad (7.10)$$

Estamos suponiendo que $X(0) = 0$, pero podemos construir un movimiento browniano con parámetros μ y σ^2 y valor inicial x simplemente sumando x a cada $X(t)$. De modo que, para $\mu(t)$ determinista y dependiente del tiempo, y $\sigma(t) > 0$, definimos un movimiento browniano con *tendencia* $\mu(t)$ y *coeficiente de difusión* $\sigma^2(t)$ mediante la EDE:

$$dX(t) = \mu(t) dt + \sigma(t) dW(t), \quad (7.11)$$

es decir,

$$X(t) = X(0) + \int_0^t \mu(s) ds + \int_0^t \sigma(s) dW(s), \quad (7.12)$$

donde $X(0)$ es una constante arbitraria. Este proceso X tiene trayectorias continuas y sus incrementos son independientes. Cada incremento $X(t) - X(s)$ es normalmente distribuido con media:

$$\mathbb{E}[X(t) - X(s)] = \int_s^t \mu(u) du \quad (7.13)$$

y varianza:

$$\text{Var}[X(t) - X(s)] = \int_s^t \sigma^2(u) du \quad (7.14)$$

Definición 7.0.4 (Movimiento Browniano Geométrico (GBM)). Sean $\mu \in \mathbb{R}$, $\sigma > 0$ y $S_0 > 0$, se define el movimiento browniano geométrico (GBM) como el proceso estocástico a tiempo continuo $\{S_t : t \geq 0\}$ que satisface la ecuación diferencial estocástica:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \quad (7.15)$$

y puede ser escrito como:

$$S_t = S_0 \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W_t \right), \quad t \geq 0 \quad (7.16)$$

donde W_t es el movimiento Browniano estándar.

Un proceso estocástico $S(t)$ es un GBM si $\log S(t)$ es un movimiento browniano con valor inicial $\log S(0)$, es decir, un GBM es simplemente un movimiento browniano estándar elevado a la exponencial.

El GBM es el modelo más importante para describir el valor de un activo financiero. A diferencia del movimiento browniano ordinario, que puede tomar valores negativos, el GBM siempre es **positivo**, ya que la función exponencial siempre toma valores positivos. Además, para el GBM, los cambios porcentuales dados por:

$$\frac{S(t_2) - S(t_1)}{S(t_1)}, \quad \frac{S(t_3) - S(t_2)}{S(t_2)}, \quad \dots, \quad \frac{S(t_n) - S(t_{n-1})}{S(t_{n-1})} \quad (7.17)$$

son independientes para $t_1 < t_2 < \dots < t_n$, en lugar de los cambios absolutos $S(t_i + 1) - S(t_i)$. De forma que, estas propiedades son las razones fundamentales por las que el GBM se utiliza tanto en la modelización de los precios de los activos.

Supongamos que W es un movimiento browniano estándar y X satisface la ecuación 7.11, entonces $X \sim \text{BM}(\mu, \sigma^2)$. Si definimos $S(t) = S(0) \exp(X(t))$, entonces aplicando la Fórmula de Itô se tiene que:

$$dS(t) = S(t) (\mu dt + \sigma dW(t)) \quad (7.18)$$

De forma que, un proceso de GBM se especifica mediante una EDE de la forma:

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW(t) \quad (7.19)$$

donde $S \sim \text{GBM}(\mu, \sigma^2)$.

Observamos que las ecuaciones anteriores presentan una ambigüedad en el papel de μ . En la ecuación 7.11, μ es la tendencia del movimiento browniano que hemos exponenciado para definir $S(t)$, es decir, la tendencia de $\log S(t)$, mientras que en la ecuación 7.18, $S(t)$ tiene tendencia $\mu S(t)$, lo que implica que:

$$d \log S(t) = \left(\mu - \frac{\sigma^2}{2} \right) dt + \sigma dW(t) \quad (7.20)$$

donde μ es el parámetro de desplazamiento, aunque no sea el desplazamiento de $S(t)$ ni de $\log S(t)$; σ es el parámetro de volatilidad de $S(t)$; y el coeficiente de difusión de $S(t)$ es $\sigma^2 S^2(t)$. En consecuencia, si $S \sim \text{GBM}(\mu, \sigma^2)$ y S tiene valor inicial $S(0)$, entonces a partir de (7.20) se tiene que:

$$S(t) = S(0) \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W(t) \right) \quad (7.21)$$

B. Código de Python

En esta sección se presenta el código de *python* completo para llevar a cabo las simulaciones, cálculos y análisis desarrollados a lo largo del trabajo. El código incluye explicaciones y comentarios de cada bloque, entre ellos la obtención de datos históricos desde Yahoo Finance, la implementación de los métodos de estimación de las griegas, la simulación Monte Carlo para la valoración de opciones, y la visualización de los resultados. Este código permite reproducir todos los resultados presentados en el trabajo.

El objetivo es estimar el valor de las griegas asociadas a una cartera de opciones financieras mediante los tres métodos de estimación descritos en el Capítulo 3. En primer lugar, se calculan los valores de las griegas utilizando el modelo de Black and Scholes, el cual proporciona una solución exacta y será considerado como la referencia para evaluar la precisión de los métodos empleados. A continuación, se obtienen los precios de las opciones mediante la simulación de Monte Carlo, siguiendo el procedimiento descrito en el Algoritmo 1 empleando 25.000 trayectorias. Finalmente, a partir de los precios simulados con Monte Carlo, estimamos las griegas utilizando los tres métodos seleccionados.

Apéndice código de python

1 Introducción

En este código en **Python** se implementan los métodos numéricos descritos en el Capítulo 3 para la estimación de las **sensibilidades del precio de una opción financiera**, conocidas como las “**griegas**”. Estas sensibilidades son fundamentales en la gestión de riesgos financieros, ya que permiten evaluar cómo varía el precio de una opción ante cambios en parámetros clave como el precio del activo subyacente, la volatilidad, el tiempo hasta el vencimiento y la tasa de interés.

Se han implementado tres métodos de estimación:

1. **Método de Diferencias Finitas**
2. **Método de la Trayectoria**
3. **Método de Razón de Verosimilitudes**

El código emplea la **simulación Monte Carlo**, una técnica ampliamente utilizada en la valoración de opciones. Se genera una gran cantidad de trayectorias aleatorias para el precio del activo subyacente y, a partir de ellas, se calculan las sensibilidades de opciones europeas.

Además se ha implementado el **modelo de Black-Scholes**, el cual proporciona fórmulas analíticas exactas para las “griegas”, lo que permite comparar la precisión de los métodos numéricos con los valores teóricos.

Finalmente, los resultados obtenidos pueden exportarse en formato Excel al final del código. Cabe destacar que el tiempo de ejecución puede ser considerable, debido a la alta cantidad de simulaciones necesarias para lograr estimaciones precisas.

1.1 Datos

En este apartado, se presentan los datos obtenidos de **Yahoo Finance** para valorar una cartera de opciones europeas sobre materias primas y acciones sin dividendos. Se recopilan datos históricos de precios y volatilidad de activos financieros a partir de sus símbolos en Yahoo Finance (denominados “tickers”). Nuestra cartera está compuesta por **20 opciones europeas sobre materias primas** y **10 opciones europeas sobre acciones sin dividendos**, tanto de compra como de venta, para 5 vencimientos diferentes.

```
[ ]: import yfinance as yf      # Descarga de datos financieros históricos
import numpy as np             # Cálculos numéricos y manejo de arrays
import pandas as pd            # Manipulación y análisis de datos
import datetime                # Manejo de fechas y tiempos
```

```
import math # Funciones matemáticas estándar
import scipy.stats as si # Distribuciones estadísticas y funciones especiales
```

```
[ ]: def obtener_precio_volatilidad(ticker, fecha_valoracion):

    """
    Esta función obtiene el último precio de cierre, la volatilidad anualizada,
    y la fecha de la última observación
    para un activo financiero utilizando datos históricos de Yahoo Finance.

    Parámetros:
    - ticker: Símbolo del activo en Yahoo Finance.
    - fecha_valoracion: Fecha límite hasta la cual se descargarán los datos
    históricos (formato 'YYYY-MM-DD').

    Resultados:
    - ultimo_precio: último precio de cierre del activo financiero (float).
    - volatilidad_anualizada: volatilidad histórica anualizada.
    - ultima_fecha: Fecha de la última observación disponible en los datos
    históricos (str, formato 'YYYY-MM-DD').
    """

    # Descargar datos históricos desde Yahoo Finance
    data = yf.Ticker(ticker)
    historial = data.history(start="2016-01-01", end=fecha_valoracion)

    if historial.empty:
        print(f"No se encontraron datos para {ticker}.")
        return None, None, None

    # Eliminar filas con valores nulos en los precios de cierre
    historial = historial.dropna(subset=["Close"])

    # Último precio de cierre
    ultimo_precio = round(historial["Close"].iloc[-1], 3)
    ultima_fecha = historial.index[-1].strftime('%Y-%m-%d')

    # Calcular los retornos logarítmicos (u_i)
    retornos_log = np.log(historial["Close"] / historial["Close"].shift(1))

    # Eliminar valores nulos en los retornos logarítmicos (por si hay NaN en la
    primera observación)
    retornos_log = retornos_log.dropna()

    # Desviación estándar de los retornos logarítmicos (s)
    desviacion_estandar = retornos_log.std()
```

```

# Calcular el número de observaciones
n = len(retornos_log)

# Estimar la longitud del tiempo en años (suponemos que la serie histórica
↪ cubre un período de días laborales)
tau = n / 252 # Suponiendo 252 días laborales por año

# Volatilidad diaria (ajustada por la longitud del tiempo)
volatilidad_diaria = desviacion_estandar / np.sqrt(tau)

# Volatilidad anualizada (ajustada por 252 días de negociación)
volatilidad_anualizada = round(volatilidad_diaria * np.sqrt(252), 3)

return ultimo_precio, volatilidad_anualizada, ultima_fecha

```

```

[ ]: def calcular_precio_ejercicio(precio_subyacente, moneyness):

    """
    Esta función calcula el precio de ejercicio de una opción financiera basado
    ↪ en el nivel de moneyness.

    Parámetros:
        - precio_subyacente: Precio actual del activo subyacente (float).
        - moneyness: Relación entre el precio de ejercicio y el precio del activo
    ↪ subyacente (float).

    Resultado:
        - Precio de ejercicio de la opción redondeado a tres decimales (float).
    """

    return round(precio_subyacente * moneyness, 3)

```

```

[ ]: # Lista de materias primas o commodities
commodities = {
    "Oro": "GC=F",
    "Plata": "SI=F",
    "Cobre": "HG=F",
    "Platino": "PL=F",
    "Paladio": "PA=F",
    "Petróleo Crudo WTI": "CL=F",
    "Petróleo Crudo Brent": "BZ=F",
    "Gas Natural": "NG=F",
    "Gasolina RBOB": "RB=F",
    "Fuel Oil": "HO=F",
    "Maíz": "ZC=F",
    "Trigo": "ZW=F",
    "Café": "KC=F",
}

```

```

    "Azúcar": "SB=F",
    "Algodón": "CT=F",
    "Soja": "ZS=F",
    "Avena": "ZO=F",
    "Arroz": "ZR=F",
    "Cacao": "CC=F",
    "Zumo de Naranja": "OJ=F"
}

```

```

# Lista de acciones
acciones = [
    'AAPL', # Apple
    'GOOGL', # Google
    'AMZN', # Amazon
    'MSFT', # Microsoft
    'TSLA', # Tesla
    'NFLX', # Netflix
    'META', # Meta
    'NOW', # ServiceNow
    'ADBE', # Adobe
    'SHOP', # Shopify
    'TEAM', # Atlassian
    'CRM', # Salesforce
]

```

```

[ ]: # Fecha de valoración
fecha_valoracion = '2016-12-01'

```

```

[ ]: # Obtener datos de commodities
resultados_commodities = {}
for commodity, ticker in commodities.items():
    ultimo_precio, volatilidad_anualizada, ultima_fecha = 
    ↪obtener_precio_volatilidad(ticker, fecha_valoracion)
    if ultimo_precio is not None:
        resultados_commodities[commodity] = {
            'ultimo_precio': ultimo_precio,
            'volatilidad': volatilidad_anualizada,
            'ultima_fecha': ultima_fecha
        }

# Obtener datos de acciones
resultados_acciones = {}
for accion in acciones:
    ultimo_precio, volatilidad_anualizada, ultima_fecha = 
    ↪obtener_precio_volatilidad(accion, fecha_valoracion)
    if ultimo_precio is not None:
        resultados_acciones[accion] = {

```



```

        'ultimo_precio': ultimo_precio,
        'volatilidad': volatilidad_anualizada,
        'ultima_fecha': ultima_fecha
    }

# Filtrar acciones sin dividendos
def obtener_acciones_sin_dividendos(acciones):

    """
    Esta función filtra una lista de acciones y devuelve aquellas que no han
    ↪ pagado dividendos.

    Parámetros:
        - acciones: lista de símbolos bursátiles (tickers) de las acciones a
        ↪ analizar.

    Resultado:
        - acciones_sin_dividendos: lista con los símbolos bursátiles de las
        ↪ acciones que no han pagado dividendos en el período analizado.
    """

    acciones_sin_dividendos = []
    for accion in acciones:
        ticker = yf.Ticker(accion)
        historial = ticker.history(start="2016-01-01", end="2019-10-01")
        if 'Dividends' in historial.columns and historial['Dividends'].sum() == 0:
            acciones_sin_dividendos.append(accion)
    return acciones_sin_dividendos

acciones_sin_dividendos = obtener_acciones_sin_dividendos(acciones)

```

```

[ ]: # Fechas de vencimiento (3, 6, 9, 12, 24 meses)
fecha_limite_vencimiento = datetime.datetime(2025, 1, 1)
fechas_vencimiento = [
    (datetime.datetime.strptime(fecha_valoracion, "%Y-%m-%d") + datetime.
    ↪ timedelta(days=30*meses)).strftime("%Y-%m-%d")
    for meses in [3, 6, 9, 12, 24]
]
fechas_vencimiento = [
    min(datetime.datetime.strptime(fecha, "%Y-%m-%d"),
    ↪ fecha_limite_vencimiento).strftime("%Y-%m-%d")
    for fecha in fechas_vencimiento
]

```

```

[ ]: # Construcción del DataFrame
datos = []

```

```

# Agregar commodities
for commodity, valores in resultados_commodities.items():
    for fecha_vencimiento in fechas_vencimiento:
        dias_hasta_vencimiento = (datetime.datetime.strptime(fecha_vencimiento, "%Y-%m-%d") - datetime.datetime.strptime(fecha_valoracion, "%Y-%m-%d")).days
        anos_hasta_vencimiento = dias_hasta_vencimiento / 365
        precio_ejercicio_0_9 =
        calcular_precio_ejercicio(valores['ultimo_precio'], 0.9)
        precio_ejercicio_1_1 =
        calcular_precio_ejercicio(valores['ultimo_precio'], 1.1)

        datos.append({
            'Fecha Valoración': fecha_valoracion,
            'Fecha Vencimiento': fecha_vencimiento,
            'Años hasta vencimiento': anos_hasta_vencimiento,
            'Activo': commodity,
            'Tipo': 'Commodity',
            'Moneyness 0.9': 0.9,
            'Precio Subyacente': valores['ultimo_precio'],
            'Precio Ejercicio 0.9': precio_ejercicio_0_9,
            'Moneyness 1.1': 1.1,
            'Precio Ejercicio 1.1': precio_ejercicio_1_1,
            'Volatilidad Anual': valores['volatilidad'],
        })

# Agregar acciones sin dividendos
for accion, valores in resultados_acciones.items():
    if accion in acciones_sin_dividendos:
        for fecha_vencimiento in fechas_vencimiento:
            dias_hasta_vencimiento = (datetime.datetime.strptime(fecha_vencimiento, "%Y-%m-%d") - datetime.datetime.strptime(fecha_valoracion, "%Y-%m-%d")).days
            anos_hasta_vencimiento = dias_hasta_vencimiento / 365
            precio_ejercicio_0_9 =
            calcular_precio_ejercicio(valores['ultimo_precio'], 0.9)
            precio_ejercicio_1_1 =
            calcular_precio_ejercicio(valores['ultimo_precio'], 1.1)

            datos.append({
                'Fecha Valoración': fecha_valoracion,
                'Fecha Vencimiento': fecha_vencimiento,
                'Años hasta vencimiento': anos_hasta_vencimiento,
                'Activo': accion, # Aquí corregimos para que se muestre el
                nombre de la acción
                'Tipo': 'Acción',
            })

```

```

        'Moneyness 0.9': 0.9,
        'Precio Subyacente': valores['ultimo_precio'],
        'Precio Ejercicio 0.9': precio_ejercicio_0_9,
        'Moneyness 1.1': 1.1,
        'Precio Ejercicio 1.1': precio_ejercicio_1_1,
        'Volatilidad Anual': valores['volatilidad'],
    })

Datos = pd.DataFrame(datos)
#print(Datos)

# Exportar datos a un archivo Excel
#Datos.to_excel("Datos.xlsx", index=False)
#from google.colab import files
#files.download('Datos.xlsx')

```

2 Black-Scholes

Las fórmulas son:

$$V_c(S, t) = S_0 N(d_1) - K e^{-rT} N(d_2)$$

$$V_p(S, t) = K e^{-rT} N(-d_2) - S_0 N(-d_1)$$

donde

$$d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

```

[ ]: def calcular_precio_griegas(S, K, r, sigma, T):

    """
    Esta función calcula el precio y las griegas de una opción europea de compra (call) y venta (put) usando el modelo de Black-Scholes.

    Parámetros:
    - S: precio actual del activo subyacente.
    - K: precio de ejercicio de la opción.
    - r: tasa libre de riesgo.
    - sigma: volatilidad del activo subyacente.
    - T: tiempo hasta el vencimiento de la opción (en años).

    Resultado:
    """

```

```

- Precio de la opción de compra y venta.
- Delta, Gamma, Vega, Theta y Rho para opciones de compra y venta.
"""

if T <= 0 or sigma <= 0:
    return [None] * 12 # Evita cálculos inválidos

# Cálculo de d1 y d2
d1 = (math.log(S / K) + (r + 0.5 * sigma**2) * T) / (sigma * math.sqrt(T))
d2 = d1 - sigma * math.sqrt(T)

# Funciones de distribución acumulada normal estándar
N_d1 = si.norm.cdf(d1)
N_d2 = si.norm.cdf(d2)
N_neg_d1 = si.norm.cdf(-d1)
N_neg_d2 = si.norm.cdf(-d2)

# Precios de las opciones de compra (call) y venta (put)
call_price = S * N_d1 - K * math.exp(-r * T) * N_d2
put_price = K * math.exp(-r * T) * N_neg_d2 - S * N_neg_d1

# Delta: Sensibilidad con respecto al precio del activo subyacente
deltaCall = N_d1
deltaPut = N_d1 - 1

# Gamma: Sensibilidad de Delta con respecto al precio del activo subyacente
gamma = si.norm.pdf(d1) / (S * sigma * math.sqrt(T))

# Vega: Sensibilidad con respecto a la volatilidad
vega = S * si.norm.pdf(d1) * math.sqrt(T)

# Theta: Sensibilidad con respecto al tiempo
thetaCall = (-S * si.norm.pdf(d1) * sigma) / (2 * math.sqrt(T)) - r * K * math.exp(-r * T) * N_d2
thetaPut = (-S * si.norm.pdf(d1) * sigma) / (2 * math.sqrt(T)) + r * K * math.exp(-r * T) * N_neg_d2

# Rho: Sensibilidad con respecto a la tasa de interés
rhoCall = K * T * math.exp(-r * T) * N_d2
rhoPut = -K * T * math.exp(-r * T) * N_neg_d2

return call_price, put_price, deltaCall, gamma, vega, rhoCall, thetaCall, deltaPut, gamma, vega, rhoPut, thetaPut

```

```

[ ]: # Tasa libre de riesgo para el 30 de noviembre de 2016 en EE.UU.
r = 0.0237

```

```

resultados_BS = []

for _, row in Datos.iterrows():
    S = row['Precio Subyacente']
    sigma = row['Volatilidad Anual']
    T = row['Años hasta vencimiento']

    # Crear diccionario para almacenar los resultados de una fila
    resultado = {
        'Activo': row['Activo'],
        'Fecha Valoración': row['Fecha Valoración'],
        'Fecha Vencimiento': row['Fecha Vencimiento'],
        'Precio Subyacente': S
    }

    # Calcular precios y griegas para moneyness 0.9 y 1.1
    for m in [0.9, 1.1]:
        K = row[f'Precio Ejercicio {m}']
        valores = calcular_precio_griegas(S, K, r, sigma, T)

        # Guardar resultados para moneyness 0.9 y 1.1 en la misma fila
        resultado[f'Moneyness {m}'] = m
        resultado[f'Precio Call {m}'] = valores[0]
        resultado[f'Precio Put {m}'] = valores[1]
        resultado[f'Delta Call {m}'] = valores[2]
        resultado[f'Delta Put {m}'] = valores[7]
        resultado[f'Gamma Call {m}'] = valores[3]
        resultado[f'Gamma Put {m}'] = valores[3] # Gamma Call y Gamma Put son
↪ iguales
        resultado[f'Vega Call {m}'] = valores[4]
        resultado[f'Vega Put {m}'] = valores[4] # Vega Call y Vega Put son
↪ iguales
        resultado[f'Rho Call {m}'] = valores[5]
        resultado[f'Rho Put {m}'] = valores[10]
        resultado[f'Theta Call {m}'] = valores[6]
        resultado[f'Theta Put {m}'] = valores[11]

    resultados_BS.append(resultado)

Resultados_BS = pd.DataFrame(resultados_BS)
#print(Resultados_BS.head())

# Exportar resultados a un archivo Excel
#Resultados_BS.to_excel("Resultados_BS.xlsx", index=False)
#from google.colab import files
#files.download('Resultados_BS.xlsx')

```

3 Simulación Monte Carlo

En este apartado, implementaremos un modelo basado en el método de Monte Carlo para valorar opciones financieras europeas.

El proceso consiste en simular múltiples trayectorias del precio del activo subyacente utilizando un modelo de Movimiento Browniano Geométrico (GBM). Este enfoque se basa en generar un gran número de trayectorias aleatorias para el precio del activo subyacente y, a partir de ellas, estimar el valor esperado de la opción.

```
[ ]: def GBM_fd(S, r, sigma, T, trayectorias, pasosTiempo):  
  
    """  
    Esta función genera múltiples trayectorias del precio de un activo subyacente,  
    ↪ de una opción europea utilizando (Movimiento Browniano Geométrico (GBM)).  
  
    Parámetros:  
    - S: precio actual del activo subyacente.  
    - r: tasa libre de riesgo.  
    - sigma: volatilidad del activo subyacente.  
    - T: tiempo hasta el vencimiento de la opción (en años).  
    - trayectorias: número de trayectorias simuladas.  
    - pasosTiempo: número de pasos de tiempo en la simulación.  
  
    Resultado:  
    - Matriz ST, donde cada fila representa una trayectoria del precio del,  
    ↪ activo a lo largo del tiempo.  
    """  
  
    # Tamaño del paso  
    dt = T / pasosTiempo  
  
    # Semilla para resultados reproducibles  
    np.random.seed(3000)  
  
    # Inicialización de la matriz de trayectorias  
    ST = np.zeros((trayectorias, pasosTiempo + 1))  
    ST[:, 0] = S  
  
    # Simulación del Movimiento Browniano Geométrico  
    for t in range(1, pasosTiempo + 1):  
        Z = np.random.randn(trayectorias) # Generan números aleatorios normales.  
        dW = Z * np.sqrt(dt) # Incremento de Wiener  
        ST[:, t] = ST[:, t-1] * np.exp((r - 0.5 * sigma**2) * dt + sigma * dW)  
  
    return ST
```

```
[ ]: def monte_carlo(payload):

    """
    Esta función calcula el valor esperado de un conjunto de pagos simulados,
    utilizando el método de Monte Carlo.

    Parámetros:
        - payoff: Un array de valores de pago obtenidos a partir de simulaciones.

    Resultado:
        - Valor esperado del payoff, es decir, el precio estimado de la opción,
        basado en la simulación de Monte Carlo.
    """

    return np.mean(payload)
```

```
[ ]: # Parámetros de la simulación
r = 0.0237 # Tasa libre de riesgo
pasosTiempo = 365 # Número de pasos por año
trayectorias = 25000 # Número de trayectorias
```

```
[ ]: def calcular_valor_opcion(S, K, r, sigma, T, trayectorias, pasosTiempo):

    """
    Esta función calcula el valor de una opción europea (call o put) utilizando,
    el método de Monte Carlo.

    parámetros:
        - s: precio actual del activo subyacente.
        - k: precio de ejercicio de la opción.
        - r: tasa libre de riesgo.
        - sigma: volatilidad del activo subyacente.
        - t: tiempo hasta el vencimiento de la opción (en años).
        - trayectorias: número de simulaciones de Monte Carlo.
        - pasos_tiempo: número de pasos de tiempo en cada simulación.

    resultado:
        - valor_call: precio estimado de la opción call.
        - valor_put: precio estimado de la opción put.
    """

    ST = GBM_fd(S, r, sigma, T, trayectorias, pasosTiempo)[: , -1]

    # Calcular los payoffs de las opciones de compra (call) y venta (put)
    payoff_call = np.maximum(ST - K, 0)
    payoff_put = np.maximum(K - ST, 0)
```

```

# Calcular el valor de la opción descontado
valor_call = np.exp(-r * T) * monte_carlo(payload_call)
valor_put = np.exp(-r * T) * monte_carlo(payload_put)

return valor_call, valor_put

```

```

[ ]: resultados_MC = []

for _, row in Datos.iterrows():
    S = row['Precio Subyacente']
    volatilidad = row['Volatilidad Anual']
    T = row['Años hasta vencimiento']

    # Precios de ejercicio para moneyness 0.9 y 1.1
    K_0_9 = row['Precio Ejercicio 0.9']
    K_1_1 = row['Precio Ejercicio 1.1']

    # Calcular el valor de las opciones para cada moneyness
    valor_call_0_9, valor_put_0_9 = calcular_valor_opcion(S, K_0_9, r,
    ↪volatilidad, T, trayectorias, pasosTiempo)
    valor_call_1_1, valor_put_1_1 = calcular_valor_opcion(S, K_1_1, r,
    ↪volatilidad, T, trayectorias, pasosTiempo)

    resultados_MC.append({
        'Fecha Valoración': row['Fecha Valoración'],
        'Fecha Vencimiento': row['Fecha Vencimiento'],
        'Activo': row['Activo'],
        'Moneyness 0.9': 0.9,
        'Precio Call 0.9': valor_call_0_9,
        'Precio Put 0.9': valor_put_0_9,
        'Moneyness 1.1': 1.1,
        'Precio Call 1.1': valor_call_1_1,
        'Precio Put 1.1': valor_put_1_1
    })

Resultados_MC = pd.DataFrame(resultados_MC)
#print(Resultados_MC.head())

# Exportar resultados a un archivo Excel
Resultados_MC.to_excel("Resultados_MC.xlsx", index=False)
from google.colab import files
files.download('Resultados_MC.xlsx')

```

```

[ ]: import matplotlib.pyplot as plt

# Generación de trayectorias de precios del activo subyacente
ST_trayectoria = GBM_fd(S, r, sigma, T, trayectorias, pasosTiempo)

```



```

# Gráfica de las trayectorias de los precios del activo subyacente
plt.figure(figsize=(10,6))
for i in range(50): # Mostrar solo 50 trayectorias para mayor claridad
    plt.plot(ST_trayectoria[i, :], alpha = 0.7)
plt.title("Simulaciones de las Trayectorias de Precios del Activo Subyacente")
plt.xlabel("Tiempo (días)")
plt.ylabel("Precio del Activo Subyacente")
plt.grid(True)
plt.show()

```

3.1 RMSE

```

[ ]: Resultados = pd.merge(Resultados_BS, Resultados_MC, on=['Fecha Valoración', 'Fecha Vencimiento', 'Activo'], suffixes=('_BS', '_MC'))

def calcular_rmse(precio_bs, precio_mc):
    """
    Calcula el RMSE entre los precios obtenidos con Black-Scholes y Monte Carlo.

    Parámetros:
    - precio_bs: Precio de la opción usando Black-Scholes.
    - precio_mc: Precio de la opción usando Monte Carlo.

    Retorna:
    - RMSE: Error cuadrático medio entre los precios.
    """
    return math.sqrt(np.mean((precio_bs - precio_mc) ** 2))

resultados_rmse = []

for _, row in Resultados.iterrows():
    resultados_rmse.append({
        'Fecha Valoración': row['Fecha Valoración'],
        'Fecha Vencimiento': row['Fecha Vencimiento'],
        'Activo': row['Activo'],
        'Moneyness 0.9': 0.9,
        'RMSE Call 0.9': calcular_rmse(row['Precio Call 0.9_BS'], row['Precio Call 0.9_MC']),
        'RMSE Put 0.9': calcular_rmse(row['Precio Put 0.9_BS'], row['Precio Put 0.9_MC']),
        'Moneyness 1.1': 1.1,
        'RMSE Call 1.1': calcular_rmse(row['Precio Call 1.1_BS'], row['Precio Call 1.1_MC']),
        'RMSE Put 1.1': calcular_rmse(row['Precio Put 1.1_BS'], row['Precio Put 1.1_MC'])
    })

```

```

    })

Errores_MC = pd.DataFrame(resultados_rmse)

def calcular_rmse_general(columna_bs, columna_mc):
    """
    Calcula el RMSE general para toda la columna.

    Parámetros:
    - columna_bs: Serie de precios de Black-Scholes.
    - columna_mc: Serie de precios de Monte Carlo.

    Retorna:
    - RMSE: Error cuadrático medio de toda la columna.
    """
    return math.sqrt(np.mean((columna_bs - columna_mc) ** 2))

# Calcular RMSE general para cada tipo de opción
rmse_call_0_9 = calcular_rmse_general(Resultados['Precio Call 0.9_BS'],
    ↪Resultados['Precio Call 0.9_MC'])
rmse_put_0_9 = calcular_rmse_general(Resultados['Precio Put 0.9_BS'],
    ↪Resultados['Precio Put 0.9_MC'])
rmse_call_1_1 = calcular_rmse_general(Resultados['Precio Call 1.1_BS'],
    ↪Resultados['Precio Call 1.1_MC'])
rmse_put_1_1 = calcular_rmse_general(Resultados['Precio Put 1.1_BS'],
    ↪Resultados['Precio Put 1.1_MC'])

# Guardar RMSE general en un DataFrame
Errores_MC_General = pd.DataFrame({
    'Tipo de Opción': ['Call 0.9', 'Put 0.9', 'Call 1.1', 'Put 1.1'],
    'RMSE': [rmse_call_0_9, rmse_put_0_9, rmse_call_1_1, rmse_put_1_1]
})

print(Errores_MC_General)

```

4 Método de Diferencias Finitas

```

[ ]: def calcular_griegas_DF(S, K, r, sigma, T, trayectorias, pasosTiempo):

    """
    Esta función calcula las griegas de una opción europea (Delta, Gamma, Vega,
    ↪Theta y Rho) utilizando el método de diferencias finitas,
    que evalúa cómo varía el valor de la opción respecto a pequeñas variaciones
    ↪en sus parámetros de entrada.

    Parámetros:

```

- *S*: precio actual del activo subyacente.
- *K*: precio de ejercicio de la opción.
- *r*: tasa libre de riesgo.
- *sigma*: volatilidad del activo subyacente.
- *T*: tiempo hasta el vencimiento de la opción (en años).
- *trayectorias*: número de trayectorias simuladas.
- *pasosTiempo*: número de pasos de tiempo en la simulación.

Resultado:

- *Delta, Gamma, Vega, Theta, Rho.*

"""

Tamaño del paso para diferencias finitas

epsilon = 0.0001

Valor de las opciones calculadas con Monte Carlo

valor_call_base, valor_put_base = calcular_valor_opcion(S, K, r, sigma, T,
↪trayectorias, pasosTiempo)

Delta: Sensibilidad con respecto al precio del activo subyacente

valor_call_up, valor_put_up = calcular_valor_opcion(S + epsilon, K, r, sigma,
↪T, trayectorias, pasosTiempo)

valor_call_down, valor_put_down = calcular_valor_opcion(S - epsilon, K, r,
↪sigma, T, trayectorias, pasosTiempo)

delta_call = (valor_call_up - valor_call_down) / (2 * epsilon)

delta_put = (valor_put_up - valor_put_down) / (2 * epsilon)

Gamma: Sensibilidad de Delta con respecto al precio del activo subyacente

valor_call_up_s, valor_put_up_s = calcular_valor_opcion(S + epsilon, K, r,
↪sigma, T, trayectorias, pasosTiempo)

valor_call_down_s, valor_put_down_s = calcular_valor_opcion(S - epsilon, K,
↪r, sigma, T, trayectorias, pasosTiempo)

gamma_call = (valor_call_up_s - 2 * valor_call_base + valor_call_down_s) /
↪(epsilon ** 2)

gamma_put = (valor_put_up_s - 2 * valor_put_base + valor_put_down_s) /
↪(epsilon ** 2)

Vega: Sensibilidad con respecto a la volatilidad

valor_call_up_vol, valor_put_up_vol = calcular_valor_opcion(S, K, r, sigma +
↪epsilon, T, trayectorias, pasosTiempo)

valor_call_down_vol, valor_put_down_vol = calcular_valor_opcion(S, K, r,
↪sigma - epsilon, T, trayectorias, pasosTiempo)

vega_call = (valor_call_up_vol - valor_call_down_vol) / (2 * epsilon)

vega_put = (valor_put_up_vol - valor_put_down_vol) / (2 * epsilon)

Theta: Sensibilidad con respecto al tiempo

```

valor_call_up_t, valor_put_up_t = calcular_valor_opcion(S, K, r, sigma, T +
↪epsilon, trayectorias, pasosTiempo)
valor_call_down_t, valor_put_down_t = calcular_valor_opcion(S, K, r, sigma, T
↪- epsilon, trayectorias, pasosTiempo)
theta_call = -(valor_call_up_t - valor_call_down_t) / (2 * epsilon)
theta_put = -(valor_put_up_t - valor_put_down_t) / (2 * epsilon)

# Rho: Sensibilidad con respecto a la tasa de interés
valor_call_up_r, valor_put_up_r = calcular_valor_opcion(S, K, r + epsilon,
↪sigma, T, trayectorias, pasosTiempo)
valor_call_down_r, valor_put_down_r = calcular_valor_opcion(S, K, r -
↪epsilon, sigma, T, trayectorias, pasosTiempo)
rho_call = (valor_call_up_r - valor_call_down_r) / (2 * epsilon)
rho_put = (valor_put_up_r - valor_put_down_r) / (2 * epsilon)

return delta_call, delta_put, vega_call, vega_put, rho_call, rho_put,
↪gamma_call, gamma_put, theta_call, theta_put

```

```

[ ]: griegas_DF = []

for _, row in Datos.iterrows():
    S = row['Precio Subyacente']
    T = row['Años hasta vencimiento']
    K_0_9 = row['Precio Ejercicio 0.9']
    K_1_1 = row['Precio Ejercicio 1.1']

    # Realizar la simulación para las dos moneyness
    griegas_0_9 = calcular_griegas_DF(S, K_0_9, r, row['Volatilidad Anual'], T,
↪trayectorias, pasosTiempo)
    griegas_1_1 = calcular_griegas_DF(S, K_1_1, r, row['Volatilidad Anual'], T,
↪trayectorias, pasosTiempo)

    # Almacenar los resultados
    griegas_DF.append({
        'Fecha Valoración': row['Fecha Valoración'],
        'Fecha Vencimiento': row['Fecha Vencimiento'],
        'Activo': row['Activo'],
        'Moneyness 0.9': 0.9,
        'Delta Call 0.9': griegas_0_9[0],
        'Delta Put 0.9': griegas_0_9[1],
        'Gamma Call 0.9': griegas_0_9[6],
        'Gamma Put 0.9': griegas_0_9[7],
        'Vega Call 0.9': griegas_0_9[2],
        'Vega Put 0.9': griegas_0_9[3],
        'Rho Call 0.9': griegas_0_9[4],
        'Rho Put 0.9': griegas_0_9[5],
    })

```

```

        'Theta Call 0.9': griegas_0_9[8],
        'Theta Put 0.9': griegas_0_9[9],
        'Moneyness 1.1': 1.1,
        'Delta Call 1.1': griegas_1_1[0],
        'Delta Put 1.1': griegas_1_1[1],
        'Gamma Call 1.1': griegas_1_1[6],
        'Gamma Put 1.1': griegas_1_1[7],
        'Vega Call 1.1': griegas_1_1[2],
        'Vega Put 1.1': griegas_1_1[3],
        'Rho Call 1.1': griegas_1_1[4],
        'Rho Put 1.1': griegas_1_1[5],
        'Theta Call 1.1': griegas_1_1[8],
        'Theta Put 1.1': griegas_1_1[9]
    })

griegas_DF = pd.DataFrame(griegas_DF)
#print(griegas_DF.head())

# Exportar resultados a un archivo Excel
#griegas_DF.to_excel("Griegas_DF.xlsx", index=False)
#from google.colab import files
#files.download('griegas_DF.xlsx')

```

4.1 RMSE

4.1.1 RMSE Total

```

[ ]: Errores_DF = Resultados_BS.merge(griegas_DF,
                                     on=['Fecha Valoración', 'Fecha_A_
                                     ↪Vencimiento', 'Activo'],
                                     suffixes=('_BS', '_MC'))

griegas = ['Delta', 'Vega', 'Gamma', 'Theta', 'Rho']

RMSE_DF = []

for griega in griegas:
    for m in [0.9, 1.1]:

        # Call y Put para cada griega
        columna_BS_call = f'{griega} Call {m}_BS'
        columna_MC_call = f'{griega} Call {m}_MC'

        columna_BS_put = f'{griega} Put {m}_BS'
        columna_MC_put = f'{griega} Put {m}_MC'

        # Verificar si las columnas existen en los datos

```

```

        if columna_BS_call in Errores_DF.columns and columna_MC_call in \
↳Errores_DF.columns and \
            columna_BS_put in Errores_DF.columns and columna_MC_put in \
↳Errores_DF.columns:

            # Calcular RMSE general para Call y Put
            rmse_call = np.sqrt(((Errores_DF[columna_BS_call] - \
↳Errores_DF[columna_MC_call]) ** 2).mean())
            rmse_put = np.sqrt(((Errores_DF[columna_BS_put] - \
↳Errores_DF[columna_MC_put]) ** 2).mean())

            # RMSE general para la griega y moneyness
            rmse_general = np.sqrt(((rmse_call ** 2) + (rmse_put ** 2)) / 2)

            RMSE_DF.append({
                'Griega': griega,
                'Moneyness': m,
                'RMSE Call': round(rmse_call, 4),
                'RMSE Put': round(rmse_put, 4),
                'RMSE General': round(rmse_general, 4)
            })

```

```
RMSE_DF = pd.DataFrame(RMSE_DF)
```

4.1.2 RMSE por Fecha de Vencimiento

```

[ ]: griegas = ['Delta', 'Vega', 'Gamma', 'Theta', 'Rho']
    moneyness = [0.9, 1.1]

    RMSE_DF_Vencimiento = []

    for T, grupo in Errores_DF.groupby('Fecha Vencimiento'):
        for griega in griegas:
            for m in moneyness:
                # Definir nombres de columnas para Call y Put
                columna_BS_call = f'{griega} Call {m}_BS'
                columna_MC_call = f'{griega} Call {m}_MC'
                columna_BS_put = f'{griega} Put {m}_BS'
                columna_MC_put = f'{griega} Put {m}_MC'

                # Verificar si las columnas existen en los datos
                if columna_BS_call in grupo.columns and columna_MC_call in grupo.
↳columns and \
                    columna_BS_put in grupo.columns and columna_MC_put in grupo.
↳columns:

```

```

        # Calcular RMSE general para Call y Put
        rmse_call = np.sqrt(((grupo[columna_BS_call] -
        grupo[columna_MC_call]) ** 2).mean())
        rmse_put = np.sqrt(((grupo[columna_BS_put] -
        grupo[columna_MC_put]) ** 2).mean())

        # RMSE general como el promedio de Call y Put
        rmse_general = np.sqrt(((rmse_call ** 2) + (rmse_put ** 2)) / 2)

        RMSE_DF_Vencimiento.append({
            'Fecha Vencimiento': T,
            'Griega': griega,
            'Moneyness': m,
            'RMSE Call': rmse_call,
            'RMSE Put': rmse_put,
            'RMSE General': rmse_general
        })

RMSE_DF_Vencimiento = pd.DataFrame(RMSE_DF_Vencimiento)

```

4.1.3 RMSE por activo subyacente

```

[ ]: griegas = ['Delta', 'Vega', 'Gamma', 'Theta', 'Rho']
     moneyness = [0.9, 1.1]

RMSE_DF_Activo = []

for griega in griegas:
    for m in moneyness:
        for activo in Errores_DF['Activo'].unique():

            # Filtrar los datos por activo
            df_activo = Errores_DF[Errores_DF['Activo'] == activo]

            # Call y Put para cada griega
            columna_BS_call = f'{griega} Call {m}_BS'
            columna_MC_call = f'{griega} Call {m}_MC'
            columna_BS_put = f'{griega} Put {m}_BS'
            columna_MC_put = f'{griega} Put {m}_MC'

            # Verificar si las columnas existen en los datos
            if columna_BS_call in df_activo.columns and columna_MC_call in
            df_activo.columns and \
                columna_BS_put in df_activo.columns and columna_MC_put in
            df_activo.columns:

```

```

        # Calcular RMSE general para Call y Put para cada activo
        rmse_call = np.sqrt(((df_activo[columna_BS_call] -
↪df_activo[columna_MC_call]) ** 2).mean())
        rmse_put = np.sqrt(((df_activo[columna_BS_put] -
↪df_activo[columna_MC_put]) ** 2).mean())

        # RMSE general para la griega y moneyness para cada activo
        rmse_general = np.sqrt((rmse_call ** 2) + (rmse_put ** 2)) / 2)

        RMSE_DF_Activo.append({
            'Activo': activo,
            'Griega': griega,
            'Moneyness': m,
            'RMSE Call': rmse_call,
            'RMSE Put': rmse_put,
            'RMSE General': rmse_general
        })

RMSE_DF_Activo = pd.DataFrame(RMSE_DF_Activo)

```

5 Método de la Trayectoria

```

[ ]: def calcular_griegas_Trayectoria(S, K, r, sigma, T, trayectorias, pasosTiempo,
↪tipo_opcion='call'):

    """
    Esta función calcula las griegas (Delta, Vega, Theta y Rho) de una opción
↪europea utilizando el método de la trayectoria.

    Parámetros:
        - S: precio actual del activo subyacente.
        - K: precio de ejercicio de la opción.
        - r: tasa libre de riesgo.
        - sigma: volatilidad del activo subyacente.
        - T: tiempo hasta el vencimiento de la opción (en años).
        - trayectorias: número de trayectorias simuladas.
        - pasosTiempo: número de pasos de tiempo en la simulación.
        - tipo_opcion: 'call' o 'put'.

    Resultado:
        - valor de la opción, Delta, Vega, Theta y Rho.
    """

    # Generar trayectorias de precios del activo subyacente
    ST = GBM_fd(S, r, sigma, T, trayectorias, pasosTiempo)[: , -1]

```



```

# Payoff descontado y función indicatriz
if tipo_opcion == 'call':
    Y = np.exp(-r * T) * np.maximum(ST - K, 0) # Payoff Call
    indicatriz = (ST > K).astype(float) # Indicatriz para Call
elif tipo_opcion == 'put':
    Y = np.exp(-r * T) * np.maximum(K - ST, 0) # Payoff Put
    indicatriz = -(ST < K).astype(float) # Indicatriz para Put
else:
    raise ValueError("El parámetro tipo_opcion debe ser 'call' o 'put'.")

# Z calculado desde las trayectorias
Z = ((np.log(ST / S) - (r - 0.5 * sigma**2) * T)) / (sigma * np.sqrt(T))

# Delta: Sensibilidad con respecto al precio del activo subyacente
delta_estimador = np.exp(-r * T) * (ST / S) * indicatriz

# Vega: Sensibilidad con respecto a la volatilidad
vega_estimador = np.exp(-r * T) * (-sigma*T + np.sqrt(T)*Z) * ST * indicatriz

# Theta: Sensibilidad con respecto al tiempo
theta_estimador = r*Y - np.exp(-r * T) * indicatriz * ((r - 0.5 * sigma**2)
↳+ (sigma * Z) / (2 * np.sqrt(T))) * ST

# Rho: Sensibilidad con respecto a la tasa de interés
rho_estimador = -T*Y + indicatriz * np.exp(-r * T) * T * ST

# Valores esperados (Monte Carlo)
valor_opcion = np.mean(Y)
delta = np.mean(delta_estimador)
vega = np.mean(vega_estimador)
theta = np.mean(theta_estimador)
rho = np.mean(rho_estimador)

# Retornar solo los valores sin las varianzas
return {
    "valor_opcion": valor_opcion,
    "delta": delta,
    "vega": vega,
    "theta": theta,
    "rho": rho,
}

```

```

[ ]: griegas_Trayectoria = []

for _, row in Datos.iterrows():
    S = row['Precio Subyacente']

```

```

T = row['Años hasta vencimiento']
K_0_9 = row['Precio Ejercicio 0.9']
K_1_1 = row['Precio Ejercicio 1.1']

# Realizar la simulación para las dos moneyness (0.9 y 1.1)
resultados_0_9_call = calcular_griegas_Trayectoria(S, K_0_9, r,
↪row['Volatilidad Anual'], T, trayectorias, pasosTiempo, tipo_opcion='call')
resultados_1_1_call = calcular_griegas_Trayectoria(S, K_1_1, r,
↪row['Volatilidad Anual'], T, trayectorias, pasosTiempo, tipo_opcion='call')

resultados_0_9_put = calcular_griegas_Trayectoria(S, K_0_9, r,
↪row['Volatilidad Anual'], T, trayectorias, pasosTiempo, tipo_opcion='put')
resultados_1_1_put = calcular_griegas_Trayectoria(S, K_1_1, r,
↪row['Volatilidad Anual'], T, trayectorias, pasosTiempo, tipo_opcion='put')

# Almacenar los resultados
griegas_Trayectoria.append({
    'Fecha Valoración': row['Fecha Valoración'],
    'Fecha Vencimiento': row['Fecha Vencimiento'],
    'Activo': row['Activo'],
    'Moneyness 0.9': 0.9,
    'Delta Call 0.9': resultados_0_9_call['delta'],
    'Delta Put 0.9': resultados_0_9_put['delta'],
    'Gamma Call 0.9': '', # Eliminar la varianza
    'Gamma Put 0.9': '',
    'Vega Call 0.9': resultados_0_9_call['vega'],
    'Vega Put 0.9': resultados_0_9_put['vega'],
    'Rho Call 0.9': resultados_0_9_call['rho'],
    'Rho Put 0.9': resultados_0_9_put['rho'],
    'Theta Call 0.9': resultados_0_9_call['theta'],
    'Theta Put 0.9': resultados_0_9_put['theta'],
    'Moneyness 1.1': 1.1,
    'Delta Call 1.1': resultados_1_1_call['delta'],
    'Delta Put 1.1': resultados_1_1_put['delta'],
    'Gamma Call 1.1': '',
    'Gamma Put 1.1': '',
    'Vega Call 1.1': resultados_1_1_call['vega'],
    'Vega Put 1.1': resultados_1_1_put['vega'],
    'Rho Call 1.1': resultados_1_1_call['rho'],
    'Rho Put 1.1': resultados_1_1_put['rho'],
    'Theta Call 1.1': resultados_1_1_call['theta'],
    'Theta Put 1.1': resultados_1_1_put['theta'],
})

griegas_Trayectoria = pd.DataFrame(griegas_Trayectoria)
#print(griegas_Trayectoria.head())

```

```
# Exportar resultados a un archivo Excel
#griegas_Trayectoria.to_excel("Griegas_Trayectoria.xlsx", index=False)
#from google.colab import files
#files.download('griegas_Trayectoria.xlsx')
```

5.1 RMSE

5.1.1 RMSE Total

```
[ ]: Errores_Trayectoria = Resultados_BS.merge(griegas_Trayectoria,
                                                on=['Fecha Valoración', 'Fecha_
↳Vencimiento', 'Activo'],
                                                suffixes=('_BS', '_MC'))

griegas = ['Delta', 'Vega', 'Theta', 'Rho']

RMSE_Trayectoria = []

for griega in griegas:
    for m in [0.9, 1.1]:

        # Call y Put para cada griega
        columna_BS_call = f'{griega} Call {m}_BS'
        columna_MC_call = f'{griega} Call {m}_MC'

        columna_BS_put = f'{griega} Put {m}_BS'
        columna_MC_put = f'{griega} Put {m}_MC'

        # Verificar si las columnas existen en los datos
        if columna_BS_call in Errores_Trayectoria.columns and columna_MC_call_
↳in Errores_Trayectoria.columns and \
            columna_BS_put in Errores_Trayectoria.columns and columna_MC_put in_
↳Errores_Trayectoria.columns:

            # Calcular RMSE general para Call y Put
            rmse_call = np.sqrt(((Errores_Trayectoria[columna_BS_call] -_
↳Errores_Trayectoria[columna_MC_call]) ** 2).mean())
            rmse_put = np.sqrt(((Errores_Trayectoria[columna_BS_put] -_
↳Errores_Trayectoria[columna_MC_put]) ** 2).mean())

            # RMSE general para la griega y moneyness
            rmse_general = np.sqrt(((rmse_call ** 2) + (rmse_put ** 2)) / 2)

            RMSE_Trayectoria.append({
                'Griega': griega,
                'Moneyness': m,
                'RMSE Call': round(rmse_call, 3),
```

```

        'RMSE Put': round(rmse_put, 3),
        'RMSE General': round(rmse_general, 3)
    })

```

```

RMSE_Trayectoria = pd.DataFrame(RMSE_Trayectoria)

```

5.1.2 RMSE por Fecha de Vencimiento

```

[ ]: griegas = ['Delta', 'Vega', 'Theta', 'Rho']
     moneyness = [0.9, 1.1]

RMSE_Trayectoria_Vencimiento = []

for T, grupo in Errores_Trayectoria.groupby('Fecha Vencimiento'):
    for griega in griegas:
        for m in moneyness:
            # Definir nombres de columnas para Call y Put
            columna_BS_call = f'{griega} Call {m}_BS'
            columna_MC_call = f'{griega} Call {m}_MC'
            columna_BS_put = f'{griega} Put {m}_BS'
            columna_MC_put = f'{griega} Put {m}_MC'

            # Verificar si las columnas existen en los datos
            if columna_BS_call in grupo.columns and columna_MC_call in grupo.
↳columns and \
                columna_BS_put in grupo.columns and columna_MC_put in grupo.
↳columns:

                # Calcular RMSE general para Call y Put
                rmse_call = np.sqrt(((grupo[columna_BS_call] -
↳grupo[columna_MC_call]) ** 2).mean())
                rmse_put = np.sqrt(((grupo[columna_BS_put] -
↳grupo[columna_MC_put]) ** 2).mean())

                # RMSE general como el promedio de Call y Put
                rmse_general = np.sqrt(((rmse_call ** 2) + (rmse_put ** 2)) / 2)

                RMSE_Trayectoria_Vencimiento.append({
                    'Fecha Vencimiento': T,
                    'Griega': griega,
                    'Moneyness': m,
                    'RMSE Call': rmse_call,
                    'RMSE Put': rmse_put,
                    'RMSE General': rmse_general
                })

```

```
RMSE_Trayectoria_Vencimiento = pd.DataFrame(RMSE_Trayectoria_Vencimiento)
```

5.1.3 RMSE por activo

```
[ ]: griegas = ['Delta', 'Vega', 'Theta', 'Rho']
moneyness = [0.9, 1.1]

RMSE_Trayectoria_Activo = []

for griega in griegas:
    for m in moneyness:
        for activo in Errores_Trayectoria['Activo'].unique(): # Iterar sobre
            cada activo

            # Filtrar los datos por activo
            df_activo = Errores_Trayectoria[Errores_Trayectoria['Activo'] ==
            activo]

            # Call y Put para cada griega
            columna_BS_call = f'{griega} Call {m}_BS'
            columna_MC_call = f'{griega} Call {m}_MC'
            columna_BS_put = f'{griega} Put {m}_BS'
            columna_MC_put = f'{griega} Put {m}_MC'

            # Verificar si las columnas existen en los datos
            if columna_BS_call in df_activo.columns and columna_MC_call in
            df_activo.columns and \
                columna_BS_put in df_activo.columns and columna_MC_put in
            df_activo.columns:

                # Calcular RMSE general para Call y Put para cada activo
                rmse_call = np.sqrt(((df_activo[columna_BS_call] -
            df_activo[columna_MC_call]) ** 2).mean())
                rmse_put = np.sqrt(((df_activo[columna_BS_put] -
            df_activo[columna_MC_put]) ** 2).mean())

                # RMSE general para la griega y moneyness para cada activo
                rmse_general = np.sqrt(((rmse_call ** 2) + (rmse_put ** 2)) / 2)

                RMSE_Trayectoria_Activo.append({
                    'Activo': activo,
                    'Griega': griega,
                    'Moneyness': m,
                    'RMSE Call': rmse_call,
                    'RMSE Put': rmse_put,
```

```

        'RMSE General': rmse_general
    })

RMSE_Trayectoria_Activo = pd.DataFrame(RMSE_Trayectoria_Activo)

```

6 Método de Razón de Verosimilitudes

```

[ ]: def calcular_griegas_RV(S, K, r, sigma, T, trayectorias, pasosTiempo):

    """
    Esta función calcula las griegas de una opción europea (Delta, Gamma, Vega,
    ↪Theta y Rho) utilizando el método de razón de verosimilitudes.

    Parámetros:
        - S: precio actual del activo subyacente.
        - K: precio de ejercicio de la opción.
        - r: tasa libre de riesgo.
        - sigma: volatilidad del activo subyacente.
        - T: tiempo hasta el vencimiento de la opción (en años).
        - trayectorias: número de simulaciones de monte carlo.
        - pasostiempo: número de pasos de tiempo en cada simulación.

    Resultado:
        - Delta, Gamma, Vega, Theta, Rho.
    """

    # Simulación de trayectorias
    ST = GBM_fd(S, r, sigma, T, trayectorias, pasosTiempo)[: , -1]

    # Payoff para CALL y PUT
    payoff_call = np.maximum(ST - K, 0)
    payoff_put = np.maximum(K - ST, 0)

    Z = ((np.log(ST / S) - (r - 0.5 * sigma**2) * T)) / (sigma * np.sqrt(T))

    # Delta: Sensibilidad con respecto al precio del activo subyacente
    delta_call = np.exp(-r * T) * payoff_call * Z / (S * sigma * np.sqrt(T))
    delta_put = np.exp(-r * T) * payoff_put * Z / (S * sigma * np.sqrt(T))

    # Gamma: Sensibilidad de Delta con respecto al precio del activo subyacente
    gamma_call = np.exp(-r * T) * payoff_call * (Z**2 - 1 - sigma * np.sqrt(T) *
    ↪Z) / (S**2 * sigma**2 * T)
    gamma_put = np.exp(-r * T) * payoff_put * (Z**2 - 1 - sigma * np.sqrt(T) *
    ↪Z) / (S**2 * sigma**2 * T)

    # Vega: Sensibilidad con respecto a la volatilidad

```

```

vega_call = np.exp(-r * T) * payoff_call * ((Z**2 - 1) / sigma - Z * np.
↪sqrt(T))
vega_put = np.exp(-r * T) * payoff_put * ((Z**2 - 1) / sigma - Z * np.sqrt(T))

# Theta: Sensibilidad con respecto al tiempo
theta_call = -np.exp(-r * T) * payoff_call * (
    0.5 * (Z**2 - 1) / T + (r - 0.5 * sigma**2) / (sigma * np.sqrt(T)) * Z)
theta_put = -np.exp(-r * T) * payoff_put * (
    0.5 * (Z**2 - 1) / T + (r - 0.5 * sigma**2) / (sigma * np.sqrt(T)) * Z)

# Rho: Sensibilidad con respecto a la tasa de interés
rho_call = np.exp(-r * T) * payoff_call * Z * np.sqrt(T) / sigma
rho_put = np.exp(-r * T) * payoff_put * Z * np.sqrt(T) / sigma

# Valores promedio de las griegas para CALL y PUT
delta_promedio_call = np.mean(delta_call)
gamma_promedio_call = np.mean(gamma_call)
vega_promedio_call = np.mean(vega_call)
theta_promedio_call = np.mean(theta_call)
rho_promedio_call = np.mean(rho_call)

delta_promedio_put = np.mean(delta_put)
gamma_promedio_put = np.mean(gamma_put)
vega_promedio_put = np.mean(vega_put)
theta_promedio_put = np.mean(theta_put)
rho_promedio_put = np.mean(rho_put)

# Almacenar los resultados en un diccionario
griegas_resultados = {
    'Delta Call': delta_promedio_call,
    'Gamma Call': gamma_promedio_call,
    'Vega Call': vega_promedio_call,
    'Theta Call': theta_promedio_call,
    'Rho Call': rho_promedio_call,
    'Delta Put': delta_promedio_put,
    'Gamma Put': gamma_promedio_put,
    'Vega Put': vega_promedio_put,
    'Theta Put': theta_promedio_put,
    'Rho Put': rho_promedio_put
}

return griegas_resultados, ST

```

```

[ ]: griegas_RV = []

for _, row in Datos.iterrows():
    S = row['Precio Subyacente']

```

```

T = row['Años hasta vencimiento']
K_0_9 = row['Precio Ejercicio 0.9']
K_1_1 = row['Precio Ejercicio 1.1']

# Realizar la simulación para las dos moneyness (0.9 y 1.1)
resultados_0_9 = calcular_griegas_RV(S, K_0_9, r, row['Volatilidad Anual'],
↪T, trayectorias, pasosTiempo)
resultados_1_1 = calcular_griegas_RV(S, K_1_1, r, row['Volatilidad Anual'],
↪T, trayectorias, pasosTiempo)

# Almacenar los resultados
griegas_RV.append({
    'Fecha Valoración': row['Fecha Valoración'],
    'Fecha Vencimiento': row['Fecha Vencimiento'],
    'Activo': row['Activo'],
    'Moneyness 0.9': 0.9,
    'Delta Call 0.9': resultados_0_9[0]['Delta Call'],
    'Delta Put 0.9': resultados_0_9[0]['Delta Put'],
    'Gamma Call 0.9': resultados_0_9[0]['Gamma Call'],
    'Gamma Put 0.9': resultados_0_9[0]['Gamma Put'],
    'Vega Call 0.9': resultados_0_9[0]['Vega Call'],
    'Vega Put 0.9': resultados_0_9[0]['Vega Put'],
    'Rho Call 0.9': resultados_0_9[0]['Rho Call'],
    'Rho Put 0.9': resultados_0_9[0]['Rho Put'],
    'Theta Call 0.9': resultados_0_9[0]['Theta Call'],
    'Theta Put 0.9': resultados_0_9[0]['Theta Put'],
    'Moneyness 1.1': 1.1,
    'Delta Call 1.1': resultados_1_1[0]['Delta Call'],
    'Delta Put 1.1': resultados_1_1[0]['Delta Put'],
    'Gamma Call 1.1': resultados_1_1[0]['Gamma Call'],
    'Gamma Put 1.1': resultados_1_1[0]['Gamma Put'],
    'Vega Call 1.1': resultados_1_1[0]['Vega Call'],
    'Vega Put 1.1': resultados_1_1[0]['Vega Put'],
    'Rho Call 1.1': resultados_1_1[0]['Rho Call'],
    'Rho Put 1.1': resultados_1_1[0]['Rho Put'],
    'Theta Call 1.1': resultados_1_1[0]['Theta Call'],
    'Theta Put 1.1': resultados_1_1[0]['Theta Put'],
})

griegas_RV = pd.DataFrame(griegas_RV)
#print(griegas_RV.head())

# Exportar resultados a un archivo Excel
#griegas_RV.to_excel("Griegas_RV.xlsx", index=False)
#from google.colab import files
#files.download('griegas_RV.xlsx')

```


6.1 Errores

6.1.1 RMSE general

```
[ ]: Errores_RV = Resultados_BS.merge(griegas_RV,
                                     on=['Fecha Valoración', 'Fecha Vencimiento',
                                     ↪ 'Activo'],
                                     suffixes=('_BS', '_MC'))

griegas = ['Delta', 'Gamma', 'Vega', 'Rho', 'Theta']

RMSE_RV = []

for griega in griegas:
    for m in [0.9, 1.1]:

        # Call y Put para cada griega
        columna_BS_call = f'{griega} Call {m}_BS'
        columna_MC_call = f'{griega} Call {m}_MC'

        columna_BS_put = f'{griega} Put {m}_BS'
        columna_MC_put = f'{griega} Put {m}_MC'

        # Verificar si las columnas existen en los datos
        if columna_BS_call in Errores_RV.columns and columna_MC_call in ↪
        ↪ Errores_RV.columns and \
            columna_BS_put in Errores_RV.columns and columna_MC_put in ↪
            ↪ Errores_RV.columns:

            # Calcular RMSE general para Call y Put
            rmse_call = np.sqrt(((Errores_RV[columna_BS_call] - ↪
            ↪ Errores_RV[columna_MC_call]) ** 2).mean())
            rmse_put = np.sqrt(((Errores_RV[columna_BS_put] - ↪
            ↪ Errores_RV[columna_MC_put]) ** 2).mean())

            # RMSE general para la griega y moneyness
            rmse_general = np.sqrt(((rmse_call ** 2) + (rmse_put ** 2)) / 2)

            RMSE_RV.append({
                'Griega': griega,
                'Moneyness': m,
                'RMSE Call': round(rmse_call, 3),
                'RMSE Put': round(rmse_put, 3),
                'RMSE General': round(rmse_general, 3)
            })

RMSE_RV = pd.DataFrame(RMSE_RV)
```

6.1.2 RMSE por vencimiento

```
[ ]: griegas = ['Delta', 'Vega', 'Gamma', 'Theta', 'Rho']
moneyness = [0.9, 1.1]

RMSE_RV_Vencimiento = []

# Iterar sobre cada fecha de vencimiento en el dataset
for T, grupo in Errores_RV.groupby('Fecha Vencimiento'):
    for griega in griegas:
        for m in moneyness:
            # Definir nombres de columnas para Call y Put
            columna_BS_call = f'{griega} Call {m}_BS'
            columna_MC_call = f'{griega} Call {m}_MC'
            columna_BS_put = f'{griega} Put {m}_BS'
            columna_MC_put = f'{griega} Put {m}_MC'

            # Verificar si las columnas existen en los datos
            if columna_BS_call in grupo.columns and columna_MC_call in grupo.
↳columns and \
                columna_BS_put in grupo.columns and columna_MC_put in grupo.
↳columns:

                # Calcular RMSE general para Call y Put
                rmse_call = np.sqrt(((grupo[columna_BS_call] -
↳grupo[columna_MC_call]) ** 2).mean())
                rmse_put = np.sqrt(((grupo[columna_BS_put] -
↳grupo[columna_MC_put]) ** 2).mean())

                # RMSE general como el promedio de Call y Put
                rmse_general = np.sqrt(((rmse_call ** 2) + (rmse_put ** 2)) / 2)

                RMSE_RV_Vencimiento.append({
                    'Fecha Vencimiento': T,
                    'Griega': griega,
                    'Moneyness': m,
                    'RMSE Call': rmse_call,
                    'RMSE Put': rmse_put,
                    'RMSE General': rmse_general
                })

RMSE_RV_Vencimiento = pd.DataFrame(RMSE_RV_Vencimiento)
```

6.1.3 RMSE por Activo

```
[ ]: griegas = ['Delta', 'Vega', 'Gamma', 'Theta', 'Rho']
moneyness = [0.9, 1.1]

RMSE_RV_Activo = []

for griega in griegas:
    for m in moneyness:
        for activo in Errores_RV['Activo'].unique(): # Iterar sobre cada activo

            # Filtrar los datos por activo
            df_activo = Errores_RV[Errores_RV['Activo'] == activo]

            # Call y Put para cada griega
            columna_BS_call = f'{griega} Call {m}_BS'
            columna_MC_call = f'{griega} Call {m}_MC'
            columna_BS_put = f'{griega} Put {m}_BS'
            columna_MC_put = f'{griega} Put {m}_MC'

            # Verificar si las columnas existen en los datos
            if columna_BS_call in df_activo.columns and columna_MC_call in df_activo.columns and \
                columna_BS_put in df_activo.columns and columna_MC_put in df_activo.columns:

                # Calcular RMSE general para Call y Put para cada activo
                rmse_call = np.sqrt(((df_activo[columna_BS_call] - df_activo[columna_MC_call]) ** 2).mean())
                rmse_put = np.sqrt(((df_activo[columna_BS_put] - df_activo[columna_MC_put]) ** 2).mean())

                # RMSE general para la griega y moneyness para cada activo
                rmse_general = np.sqrt(((rmse_call ** 2) + (rmse_put ** 2)) / 2)

                RMSE_RV_Activo.append({
                    'Activo': activo,
                    'Griega': griega,
                    'Moneyness': m,
                    'RMSE Call': rmse_call,
                    'RMSE Put': rmse_put,
                    'RMSE General': rmse_general
                })

RMSE_RV_Activo = pd.DataFrame(RMSE_RV_Activo)
```

7 Gráficos

7.1 Gráfico 1: Griegas para cada activo y cada método de estimación

```
[ ]: activos_unicos = griegas_RV['Activo'].unique()

metricas = ['Delta', 'Vega', 'Theta', 'Rho']

estilos_linea = {
    "Black-Scholes": {"linestyle": "--", "linewidth": 2, "alpha": 0.8},
    "Diferencias Finitas": {"linestyle": "--", "linewidth": 2, "alpha": 0.8},
    "Trayectoria": {"linestyle": ":", "linewidth": 2, "alpha": 0.8}, # Línea
    ↪punteada
    "Razón de Verosimilitudes": {"linestyle": "-.", "linewidth": 2, "alpha": 0.
    ↪8}, # Línea guion-punto
}

colores = {
    "Black-Scholes": "green",
    "Diferencias Finitas": "orange",
    "Trayectoria": "blue",
    "Razón de Verosimilitudes": "red",
}

for activo in activos_unicos:
    data_sources = {
        "Black-Scholes": Resultados_BS[Resultados_BS['Activo'] == activo],
        "Diferencias Finitas": griegas_DF[griegas_DF['Activo'] == activo],
        "Trayectoria": griegas_Trayectoria[griegas_Trayectoria['Activo'] ==
    ↪activo],
        "Razón de Verosimilitudes": griegas_RV[griegas_RV['Activo'] == activo],
    }

    for metrica in metricas:
        # Crear figura con 4 subgráficos
        fig, axes = plt.subplots(2, 2, figsize=(14, 10))
        fig.suptitle(f'{activo} - {metrica}', fontsize=16)

        # Crear nombres de columnas
        call_09, call_11 = f"{metrica} Call 0.9", f"{metrica} Call 1.1"
        put_09, put_11 = f"{metrica} Put 0.9", f"{metrica} Put 1.1"

        # Función auxiliar para graficar cada subplot
        def graficar(ax, titulo, columna):
            for nombre, df in data_sources.items():
                if columna in df.columns:
                    ax.plot(
```

```

        df['Fecha Vencimiento'], df[columna],
        label=nombre, marker='o', color=colores[nombre],
        linestyle=estilos_linea[nombre]["linestyle"],
        linewidth=estilos_linea[nombre]["linewidth"],
        alpha=estilos_linea[nombre]["alpha"]
    )
    ax.set_title(titulo)
    ax.set_xlabel('Fecha de Vencimiento')
    ax.set_ylabel(metrica)
    ax.tick_params(axis='x', rotation=45)
    ax.legend()

# Graficar en cada subplot
graficar(axes[0, 0], 'Call 0.9', call_09)
graficar(axes[0, 1], 'Call 1.1', call_11)
graficar(axes[1, 0], 'Put 0.9', put_09)
graficar(axes[1, 1], 'Put 1.1', put_11)

plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()

```

7.2 Gráfico 2: RMSE vs Fecha de Vencimiento para cada griega y método de estimación

```
griegas = ['Delta', 'Vega', 'Theta', 'Rho']
```

```
estilos_linea = { "Diferencias Finitas": {"linestyle": "-", "linewidth": 3, "alpha": 0.8}, # Línea
discontinua "Trayectoria": {"linestyle": ":", "linewidth": 3, "alpha": 0.8}, # Línea punteada
"Razón de Verosimilitudes": {"linestyle": "-.", "linewidth": 3, "alpha": 0.8}, # Línea guion-punto
}
```

```
colores = { "Diferencias Finitas": "orange", "Trayectoria": "blue", "Razón de Verosimilitudes":
"red", }
```

```
for griega in griegas:
```

```
plt.figure(figsize=(14, 8))
```

```
fechas_vencimiento = RMSE_Trayectoria_Vencimiento['Fecha Vencimiento'].unique()
```

```
# Inicializar listas para almacenar los RMSE promedio de cada método
```

```
rmse_df_prom = []
```

```
rmse_tray_prom = []
```

```
rmse_rv_prom = []
```

```
for T in fechas_vencimiento:
```

```
    datos_df = RMSE_DF_Vencimiento[(RMSE_DF_Vencimiento['Griega'] == griega) &
(RMSE_DF_Vencimiento['Fecha Vencimiento'] == T)][ 'RMSE Gene
```

```

datos_tray = RMSE_Trayectoria_Vencimiento[(RMSE_Trayectoria_Vencimiento['Griega'] == griega) &
                                            (RMSE_Trayectoria_Vencimiento['Fecha Vencimiento'] == T)][['RMSE Genera

datos_rv = RMSE_RV_Vencimiento[(RMSE_RV_Vencimiento['Griega'] == griega) &
                                (RMSE_RV_Vencimiento['Fecha Vencimiento'] == T)][['RMSE Genera

# Calcular el RMSE promedio para cada método
rmse_df_prom.append(np.mean(datos_df) if len(datos_df) > 0 else np.nan)
rmse_tray_prom.append(np.mean(datos_tray) if len(datos_tray) > 0 else np.nan)
rmse_rv_prom.append(np.mean(datos_rv) if len(datos_rv) > 0 else np.nan)

# Graficar los RMSE promedio para cada método con diferentes estilos de línea
plt.plot(fechas_vencimiento, rmse_df_prom, label="Diferencias Finitas", color=colores["Diferencias Finitas"],
         linestyle=estilos_linea["Diferencias Finitas"]["linestyle"], linewidth=3, markersize=8)

plt.plot(fechas_vencimiento, rmse_tray_prom, label="Trayectoria", color=colores["Trayectoria"],
         linestyle=estilos_linea["Trayectoria"]["linestyle"], linewidth=3, markersize=8)

plt.plot(fechas_vencimiento, rmse_rv_prom, label="Razón de Verosimilitudes", color=colores["Razón de Verosimilitudes"],
         linestyle=estilos_linea["Razón de Verosimilitudes"]["linestyle"], linewidth=3, markersize=8)

# Configurar etiquetas y título
plt.xticks(rotation=45)
plt.title(f'RMSE vs Fecha de Vencimiento - {griega}')
plt.xlabel('Fecha de Vencimiento')
plt.ylabel('RMSE Promedio')
plt.grid(True, linestyle='--', alpha=0.5)

# Leyenda
plt.legend(loc='upper left', fontsize=14)

plt.tight_layout()
plt.show()

```

8 Exportación de resultados

```

[ ]: import os

archivo_excel = "Resultados.xlsx"

if not os.path.exists(archivo_excel):

    with pd.ExcelWriter(archivo_excel, engine='openpyxl') as writer:
        Resultados_BS.to_excel(writer, sheet_name="Black-Scholes", index=False)
        Resultados_MC.to_excel(writer, sheet_name="Monte Carlo", index=False)

```

```

        griegas_DF.to_excel(writer, sheet_name="Diferencias Finitas",
        ↪index=False)
        griegas_Trayectoria.to_excel(writer, sheet_name="Trayectoria",
        ↪index=False)
        griegas_RV.to_excel(writer, sheet_name="Razón de verosimilitudes",
        ↪index=False)

else:

    with pd.ExcelWriter(archivo_excel, engine='openpyxl', mode='a',
    ↪if_sheet_exists='replace') as writer:
        Resultados_BS.to_excel(writer, sheet_name="Black-Scholes", index=False)
        Resultados_MC.to_excel(writer, sheet_name="Monte Carlo", index=False)
        griegas_DF.to_excel(writer, sheet_name="Diferencias Finitas",
        ↪index=False)
        griegas_Trayectoria.to_excel(writer, sheet_name="Trayectoria",
        ↪index=False)
        griegas_RV.to_excel(writer, sheet_name="Razón de verosimilitudes",
        ↪index=False)

archivo_excel_Errores = "Errores.xlsx"

if os.path.exists(archivo_excel_Errores):
    try:
        pd.read_excel(archivo_excel_Errores)
    except Exception:
        os.remove(archivo_excel_Errores)
        print("Archivo corrupto eliminado.")

with pd.ExcelWriter(archivo_excel_Errores, engine='openpyxl') as writer:
    RMSE_DF.to_excel(writer, sheet_name="RMSE Diferencias Finitas", index=False)
    RMSE_Trayectoria.to_excel(writer, sheet_name="RMSE Trayectoria",
    ↪index=False)
    RMSE_RV.to_excel(writer, sheet_name="RMSE Razón de verosimilitudes",
    ↪index=False)
    RMSE_DF_Vencimiento.to_excel(writer, sheet_name="RMSE Fecha Vencimiento
    ↪DF", index=False)
    RMSE_Trayectoria_Vencimiento.to_excel(writer, sheet_name="RMSE Fecha
    ↪Vencimiento Trayectoria", index=False)
    RMSE_RV_Vencimiento.to_excel(writer, sheet_name="RMSE Fecha Vencimiento
    ↪RV", index=False)
    RMSE_DF_Activo.to_excel(writer, sheet_name="RMSE Activo DF", index=False)
    RMSE_Trayectoria_Activo.to_excel(writer, sheet_name="RMSE Activo
    ↪Trayectoria", index=False)
    RMSE_RV_Activo.to_excel(writer, sheet_name="RMSE Activo RV", index=False)

```

```
Errores_MC.to_excel(writer, sheet_name="RMSE MC", index=False)
Errores_MC_General.to_excel(writer, sheet_name="RMSE Total MC", index=False)
```


Declaración Responsable sobre Autoría y Uso Ético de
Herramientas de Inteligencia Artificial (IA)

Yo, Lastra Incera Fátima

Con DNI: 51540429M

declaro de manera responsable que el/la presente:

Trabajo de Fin de Grado (TFG)

Titulado

Estimación de las “griegas” de opciones europeas a partir de simulaciones Monte Carlo

es el resultado de mi trabajo intelectual personal y creativo, y ha sido elaborado de acuerdo con los principios éticos y las normas de integridad vigentes en la comunidad académica y, más específicamente, en la Universidad Complutense de Madrid.

Soy, pues, autor del material aquí incluido y, cuando no ha sido así y he tomado el material de otra fuente, lo he citado o bien he declarado su procedencia de forma clara -incluidas, en su caso, herramientas de inteligencia artificial-. Las ideas y aportaciones principales incluidas en este trabajo, y que acreditan la adquisición de competencias, son mías y no proceden de otras fuentes o han sido reescritas usando material de otras fuentes.

Asimismo, aseguro que los datos y recursos utilizados son legítimos, verificables y han sido obtenidos de fuentes confiables y autorizadas. Además, he tomado medidas para garantizar la confidencialidad y privacidad de los datos utilizados, evitando cualquier tipo de sesgo o discriminación injusta en el tratamiento de la información.

En Madrid a 23/05/2025

