

Comandos propios de I.D.E.A.L.¹

1. **Call(s,K)**; Calcula el pago de una opción Call al vencimiento, es decir, $\max\{s - K, 0\}$

Parámetros:

- s = precio del subyacente
- K = precio de ejercicio

2. **Put(s,K)**; Calcula el pago de una opción Put al vencimiento, es decir, $\max\{K - s, 0\}$

Parámetros:

- s = precio del subyacente
- K = precio de ejercicio

3. **Stripay(nudos,resultante)**; Calcula la cartera de opciones Put y Call que replica el perfil determinado por el vector “nudos” y lo guarda en la matriz “resultante”.

Parámetros:

- nudos = el vector de puntos de donde el perfil que queremos replicar tiene picos.
- resultante = nombre de la matriz donde queremos que se guarde el resultado.

4. **CheckNA(pagos,precios)**; Informa si el mercado finito de un período determinado por los vectores de “pagos” y “precios” satisface la condición de no arbitraje y si es o no completo.

Parámetros:

- pagos = el vector de pagos de los activos.
- precios = el vector de precios de los activos.

¹Interactive **D**inamical **E**nvironment for **A**dvanced **L**earning. Programa informático escrito en Maple V complementario del libro: Prisman, E. Z., 2000. Pricing derivative securities. An interactive dynamic environment with Maple V and Matlab. Academic Press, Inc., San Diego

5. **Narbit(pagos,precios)**; Informa si el mercado finito de un período determinado por los vectores de “precios” y “pagos” satisface la condición de no arbitraje y si es o no completo. En caso afirmativo, calcula un vector de precios de estado, la correspondiente medida de martingala, el tipo de interés en el mercado y define la función de valoración **Vdis(contrato)**; La función **Vdis(contrato)**; devuelve el precio del pago “contrato” valorado con el vector de precios de estado calculado.

Parámetros:

- pagos = el vector de pagos de los activos.
- precios = el vector de precios de los activos.

6. **Valucash(pagos,precios,contrato)**; Informa si el mercado finito de un período determinado por los vectores de “precios” y “pagos” satisface la condición de no arbitraje y si es o no completo. Si el pago “contrato” es alcanzable devuelve su precio por arbitraje. Si el pago “contrato” no es alcanzable, devuelve las cotas superior e inferior del precio por arbitraje.

Parámetros:

- pagos = el vector de pagos de los activos.
- precios = el vector de precios de los activos.
- contrato = el vector con el perfil que queremos valorar.

7. **BinTree(n)**; Dibuja un árbol binomial de “n” períodos.

Parámetros:

- n = número de períodos.

8. **AllWays(n,m)**; Dibuja todas las trayectorias en un árbol binomial de “n” períodos que finalizan en el nodo (n, m) .

Parámetros:

- n = número de períodos.
- m = altura en el árbol binomial.

9. **Valbin(N,función,T,P, σ ,r)**; Precio del contrato dado por “función” calculado mediante un árbol binomial.

Parámetros:

- N = número de períodos.
- función = pago que se quiere valorar. Sintaxis: $x \rightarrow f(x)$. Ejemplo: $x \rightarrow Call(x, 10)$.
- T = vencimiento.

- P = precio actual del subyacente.
- σ = volatilidad del árbol binomial.
- r = tipo de interés.

10. **Normalpdf**(x, μ, σ); Función de densidad de una variable aleatoria normal de media μ y varianza σ^2 .

Parámetros:

- x = variable.
- μ = media.
- σ^2 = varianza.

11. **Normalcdf**(x, μ, σ); Función de distribución de una variable aleatoria normal de media μ y varianza σ^2 .

Parámetros:

- x = variable.
- μ = media.
- σ^2 = varianza.

12. **Gmbm**($P, t_0, T, N, \mu, \sigma$); Simulación del movimiento browniano del precio de un activo.

Parámetros:

- P = Precio del activo en t_0 .
- t_0 = Instante inicial.
- T = Instante final.
- N = Número de etapas.
- μ = Rentabilidad media del activo.
- σ = Volatilidad del activo.

13. **Lnrn**(x, T, P, μ, σ); Función de densidad de la variable log-normal de los precios de un activo.

Parámetros:

- x = variable.
- T = Instante futuro.
- P = Precio actual del activo.
- μ = Rentabilidad media del activo.

- σ = Volatilidad del activo.

14. **BstCall(K,T,P, σ ,r)**; Valor de una opción Call Europea calculado por la fórmula de Black-Scholes.

Parámetros:

- K = precio de ejercicio.
- T = Vencimiento.
- P = Precio actual del activo.
- σ = Volatilidad del activo.
- r = Tipo de interés.

15. **BstPut(K,T,P, σ ,r)**; Valor de una opción Put Europea calculado por la fórmula de Black-Scholes.

Parámetros:

- K = precio de ejercicio.
- T = Vencimiento.
- P = Precio actual del activo.
- σ = Volatilidad del activo.
- r = Tipo de interés.

16. **PayoffCost(nudos,T,P, σ ,r)**; Calcula el precio por Black-Scholes de la cartera de opciones Put y Call que replica el perfil determinado por el vector “nudos” determinada por el procedimiento **Stripay**.

Parámetros:

- nudos = el vector de puntos de donde el perfil que queremos replicar tiene picos.
- T = Vencimiento.
- P = Precio actual del activo.
- σ = Volatilidad del activo.
- r = Tipo de interés.

17. **Binomvsln(N,T,P, σ ,r,opción)**; Dibuja en la misma ventana las funciones de densidad (si opción=0) o de distribución (si opción=1) de una distribución binomial y una log-normal.

Parámetros:

- N = Número de etapas en la binomial.

- T = Vencimiento.
- P = Precio actual del activo.
- σ = Volatilidad del activo.
- r = Tipo de interés.
- opción = 0 (densidad) 1 (distribución).

18. **Simudif(P,t₀,T,N,μ,σ,“plot”)**; Simulación de las trayectorias de un proceso de difusión dado por la ecuación diferencial estocástica

$$dP = \mu(P, t)dt + \sigma(P, t)dB$$

Parámetros:

- P = Valor inicial en t_0 .
- t_0 = Instante inicial.
- T = Instante final.
- μ = Una función $\mu(s, t)$ (utilízense estas letras). Ejemplo: $2 * s + t$.
- σ = Una función $\sigma(s, t)$ (utilízense estas letras). Ejemplo: $s^2 + 0.5 * t$.
- “plot” = Si se omite esta opción, se realiza una animación y no un dibujo estático.

19. **ItosLemma(μ,σ,función,GBMmu,GBMsigma)**; Si X sigue la ecuación estocástica

$$dX = \mu(x, t)dt + \sigma(x, t)dB$$

este procedimiento nos devuelve la ecuación estocástica que rige la evolución de $f(x)$, donde f es la especificada en “función”. En concreto,

$$d(f(x)) = GBMmu(x, t)dt + GBMsigma(x, t)dB$$

y guarda los términos correspondientes en las variables cuyos nombres especificamos.

Parámetros:

- μ = Una función $\mu(x, t)$ (utilízense estas letras). Ejemplo: $2 * x + t$.
- σ = Una función $\sigma(x, t)$ (utilízense estas letras). Ejemplo: $x^2 + 0.5 * t$.
- función = La función $f(x)$. Sintaxis: $x \rightarrow f(x)$.
- GBMmu = nombre de la variable donde se guarda el término en dt .
- GBMsigma = nombre de la variable donde se guarda el término en dB .

20. **ImpliedVol(C,P,K,T,r)**; Estimación de la volatilidad implicada por la fórmula de Black-Scholes de valoración de opciones Call Europeas.

Parámetros:

- C = precio de la opción Call.
- P = Precio actual del subyacente.
- K = Precio de ejercicio.
- T = Vencimiento.
- r = tipo de interés.

21. **IVtry(a,b,K,T,P,r,ε,C)**; Estimación de la volatilidad implicada por la fórmula de Black-Scholes de valoración de opciones Call Europeas.

Parámetros:

- a = Cota inferior.
- b = Cota superior.
- K = Precio de ejercicio.
- T = Vencimiento.
- P = Precio actual del subyacente.
- r = tipo de interés.
- ε = Tolerancia.
- C = precio de la opción Call.