Nicolas Maldonado Baracaldo
201423809
n.maldonado10@uniandes.edu.co

Introducción a los Modelos Matemáticos en Gestión Financiera (MATE2714)

Rene Joaquin Meziat Velez
Proyecto 3

4 de octubre de 2022

1. *Elabore un programa de computadora que simule el comportamiento de un modelo de árbol binomial para precios de activos. Éste debe recibir como parámetros el número de periodos, el factor de valorización, el factor de depreciación, sus respectivas probabilidades y un precio inicial. El programa debe mostrar la evolución del precio del activo en los diferentes periodos.*

   Sol. Se escribió en `Wolfram Mathematica 12.1` la siguiente función que genera contenido dinámico

   Manipulate[$S = \{$**S1**$\}$; **SeedRandom**[2248];

   For[$n = 1, n < T, n{+}{+}$, **ran** = **RandomVariate**[**BernoulliDistribution**[p]]];

   If[**ran** == 1, **AppendTo**[$S, S[[-1]]u$], **AppendTo**[$S, S[[-1]]d$]]];

   ListLinePlot [S, **Axes** → **False**, **Frame** → **True**, **FrameLabel** → {**"Periodo (n)"**, **"Precio (**S_n**)"**},

   ImageSize → **Full**, **PlotRange** → **All**],

   {{**S1**, 4, **"**S_1**(Precio Inicial)"**}}, {{p, 0,5, **"p (Probabilidad de Valorización)"**}, 0,01, 0,99},

   {{u, 2, **"u (Factor de Valorización)"**}}, {{d, $\frac{1}{u}$, **"d (Factor de Depreciación)"**}, 0, u},

   {{T, 10, **"N (Número de Periodos)"**}, 10, 1000, 1}, **TrackedSymbols** :→ {**S1**, p, u, d, T}]

   cuyo resultado es una ventana que permite modificar todos los parámetros y genera la gráfica resultante de dicho proceso como puede observarse en la figura 1

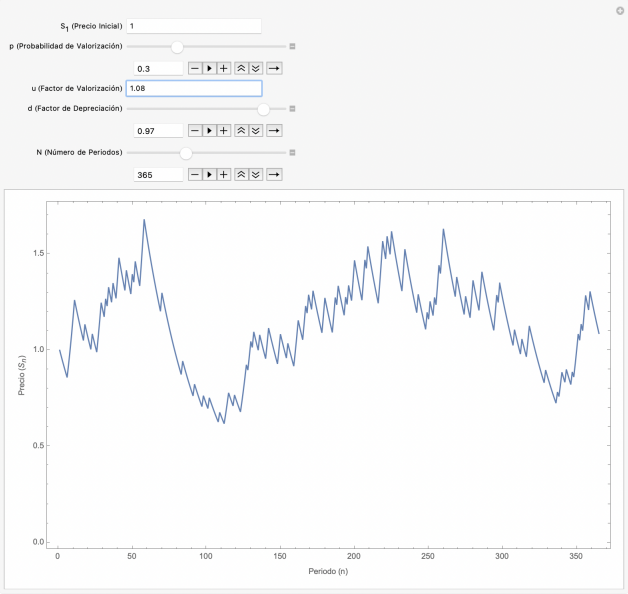

Figura 1: Visualización del modelo de árbol binomial. Puede verse cómo los distintos parámetros son modificables por el usuario así como la gráfica que se genera.

2. *Elabore un programa como el anterior, de manera que el usuario introduzca la tasa de interés y los factores de valorización y depreciación. Con ellos el computador debe realizar la simulación empleando la probabilidad de riesgo neutral del modelo. Evidentemente su programa debe rechazar datos que no correspondan a la condición de no arbitraje.*

   Sol. Se escribió en `Wolfram Mathematica 12.1` la siguiente función que genera contenido dinámico

   Manipulate[If[$d < 1 + r < u, S = \{S1\}; p = \frac{1+r-d}{u-d}$; SeedRandom[2248];

   For[$n = 1, n < T, n{+}{+}$, ran = RandomVariate[BernoulliDistribution[p]];

   If[ran == 1, AppendTo[$S, S[[-1]]u$], AppendTo[$S, S[[-1]]d$]];

   ListLinePlot [S, Axes → False, Frame → True, FrameLabel → {"Periodo (n)", "Precio (S_n)"} , ImageSize → Full] ,

   Catch[Throw["Los parámetros introducidos no corresponden a la condición de no arbitraje"]]],

   {{S1, 4, "S_1 (Precio Inicial)"}} , {{r, 0,25, "r (Tasa de Interés)"}},

   {{u, 2, "u (Factor de Valorización)"}}, {{d, $\frac{1}{u}$, "d (Factor de Depreciación)"} , 0, u} ,

   {{T, 10, "N (Número de Periodos)"}, 10, 1000, 1}, TrackedSymbols :→ {S1, u, d, T}]

   cuyo resultado es una ventana que permite modificar todos los parámetros y genera la gráfica resultante de dicho proceso como puede observarse en la figura 2, además arroja un mensaje cuando los parámetros ingresados no corresponden con la condición de no arbitraje como puede observarse en la figura 3

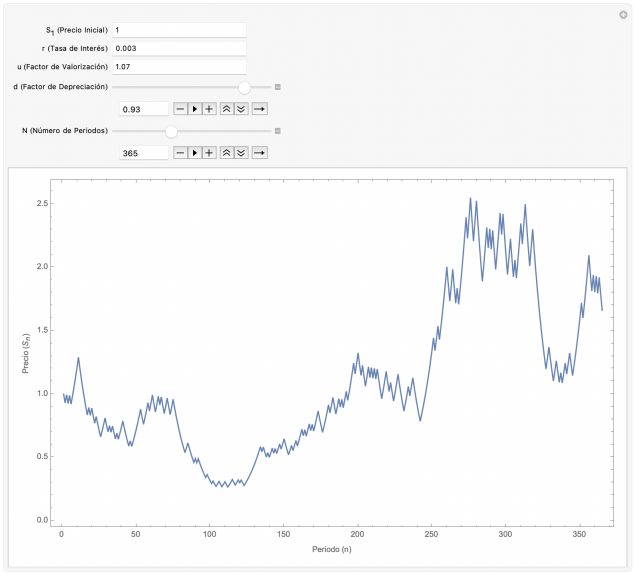

Figura 2: Visualización del modelo de árbol binomial con probabilidad de riesgo neutral. Puede verse cómo los distintos parámetros son modificables por el usuario así como la gráfica que se genera.

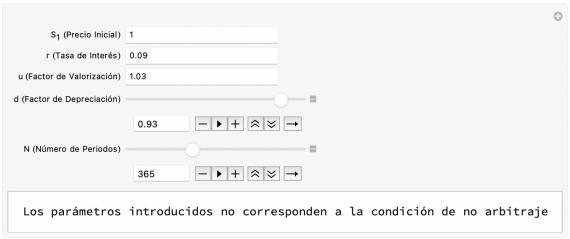

Figura 3: Mensaje de error cuando se introducen parámetros que no corresponden con la condición de no arbitraje.

3. *Elabore un programa de computadora que simule un camino aleatorio simétrico M_n, luego utilícelo para simular la evolución de precios dada por la fórmula:*

$$S_n = e^{\sigma M_n} \left(\frac{2}{e^{\sigma} + e^{-\sigma}} \right)^n .$$

El usuario debe proporcionar el parámetro σ.

Sol. Se escribió en `Wolfram Mathematica 12.1` la siguiente función que genera contenido dinámico

Manipulate[$M = \{0\}$; **SeedRandom**[2248];

For[$t = 1, t < T, t{+}{+}$, ran = **RandomVariate**[**BernoulliDistribution**[0,5]];

If[ran == 1, **AppendTo**[$M, M[[-1]] + 1$], **AppendTo**[$M, M[[-1]] - 1$]]];

S = **Table** $\left[e^{\sigma M[[n]]} \left(\frac{2}{e^{\sigma}+e^{-\sigma}} \right)^n, \{n, 1, T\} \right]$;

ListLinePlot [S, **Axes** → **False**, **Frame** → **True**, **FrameLabel** → {"Periodo (n)", "Precio (S_n)"},

ImageSize → **Full**, **PlotRange** → **All**] ,

{{σ, 0,5, "σ"}}, {{T, 10, "N (Número de Periodos)"}, 10, 1000, 1}, **TrackedSymbols** :→ {σ, T}]

cuyo resultado es una ventana que permite modificar todos los parámetros y genera la gráfica resultante de dicho proceso como puede observarse en la figura 4

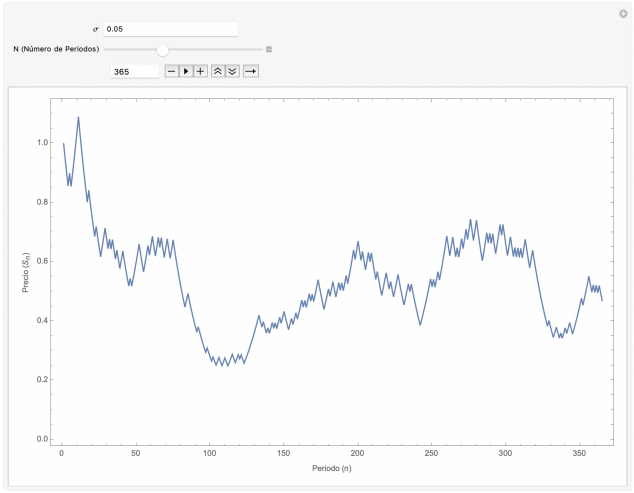

Figura 4: Visualización del modelo de caminata aleatoria simétrica. Puede verse cómo los distintos parámetros son modificables por el usuario así como la gráfica que se genera.

4. *Repita el ejercicio anterior para un caso general que admita caminos aleatorios no simétricos (deriva hacia arriba y hacia abajo) donde el usuario debe dar las probabilidades de ascenso y descenso.*

   Sol. Se escribió en `Wolfram Mathematica 12.1` la siguiente función que genera contenido dinámico

   $$\textbf{Manipulate}[M = \{0\}; \textbf{SeedRandom}[2248];$$

   $$\textbf{For}[t = 1, t < T, t{+}{+}, \text{ran} = \textbf{RandomVariate}[\textbf{BernoulliDistribution}[p]];$$

   $$\textbf{If}[\text{ran} == 1, \textbf{AppendTo}[M, M[[-1]] + 1], \textbf{AppendTo}[M, M[[-1]] - 1]]];$$

   $$S = \textbf{Table}\left[e^{\sigma M[[n]]}\left(\tfrac{2}{e^{\sigma}+e^{-\sigma}}\right)^{n}, \{n, 1, T\}\right];$$

   $$\textbf{ListLinePlot}\,[S, \textbf{Axes} \to \textbf{False}, \textbf{Frame} \to \textbf{True}, \textbf{FrameLabel} \to \{\text{“Periodo (n)”}, \text{“Precio } (S_n)\text{”}\},$$

   $$\textbf{ImageSize} \to \textbf{Full}, \textbf{PlotRange} \to \textbf{All}],$$

   $$\{\{p, 0{,}5, \text{“p (Probabilidad de Ascenso)”}\}, 0{,}01, 0{,}99\}, \{\{\sigma, 0{,}5, \text{“}\sigma\text{”}\}\},$$

   $$\{\{T, 10, \text{“N (Número de Periodos)”}\}, 10, 1000, 1\}, \textbf{TrackedSymbols} :\to \{p, \sigma, T\}]$$

   cuyo resultado es una ventana que permite modificar todos los parámetros y genera la gráfica resultante de dicho proceso como puede observarse en la figura 5

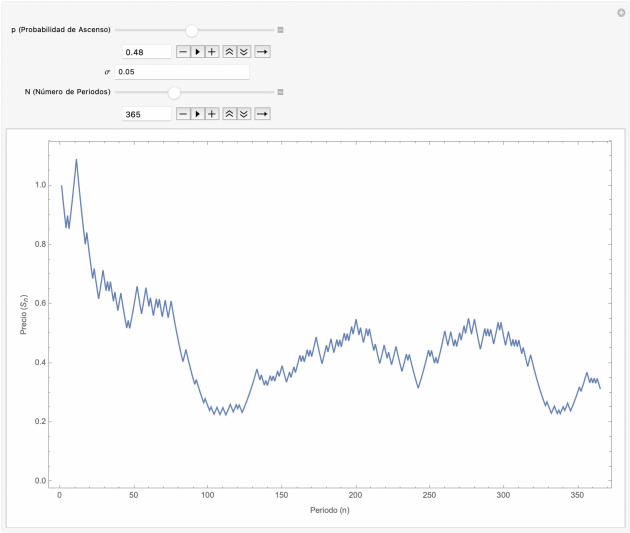

Figura 5: Visualización del modelo de caminata aleatoria no simétrica. Puede verse cómo los distintos parámetros son modificables por el usuario así como la gráfica que se genera.

5. *Utilizando los algoritmos anteriores proponga un módulo que le permita simular tiempos de parada para modelos binomiales con periodos finitos y tiempos de alcance para valores prefijados en modelos basados en caminos aleatorios.*

   Sol.

6. *Realice más de 10 simulaciones para cada caso y describa sus observaciones. En especial describa las diferencias observadas entre los modelos del punto 1) y del punto 3).*

   Sol. Se realizaron varias simulaciones con cada uno de los módulos, no todas las cuales vale la pena poner acá pues ocuparían mucho espacio, sin embargo son particularmente interesantes algunas observaciones.

   - Como era de esperarse, los módulos desarrollados para los ejercicios *(1)* y *(2)* generan el mismo comportamiento siempre que se introduzcan la probabilidad y la tasa de interés correctas (i.e., tales que la probabilidad de riesgo neutral calculada con dicha tasa de interés sea la misma probabilidad introducida directamente en *(1)*), un ejemplo tal puede verse en la figura 6.

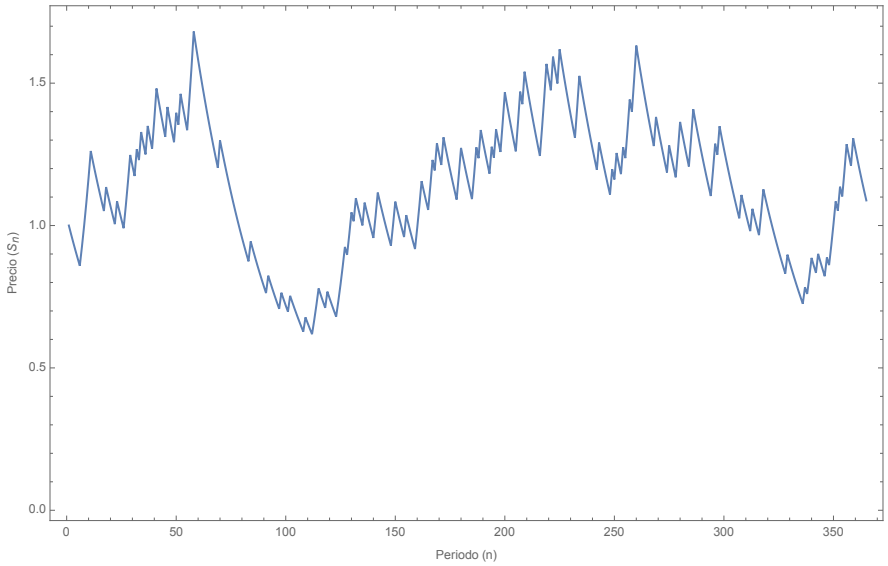

Figura 6: Comportamiento para 365 periodos generado tanto por el primer como por el segundo módulo usando $u = 1{,}08, d = 0{,}97$ en ambos y $p = 0{,}3$ en el primero y $r = 0{,}0018$ en el segundo.

- El comportamiento generado por el módulo desarrollado para el ejercicio *(3)*, el cual se basa en una caminata aleatoria simétrica, es exactamente el comportamiento que se observa en el modelo de árbol binomial (primer módulo) cuando la probabilidad de valorización es $0{,}5$ y cuando se toma $u = 1 + \sigma, d = 1 - \sigma$, un ejemplo de esto puede verse en la figura 7.

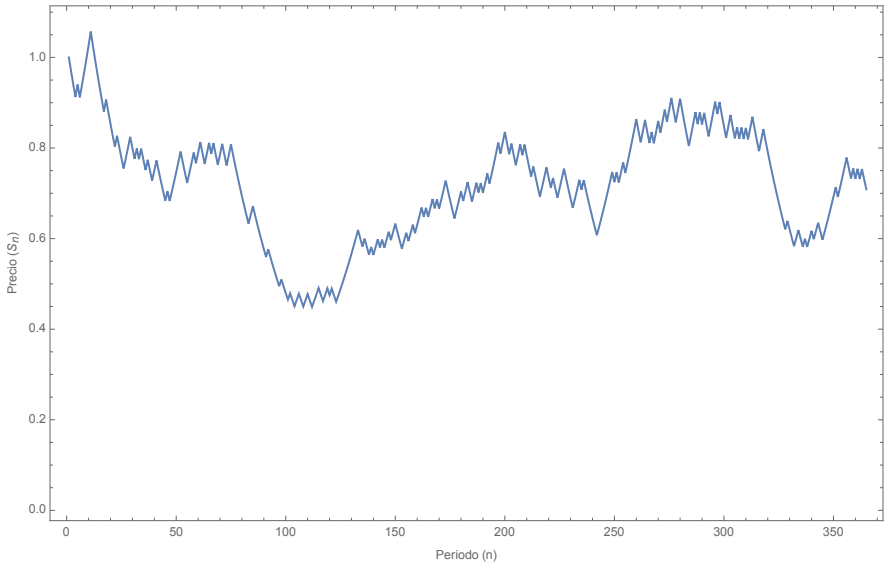

Figura 7: Comportamiento para 365 periodos generado tanto por el primer como por el tercer módulo usando $u = 1{,}03, d = 0{,}97, p = 0{,}5$ en el primero y $\sigma = 0{,}03$ en el tercero.

- El comportamiento generado por el módulo desarrollado para el ejercicio *(4)*, el cual se basa en una caminata aleatoria no simétrica, es exactamente el comportamiento que se observa en el modelo de árbol binomial (primer módulo) cuando la probabilidad de valorización es igual a la probabilidad de ascenso y cuando se toma $u = 1 + \sigma, d = 1 - \sigma$, un ejemplo de esto puede verse en la figura 8.

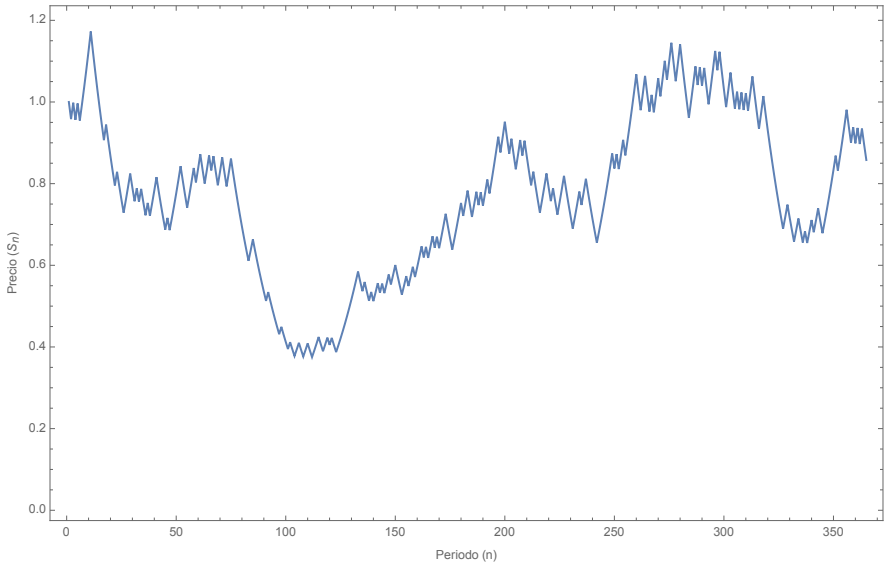

Figura 8: Comportamiento para 365 periodos generado tanto por el primer como por el cuarto módulo usando $u = 1{,}042, d = 0{,}958, p = 0{,}51$ en el primero y $\sigma = 0{,}042, p = 0{,}51$ en el cuarto.

7. *Desarrolle un módulo que le permita al usuario introducir una gestión del portafolio tipo delta para visualizar procesos de capitalización regidos por la fórmula*

$$X_{n+1} = \Delta_n S_{n+1} + (1+r)(X_n - \Delta_n S_n).$$

En otras palabras, realice un algoritmo que (con base en la simulación para la serie de precios S_n) le permita al usuario ingresar las operaciones Δ_n para obtener la serie de capital acumulado X_n. Su algoritmo debe permitirle al usuario saber cuándo debe apalancar recursos, en otras palabras debe alertar aquellas situaciones en que $Xn - \Delta_n S_n < 0$.

Sol. Se escribió en `Wolfram Mathematica 12.1` la siguiente función que genera contenido dinámico

Manipulate[$T =$ **Length**[Δ]; **SeedRandom**[2248];

If[$d < 1 + r < u, S =$ {**S1**}; $p = \frac{1+r-d}{u-d}$;

For[$n = 1, n < T, n{+}{+}$, **ran** = **RandomVariate**[**BernoulliDistribution**[p]]];

If[**ran** == 1, **AppendTo**[$S, S[[-1]]u$], **AppendTo**[$S, S[[-1]]d$]],

Catch[**Throw**[**"Los parámetros introducidos no corresponden a la condición de no arbitraje"**]]];

$X =$ {$S[[1]]$}; **For**[$n = 1, n < T, n{+}{+}$, **If**[$X[[n]] - \Delta[[n]]S[[n]] < 0$, **Echo**[$n$, **"Apalancar en "**];

AppendTo[$X, \Delta[[n]]S[[n+1]] + (1+r)(X[[n]] - \Delta[[n]]S[[n]])$],

AppendTo[$X, \Delta[[n]]S[[n+1]] + (1+r)(X[[n]] - \Delta[[n]]S[[n]])$]]];

ListLinePlot [X, **Axes** $\rightarrow$ **False**, **Frame** $\rightarrow$ **True**, **FrameLabel** $\rightarrow$ {**"Periodo (n)"**, **"Capital Acumulado (X_n)"**} ,

ImageSize $\rightarrow$ **Full**, **PlotRange** $\rightarrow$ **All**] ,

{{**S1**, 4, **"S_1 (Precio Inicial)"**}} , {{r, 0,05, **"r (Tasa de Interés)"**}}, {{u, 2, **"u (Factor de Valorización)"**}},

{{$d, \frac{1}{u}$, **"d (Factor de Depreciación)"**} , 0, u} , {{Δ, {0,5, 0,4, 0,3, 0,6, 0,7, 0,8},

"Δ (Operaciones Δ_n, introducir entre corchetes y separándo los números por comas)"}} ,

TrackedSymbols :$\rightarrow$ {**S1**, r, σ, Δ}]

cuyo resultado es una ventana que permite modificar todos los parámetros y genera la gráfica resultante de dicho proceso, además de que imprime mensajes indicando los periodos en los que se debe apalancar recursos, como puede observarse en la figura 9

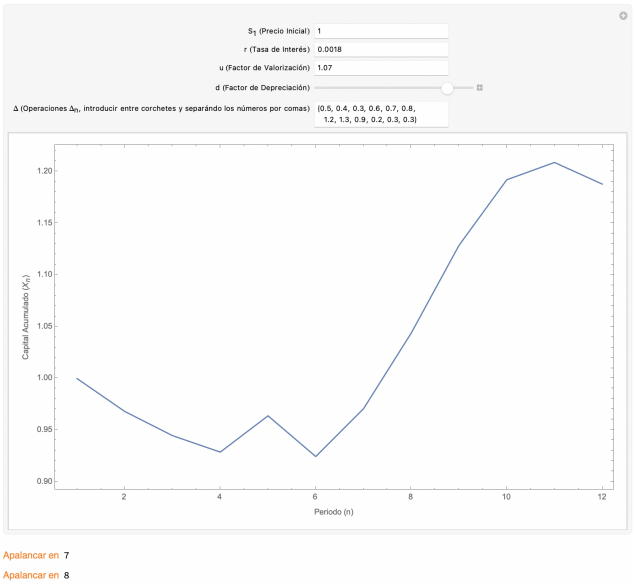

Figura 9: Visualización de gestión de portafolio *delta*. Puede verse cómo los distintos parámetros son modificables por el usuario así como la gráfica que se genera y los mensajes de alerta.

8. *Elabore un módulo que reciba la función de pagos de un derivado tipo europeo $V_N = V_N(\omega_1, \ldots, \omega_N)$, los parámetros de un modelo binomial y que devuelva la valoración del derivado en los periodos intermedios $n = 0, \ldots, N-1$.*

   Sol. Se escribió en `Wolfram Mathematica 12.1` la siguiente función que genera contenido dinámico

$$\textbf{Manipulate}[V = \{\text{VN}\}; \textbf{SeedRandom}[2248];$$
$$\textbf{For}\left[i = 1, i < T, i{+}{+}, \textbf{PrependTo}\left[V, \frac{\textbf{Expectation}\left[V[[1]], n\textbf{BinomialDistribution}\left[T-i, \frac{1+r-d}{u-d}\right]\right]}{1+r}\right]\right];$$
$$\textbf{ListLinePlot}[V, \textbf{Axes} \to \textbf{False}, \textbf{Frame} \to \textbf{True}, \textbf{FrameLabel} \to \{\text{“Periodo } (n)\text{”}, \text{“Precio } (V_n)\text{”}\},$$
$$\textbf{ImageSize} \to \textbf{Full}, \textbf{PlotRange} \to \textbf{All}], \left\{\left\{\text{VN}, 4u^n d^{T-n}, \text{“}V_N \text{ (Función de Pagos)”}\right\}\right\},$$
$$\{\{u, 2, \text{“u (Factor de Valorización)”}\}\}, \left\{\left\{d, \frac{1}{u}, \text{“d (Factor de Depreciación)”}\right\}, 0, u\right\},$$
$$\{\{r, 0{,}05, \text{“r (Tasa de Interés)”}\}\}, \{\{T, 10, \text{“Número de Periodos”}\}, 10, 1000, 1\}, \textbf{TrackedSymbols} :\to \{\text{VN}, u, d, r, T\}]$$

   cuyo resultado es una ventana que permite modificar todos los parámetros y genera la gráfica resultante de dicho proceso como puede observarse en la figura 10

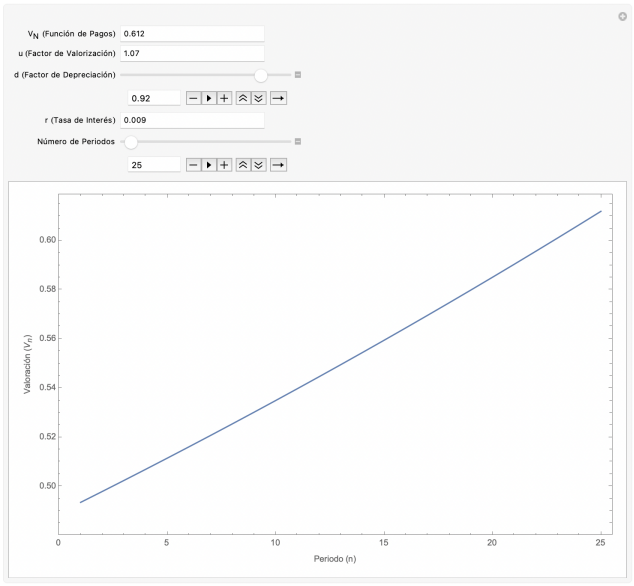

Figura 10: Visualización de valoración de un derivado dada una función de pagos. Puede verse cómo los distintos parámetros son modificables por el usuario así como la gráfica que se genera.

9. *Complete el punto anterior para que el módulo correspondiente proporcione al usuario la operación delta que replica el derivado en cuestión. Esta es:*

$$\Delta_n(\omega_1,\ldots,\omega_n) = \frac{V_{n+1}(\omega_1,\ldots,\omega_n u) - V_{n+1}(\omega_1,\ldots,\omega_n d)}{S_{n+1}(\omega_1,\ldots,\omega_n u) - S_{n+1}(\omega_1,\ldots,\omega_n d)}$$

Introduzca estos datos en el punto 7) para verificar que efectivamente replica el derivado.

Sol. Se le añadió al módulo anterior la instrucción

$$\mathbf{\Delta rep = Table}\left[\frac{uV[[i]]-dV[[i]]}{uS[[i]]-dS[[i]]}, \{i, 1, T\}\right]$$

que guarda en una variable **Δrep** la estrategia de replicación, la cual al introducirse en el módulo del ejercicio *(7)* efectivamente replica el derivado, como puede verse en las figuras 11 y 12.

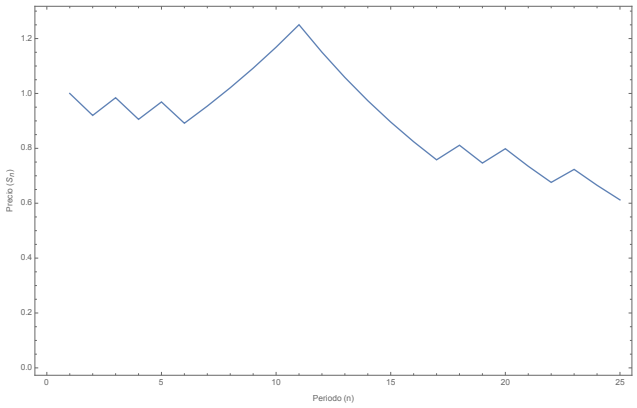

Figura 11: Comportamiento del derivado según simulación con módulo de ejercicio (2).

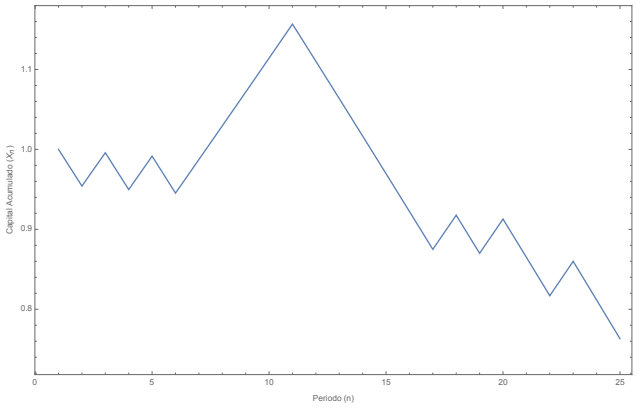

Figura 12: Comportamiento del portafolio *delta* usando estrategia de replicación.

10. *Utilice estadística descriptiva sobre varias simulaciones con los resultados del punto 1) para mostrar que* $\frac{S_n}{(1+r)^n}$ *es una martingala.*

Sol. Se realizaron varias simulaciones usando el módulo del ejercicio *(2)* para 365 periodos y luego se calculó directamente el valor esperado de $\frac{S_{n+1}}{(1+r)^{n+1}}$ como

$$\mathbb{E}\left[\frac{S_{n+1}}{(1+r)^{n+1}}\right] = \frac{p^* u S_n + (1-p^*) d S_n}{(1+r)^{n+1}},$$

tras lo cual se calculó la media de las desviaciones entre éstos y el valor en la simulación de $\frac{S_n}{(1+r)^n}$. Se esperaba que estas desviaciones fuesen 0 pues para una martingala

$$\mathbb{E}\left[\frac{S_{n+1}}{(1+r)^{n+1}}\right] = \frac{S_n}{(1+r)^n}.$$

En efecto estas desviaciones estuvieron siempre muy cerca de 0, en el orden de 10^{-17}, lo cual se cree que se debe a pequeños errores en distintos puntos de los cálculos al ser éstos enteramente numéricos, por lo cual tienen una precisión máxima.