



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# Repos: prestamos con colateral riesgoso

Ivonne Lorena Guevara Cortina

Universidad Nacional de Colombia  
Departamento de Matemáticas  
Bogotá, Colombia  
2017

# Repos: prestamos con colateral riesgoso

Ivonne Lorena Guevara Cortina

Trabajo de grado presentada como requisito parcial para optar al título de:  
**Magister en Actuaría y Finanzas**

Director:  
Oscar Javier López Alfonso

Línea de Investigación:  
Finanzas

Universidad Nacional de Colombia  
Departamento de Matemáticas  
Bogotá, Colombia  
2017

## Resumen

Las opciones Repo son préstamos a corto plazo que sirven para satisfacer necesidades inmediatas de liquidez. En este tipo de contratos, el deudor dará un paquete de acciones como garantía. En cualquier momento, antes de la fecha de vencimiento del préstamo, dependiendo del valor de las acciones, el poseedor de un Repo podría tener incentivos para pagar la deuda más los intereses y terminar el contrato anticipadamente, recuperando sus acciones. Este es el riesgo que modelaremos desde el punto de vista de las matemáticas financieras, utilizando una opción americana de compra con precio de ejercicio variable. En primer lugar, haremos la modelación matemática del contrato bajo el modelo de Black-Scholes dependiendo del trato que se les da a los dividendos que paga la acción, dando como resultado un sistema de ecuaciones diferenciales parciales con condiciones de frontera. A continuación, proponemos un método numérico para la aproximación de la solución de estos sistemas de ecuaciones, realizaremos un análisis de sensibilidad de los principales parámetros involucrados en las operaciones de este tipo y por último una aplicación al caso colombiano.

# Índice

<b>1</b>	<b>Introducción</b>	<b>7</b>
<b>2</b>	<b>Modelación matemática de un contrato repo</b>	<b>8</b>
2.1	Repos sin dividendos . . . . .	9
2.2	Repos con dividendos recibidos por el banco . . . . .	10
2.3	Repos con dividendos reinvertidos . . . . .	10
2.4	Repos con dividendos retornados al cliente durante el tiempo del contrato . .	11
<b>3</b>	<b>Metodología</b>	<b>12</b>
3.1	El método de Crank-Nicolson . . . . .	12
3.2	Método SOR . . . . .	14
3.3	Método SOR Proyectada . . . . .	15
3.4	Frontera eficiente . . . . .	16
<b>4</b>	<b>Implementación del método numérico y análisis del contrato</b>	<b>17</b>
4.1	Comparación con Prager . . . . .	18
4.2	Análisis de los contratos Repo . . . . .	19
4.3	Frontera de ejercicio . . . . .	21
4.4	Análisis de sensibilidad . . . . .	22
<b>5</b>	<b>Aplicación al mercado colombiano</b>	<b>25</b>
<b>6</b>	<b>Conclusiones</b>	<b>28</b>
	<b>Anexos</b>	<b>30</b>
<b>A</b>	<b>Algoritmo SOR</b>	<b>30</b>
<b>B</b>	<b>Algoritmo SOR proyectada</b>	<b>31</b>
<b>C</b>	<b>Contrato Repo sin dividendos</b>	<b>32</b>
<b>D</b>	<b>Contrato Repo con dividendos recibido por el banco</b>	<b>34</b>
<b>E</b>	<b>Contrato Repo con dividendos entregados al cliente</b>	<b>37</b>
<b>F</b>	<b>Contrato Repo con dividendos reinvertidos</b>	<b>39</b>
<b>G</b>	<b>Algoritmo frontera de ejercicio</b>	<b>42</b>

# Índice de figuras

Figura 1	Grafica de $\omega$ respecto al numero de iteraciones algoritmo SOR con $S_0 = 50; K = 40; \sigma = 0,5; \gamma = 0,1; r = 0,05; T = 1$ . . . . .	15
Figura 2	Grafico aproximación de $S_f(t)$ usando diferencias finitas . . . . .	17
Figura 3	Valor de la prima $C$ variando $\alpha$ con $S_0 = 100, \sigma = 0,25, \gamma = 0,12,$ y $\delta = 0,02.$ . . . . .	20
Figura 4	Valor de la prima $C$ variando $\alpha$ con $T = 3$ meses para los cuatro contratos. . . . .	21
Figura 5	Frontera de ejercicio. . . . .	21
Figura 6	Graficas análisis de sensibilidad con respecto a $\gamma.$ . . . . .	23
Figura 7	Graficas prima y sensibilidad con respecto a $\delta$ . . . . .	24

# Índice de tablas

Tabla 1	Comparación con Prager utilizando los parámetros: $S_0 = 50$ , $r = 5\%$ y $T = 1$ . . . . .	18
Tabla 2	Comparación $SOR$ y $SOR$ proyectada bajo los parámetros: $\omega = 1,1$ , $S_0 = 50$ , $r = 5\%$ y $T = 1$ . . . . .	19
Tabla 3	Mayores valores de $\alpha$ para los cuales la prima $C$ es cero. . . . .	20
Tabla 4	Máximos gráfica de sensibilidad contrato Repo con dividendos entregados vs $\delta$ . . . . .	24
Tabla 5	Acciones que generan dividendos. . . . .	25
Tabla 6	Acciones que no generan dividendos. . . . .	26
Tabla 7	Valor de la prima $C$ contrato sin dividendos, $\gamma = 12\%$ , $r = 7,5\%$ . . . . .	26
Tabla 8	Valor de la prima $C$ por contratos, $\gamma = 12\%$ , $r = 7,5\%$ . . . . .	27

# 1. Introducción

Las operaciones Repo (*stock loans*) son préstamos a corto plazo entre dos partes, un inversionista y un prestamista, que le sirven al deudor de este préstamo (el inversionista) para obtener efectivo (liquidez) entregando como garantía (colateral) acciones de su propiedad al prestamista. El inversionista recibe el dinero del préstamo solicitado, una tasa de interés y un plazo para pagar este, a cambio de entregar las acciones como garantía del préstamo. En Colombia a estas operaciones se le adiciona la condición de pacto de recompra, lo cual obliga al inversionista a pagar el préstamo más los intereses en la fecha de finalización del préstamo.

Nótese que, si el inversionista pudiera anticipar el pago del préstamo y recuperar sus acciones, este lo haría si el precio de las acciones sube y es mayor que el valor del préstamo más los intereses, obteniendo una ganancia vendiendo las acciones. Por el contrario, si las acciones caen, el cliente preferirá no pagar el préstamo más los intereses y esperar a que las acciones suban o a la finalización del contrato. Aun así, en la finalización del contrato el cliente podría tomar la decisión de pagar o no el préstamo, en cuyo último caso perdería sus acciones. En este trabajo queremos analizar el riesgo de anticipación del pago del préstamo en cualquier instante entre el desembolso del préstamo y la fecha de finalización del mismo, para esto tenemos que suprimir la condición de pacto de recompra de las operaciones Repo.

Este riesgo lo vamos a modelar desde el punto de vista de las matemática financieras, modelándolo como una opción americana de compra con precio de ejercicio variable. Los primeros en hacer esto, fueron Xia y Zou [XZ07] bajo el modelo de Black-Scholes-Merton, encontrando una solución analítica suponiendo que la duración del contrato es infinita. Los siguientes, fueron Dai y Xu [DX11], quienes modelaron este riesgo dependiendo del trato que se le da a los dividendos que pagan las acciones dejadas como colateral en un contrato con duración finita, sin embargo, la implementación de sus contratos fue hecha en el modelo discreto de Cox-Ross-Rubinstein.

Se pretende implementar la modelación del contrato Repo en un tiempo finito, aplicando un método numérico continuo, teniendo en cuenta el trato que se le da a los dividendos (sugeridos por Dai y Xu [DX11]), encontrando un sistema de ecuaciones diferenciales variacionales con problemas de frontera, que se solucionara usando los métodos de Cranck-Nicolson y los algoritmos SOR y SOR proyectada, esta valoración del riesgo se programara y se aplicara al caso Colombiano con algunas acciones listadas en la Bolsa de Valores de Colombia.

En la sección 1, se muestra la modelación matemática de una operación Repo como una opción americana con precio de ejercicio variable, considerando cuatro contratos dependiendo del manejo que se le da a los dividendos del colateral, en la sección 2, se expone la metodología usada para la valoración del contrato: el método numérico de Cranck-Nicolson, los algoritmos SOR y SOR proyectada y el algoritmo para encontrar la frontera de ejercicio, en la sección 3, se hace la implementación del método numérico para la valoración del contrato y la prima de riesgo de pago anticipado, en la sección 4 se hace la implementación al caso colombiano y en la sección 5 se presentan las conclusiones.

## 2. Modelación matemática de un contrato repo

En esta sección bajo el modelo de Black-Scholes [BS73], se hace la modelación matemática del precio de la acción y el cálculo de la prima de un contrato repo de forma similar a Xia & Zhou [XZ07] y Dai & Xu [DX11]. Esta prima se obtiene de la valoración de una opción Americana de compra con precio de ejercicio variable como veremos a continuación.

Sea  $T (> 0)$  el tiempo de vencimiento del préstamo y sea  $(\Omega, \mathcal{F}, \mathbb{Q})$  un espacio de probabilidad completo, donde  $\mathbb{Q}$ , denota la medida de probabilidad neutral al riesgo. Denotando por  $\{S_t\}_{t \in [0, T]}$  el proceso estocástico que describe el precio de la acción y asumiendo que este satisface el modelo de Black-Scholes, entonces, se tiene que el precio del subyacente (o acción) satisface la siguiente ecuación diferencial estocástica

$$dS_t = (r - \delta)S_t dt + \sigma S_t dW_t. \quad (1)$$

Aquí, el proceso  $\{W_t\}_{t \in [0, T]}$  denota un movimiento Browniano,  $\delta (\geq 0)$  la tasa de dividendos de la acción compuesta continuamente,  $\sigma (> 0)$  la volatilidad de la acción y  $r (\geq 0)$  la tasa de interés del mercado compuesta continuamente. La solución de la ecuación (1) está dada por

$$S_t = S_0 \exp \left( \left( r - \delta - \frac{1}{2} \sigma^2 \right) t + \sigma W_t \right), \quad (2)$$

para cada  $t \in [0, T]$ , donde,  $S_0$  denota el precio inicial de la acción.

En el momento inicial ( $t = 0$ ) el cliente solicita un préstamo por un valor  $L$  y dará a cambio un número de acciones  $N$  (nocional) como garantía del préstamo (colateral), a un valor de  $S_0$  por acción. Por este préstamo, el cliente debe pagar una tasa de interés  $\gamma$  (compuesta continuamente) y una prima de riesgo por acción, denotada por  $C$ , que le da el derecho más no la obligación de recuperar sus acciones en cualquier instante de tiempo  $t$  pagando el principal, más los intereses generados hasta la fecha. Por lo tanto, el flujo inicial del cliente es

$$L - NS_0 - NC. \quad (3)$$

Un cliente racional recuperaría sus acciones si el valor de estas está por encima del valor del préstamo; en caso contrario, el preferiría esperar. Por lo tanto, si denotamos por  $K = L/N$  el valor por unidad de acción que pide prestado el cliente, la ganancia del cliente si este decide recuperar sus acciones sería  $NS_t - NKe^{\gamma t}$  o en caso contrario cero. Esta situación se puede modelar como

$$N(S_t - Ke^{\gamma t})^+, \quad (4)$$

donde  $(x)^+ = \text{máximo}\{x, 0\}$ .

En cada momento del tiempo, se establece el precio mínimo de la acción en el cual es conveniente pagar el préstamo, esto se conoce como frontera de ejercicio, su existencia, se garantiza cuando la tasa libre de riesgo es menor que la tasa de interés que cobra el banco  $r < \gamma$ , ver Proposición 2.1 en [DX11], en la práctica, siempre el banco presta a una tasa mayor que la tasa del mercado.



Por lo tanto, para el cálculo de la prima  $C$  se debe valorar una opción Americana con precio de ejercicio variable, como veremos a continuación.

## 2.1. Repos sin dividendos

Si el colateral no produce dividendos,  $\delta = 0$  en la ecuación (2), cuando el cliente toma la decisión de pagar el préstamo en un tiempo  $t$ , es decir, paga la cantidad  $Ke^{\gamma t}$  por acción y en consecuencia recupera sus acciones al precio  $S_t$ , entonces su beneficio por acción está dado por

$$(S_t - Ke^{\gamma t})^+. \quad (5)$$

Luego el precio de la opción con valor intrínseco dado por la anterior ecuación queda determinado por:

$$V(t, S_t) = \sup_{\tau \in \mathcal{T}[t, T]} \mathbb{E} [e^{-r(\tau-t)}(S_\tau - Ke^{\gamma \tau})^+ | \mathcal{F}_t] \quad (6)$$

Donde  $\{\mathcal{F}_t\}_{t \in [0, T]}$  es la filtración natural asociada al movimiento Browniano  $\{W_t\}_{t \in [0, T]}$ ,  $\mathbb{E}[\cdot | \mathcal{F}_t]$  es la esperanza bajo  $\mathbb{Q}$  condicionada a  $\mathcal{F}_t$  y  $\mathcal{T}[t, T]$  denota al conjunto de todos los tiempos de paro con valores en el intervalo  $[t, T]$ .

Si denotamos por  $S_f(t)$  la frontera de ejercicio del contrato, es decir, el mínimo valor del colateral, para cada  $t \in [0, T]$ , que cumple la desigualdad  $S_f(t) \geq Ke^{\gamma t}$ . El problema definido en (6) se puede escribir en forma equivalente como un sistema de ecuaciones diferenciales parciales con condiciones iniciales y de frontera, véase por ejemplo Lu & Putri [LP15]

$$\left\{ \begin{array}{ll} \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0, & S < S_f(t), t \in [0, T], \\ V(t, 0) = 0, & t \in [0, T], \\ V(T, S) = (S - Ke^{\gamma T})^+, & S \geq 0, \\ V(t, S) = S - Ke^{\gamma t}, & S > S_f(t), t \in [0, T], \\ V(t, S_f(t)) = S_f(t) - Ke^{\gamma t}, & \\ \frac{\partial V}{\partial S}(t, S_f(t)) = 1. & \end{array} \right. \quad (7)$$

Entonces el cálculo de la prima  $C$  se reduce a encontrar:

$$C = (K + V(0, S_0) - S_0)^+. \quad (8)$$

Esta definición difiere de Grasselli & Gómez [GG12] (ecuación (3)) en donde no aparece la función  $(\cdot)^+$ , por la siguiente observación.

**Observación 2.1.** Por lo general  $K$  es un porcentaje del valor  $S_0$  (70% en Colombia), entonces,  $K - S_0 < 0$ . Si  $V(0, S_0)$  es muy pequeño o cero, el valor de  $C$  sería negativo, lo cual no tiene ningún sentido siendo el valor de una prima (un precio).

## 2.2. Repos con dividendos recibidos por el banco

Suponga que el banco se queda con los dividendos generados por la acción hasta el tiempo de redención del préstamo, tomando estos dividendos como parte de pago del préstamo, esto reduce el valor de la prima cobrada al cliente. Entonces, este problema modelado por (6) con  $\delta > 0$  en (2) se puede escribir en forma equivalente como un sistema de ecuaciones diferenciales parciales con condiciones iniciales y de frontera, véase Lu & Putri [LP15] (ecuación (3.3))

$$\left\{ \begin{array}{l} \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - \delta)S \frac{\partial V}{\partial S} - rV = 0, \quad S < S_f(t), \quad t \in [0, T], \\ V(t, 0) = 0, \quad t \in [0, T], \\ V(T, S) = (S - Ke^{\gamma T})^+, \quad S \geq 0, \\ V(t, S) = S - Ke^{\gamma t}, \quad S > S_f(t), \quad t \in [0, T], \\ V(t, S_f(t)) = S_f(t) - Ke^{\gamma t}, \\ \frac{\partial V}{\partial S}(t, S_f(t)) = 1. \end{array} \right. \quad (9)$$

Al igual que en la anterior sección  $C$  se calcula por la ecuación (8).

## 2.3. Repos con dividendos reinvertidos

En esta sección vamos a asumir que los dividendos son reinvertidos por el banco en la compra de más colateral, hasta el tiempo de redención del préstamo. Es decir, con el dinero recibido por dividendos, el banco compra más acciones; de esta manera, si las acciones suben el cliente pagara el préstamo y recibirá sus acciones originales más las compradas con los dividendos, y si no sucede, el valor de pérdida para el banco sera menor, ya que el valor total del colateral aumenta. Entonces, el beneficio del cliente por recuperar sus acciones en el tiempo  $t$  esta dado por:

$$(e^{\delta t} S_t - Ke^{\gamma t})^+. \quad (10)$$

Luego el precio de la opción con valor intrínseco dado por la anterior ecuación queda determinado por:

$$V(t, S_t) = \sup_{\tau \in \mathcal{T}[t, T]} \mathbb{E} [e^{-r(\tau-t)} (e^{\delta t} S_\tau - Ke^{\gamma \tau})^+ | \mathcal{F}_t]. \quad (11)$$

Este problema se puede simplificar usando la transformación:

$$\hat{S}_t = e^{\delta t} S_t = S_0 \exp \left( \left( r - \frac{\sigma^2}{2} \right) t + \sigma W_t \right). \quad (12)$$

Por lo tanto, (11) se puede reescribir como:

$$\hat{V}(t, \hat{S}_t) = \sup_{\tau \in \mathcal{T}[t, T]} \mathbb{E} [e^{-r(\tau-t)} (\hat{S}_\tau - Ke^{\gamma \tau})^+ | \mathcal{F}_t]. \quad (13)$$

Donde

$$\hat{V}(t, \hat{S}_t) = V(t, S_t). \quad (14)$$

Resolver el problema dado por (13) es equivalente a resolver, veáse Lu & Putri [LP15] (ecuación (3.17))

$$\left\{ \begin{array}{l} \frac{\partial \hat{V}}{\partial t} + \frac{1}{2} \sigma^2 \hat{S}^2 \frac{\partial^2 \hat{V}}{\partial \hat{S}^2} + r \hat{S} \frac{\partial \hat{V}}{\partial \hat{S}} - r \hat{V} = 0, \quad \hat{S} < \hat{S}_f(t), \quad t \in [0, T], \\ \hat{V}(t, 0) = 0, \quad t \in [0, T] \\ \hat{V}(T, \hat{S}) = (\hat{S} - Ke^{\gamma T})^+, \quad \hat{S} \geq 0, \\ \hat{V}(t, \hat{S}) = \hat{S} - Ke^{\gamma t}, \quad \hat{S} > \hat{S}_f(t), \quad t \in [0, T], \\ \hat{V}(t, \hat{S}_f(t)) = \hat{S}_f(t) - Ke^{\gamma t}, \\ \frac{\partial \hat{V}}{\partial \hat{S}}(t, \hat{S}_f(t)) = 1. \end{array} \right. \quad (15)$$

Por (14) el valor de  $C$  se calcula por la ecuación (8).

## 2.4. Repos con dividendos retornados al cliente durante el tiempo del contrato

Si se considera el caso en el que los dividendos son entregados al cliente inmediatamente después del tiempo de redención del contrato, lo que quiere decir que, el banco recibe los dividendos y los reinvierte a la tasa libre de riesgo hasta que el cliente decida recuperar sus acciones. Entonces, el beneficio del cliente por recuperar sus acciones en el tiempo  $t$  esta dado por:

$$(S_t - Ke^{\gamma t})^+ + I_t, \quad (16)$$

donde

$$I_t = \int_0^t \delta e^{r(t-u)} S_u du. \quad (17)$$

Dado que el dividendo se paga continuamente, la anterior integral representa el monto que recibe el cliente en  $t$ , por dividendos. Luego el precio de la opción con valor intrínseco dada por la ecuación (16) es:

$$\begin{aligned} V(t, S_t, I_t) &= \sup_{\tau \in \mathcal{T}[t, T]} \mathbb{E} \left[ e^{-r(\tau-t)} \left( (S_\tau - Ke^{\gamma \tau})^+ + I_\tau \right) \middle| \mathcal{F}_t \right] \\ &= I_t + \sup_{\tau \in \mathcal{T}[t, T]} \mathbb{E} \left[ e^{-r(\tau-t)} \left( (S_\tau - Ke^{\gamma \tau})^+ + \int_t^\tau \delta e^{r(t-u)} S_u du \right) \middle| \mathcal{F}_t \right]. \end{aligned} \quad (18)$$

Nótese que  $V(t, S_t, I_t) - I_t$  es independiente de  $I_t$ , entonces se puede utilizar la transformación:

$$U(t, S_t) = V(t, S_t, I_t) - I_t. \quad (19)$$

Luego para resolver el problema (18) utilizando la transformación (19) encontramos el siguiente sistema de ecuaciones diferenciales parciales, vease Lu & Putri [LP15] (ecuación 3.23)

$$\left\{ \begin{array}{l} \frac{\partial U}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 U}{\partial S^2} + (r - \delta)S \frac{\partial U}{\partial S} - rU + \delta S = 0, \quad S < S_f(t), \quad t \in [0, T], \\ U(t, 0) = 0, \quad t \in [0, T], \\ U(T, S) = (S - Ke^{\gamma T})^+, \quad S \geq 0, \\ U(t, S) = S - Ke^{\gamma T}, \quad S > S_f(t), \quad t \in [0, T], \\ U(t, S_f(t)) = S_f(t) - Ke^{\gamma T}, \\ \frac{\partial U}{\partial S}(t, S_f(t)) = 1. \end{array} \right. \quad (20)$$

Nótese que encontrar el valor de la opción, es decir  $V(0, S_0, 0)$ , es equivalente a calcular  $U(0, S_0)$ . Por lo tanto, el valor de  $C$  se calcula por la ecuación (8).

### 3. Metodología

En esta sección vamos a desarrollar los métodos numéricos que nos aproximarán las soluciones de los sistemas de ecuaciones diferenciales parciales de la sección anterior.

#### 3.1. El método de Crank-Nicolson

Vamos a aproximar la solución de los sistemas de ecuaciones diferenciales parciales encontrados en la anterior sección por el método de Crank-Nicolson. Para esto, se define una malla uniforme  $\{0, \Delta t, 2\Delta t, \dots, N\Delta t\} \times \{0, \Delta S, 2\Delta S, \dots, I\Delta S\}$  sobre el rectángulo  $[0, T] \times [0, S_{\text{máx}}]$ , donde  $\Delta t = T/N$ ,  $\Delta S = S_{\text{máx}}/I$  y  $S_{\text{máx}}$  es un valor suficientemente grande para el precio de la acción  $S$ .

Ahora, tomando la aproximación para las derivadas que aparecen en la primera ecuación de (7)

$$\frac{\partial V}{\partial t} \approx \frac{V_i^n - V_i^{n+1}}{\Delta t}, \quad \frac{\partial V}{\partial S} \approx \frac{V_{i+1}^n - V_{i-1}^n}{2\Delta S}, \quad \frac{\partial V}{\partial S^2} \approx \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{(\Delta S)^2}, \quad (21)$$

donde  $V_i^n = V(n\Delta t, i\Delta S)$ , obtenemos la aproximación de esta ecuación diferencial parcial

$$\begin{aligned} \frac{V_i^{n+1} - V_i^n}{\Delta t} + \frac{\sigma^2 i^2 (\Delta S)^2}{4} \left( \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{(\Delta S)^2} \right) + \frac{\sigma^2 i^2 (\Delta S)^2}{4} \left( \frac{V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}}{(\Delta S)^2} \right) \\ + \frac{ri\Delta S}{2} \left( \frac{V_{i+1}^n - V_{i-1}^n}{2\Delta S} \right) + \frac{ri\Delta S}{2} \left( \frac{V_{i+1}^{n+1} - V_{i-1}^{n+1}}{2\Delta S} \right) - \frac{r}{2} V_i^n - \frac{r}{2} V_i^{n+1} = 0. \end{aligned} \quad (22)$$

Incluyendo las condiciones de frontera

$$\begin{cases} V_0^n = 0, & n = 0, 1, \dots, N, \\ V_i^N = (i\Delta S - Ke^{\gamma T})^+, & i = 0, 1, \dots, I, \\ V_I^n = S_{\text{máx}} - Ke^{\gamma n \Delta t}, & n = 0, 1, \dots, N. \end{cases} \quad (23)$$

Reorganizando (22) obtenemos que

$$-A_i^n V_{i-1}^n + (1 - B_i^n) V_i^n - C_i^n V_{i+1}^n = A_i^{n+1} V_{i-1}^{n+1} + (1 + B_i^{n+1}) V_i^{n+1} + C_i^{n+1} V_{i+1}^{n+1} \quad (24)$$

donde

$$\begin{aligned} A_i^n &= \frac{v_1 \sigma^2 i^2 (\Delta S)^2}{4} - \frac{v_2 r i \Delta S}{4}, \\ B_i^n &= -\frac{v_1 \sigma^2 i^2 (\Delta S)^2}{2} - \frac{r \Delta t}{2}, \\ C_i^n &= \frac{v_1 \sigma^2 i^2 (\Delta S)^2}{4} + \frac{v_2 r i \Delta S}{4}, \end{aligned} \quad (25)$$

con

$$v_1 = \frac{\Delta t}{\Delta S^2} \quad v_2 = \frac{\Delta t}{\Delta S}.$$

Las ecuaciones en (24) pueden escribirse en forma matricial como

$$M \mathbf{V}^n = \mathbf{q}, \quad (26)$$

donde

$$M \mathbf{V}^n = \begin{bmatrix} 1 - B_1^n & -C_1^n & 0 & \dots & 0 & 0 & 0 \\ -A_2^n & 1 - B_2^n & -C_2^n & \dots & 0 & 0 & 0 \\ 0 & -A_3^n & 1 - B_3^n & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 - B_{I-3}^n & -C_{I-3}^n & 0 \\ 0 & 0 & 0 & \dots & -A_{I-2}^n & 1 - B_{I-2}^n & -C_{I-2}^n \\ 0 & 0 & 0 & \cdot & 0 & -A_{I-1}^n & 1 - B_{I-1}^n \end{bmatrix} \begin{bmatrix} V_1^n \\ V_2^n \\ V_3^n \\ \vdots \\ V_{I-3}^n \\ V_{I-2}^n \\ V_{I-1}^n \end{bmatrix} \quad (27)$$

y

$$\mathbf{q} = \begin{bmatrix} A_1^{n+1} & 1 + B_1^{n+1} & C_1^{n+1} & \dots & 0 & 0 & 0 \\ 0 & A_2^{n+1} & 1 + B_2^{n+1} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 + B_{I-2}^{n+1} & C_{I-2}^{n+1} & 0 \\ 0 & 0 & 0 & \cdot & A_{I-1}^{n+1} & 1 + B_{I-1}^{n+1} & C_{I-1}^{n+1} \end{bmatrix} \begin{bmatrix} V_0^{n+1} \\ V_1^{n+1} \\ \vdots \\ V_{I-1}^{n+1} \\ V_I^{n+1} \end{bmatrix} - \begin{bmatrix} -A_1^{n+1} V_0^{n+1} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Para solucionar el sistema lineal de ecuaciones (26) vamos a utilizar dos métodos iterativos. La ventaja de los métodos iterativos sobre los métodos directos es que estos se adaptan de una forma natural a los problemas de valoración de opciones Americanas, como es nuestro caso.

### 3.2. Método SOR

El algoritmo Sucesivo de Sobre Relajación (SOR)<sup>1</sup> es un ejemplo de un método iterativo para la solución del sistema (26), el cual a su vez, es un refinamiento de otro método iterativo conocido como el método de Gauss-Seidel, véase [WHD95].

Suponga que la matriz  $M$  en el sistema (26) tiene entradas  $M_{ij}^n$ , entonces la ecuación  $i$ -ésima del sistema (26), se puede escribir como

$$M_{i1}^n v_1^n + M_{i2}^n v_2^n + \cdots + M_{ii}^n v_i^n + \cdots + M_{iI-1}^n v_{I-1}^n = q_i.$$

De donde

$$v_i^n = \frac{1}{M_{ii}^n} \left( q_i - \left( M_{i1}^n v_1^n + M_{i2}^n v_2^n + \cdots + M_{iI-1}^n v_{I-1}^n \right) \right).$$

Para poder resolver este problema iterativamente, notamos por  $v_i^{n,k}$  la  $k$ -ésima iteración y tomando un valor inicial  $v_i^{n,0}$  obtenemos una aproximación para la solución  $v_i^n$  de la siguiente forma

$$v_i^{n,k+1} = \frac{1}{M_{ii}^n} \left( q_i - \left( M_{i1}^n v_1^{n,k} + M_{i2}^n v_2^{n,k} + \cdots + M_{iI-1}^n v_{I-1}^{n,k} \right) \right).$$

Como para poder encontrar  $v_i^{n,k+1}$  ya se ha encontrado  $v_{i-1}^{n,k+1}$  se pueden tomar estos valores actualizados y reemplazarlos en la anterior ecuación obteniendo<sup>2</sup>:

$$v_i^{n,k+1} = \frac{1}{M_{ii}^n} \left( q_i - \sum_{j=1}^{i-1} M_{ij}^n v_j^{n,k+1} - \sum_{j=i+1}^{I-1} M_{ij}^n v_j^{n,k} \right). \quad (28)$$

Este proceso se repite hasta que la medida del error  $\|\mathbf{V}^{n,k+1} - \mathbf{V}^{n,k}\|$  sea lo suficientemente pequeña, entonces se toma a  $v_i^{n,k+1}$  como la mejor aproximación del valor de  $v_i^n$ .

Ahora, observemos que

$$v_i^{n,k+1} = v_i^{n,k} + \left( v_i^{n,k+1} - v_i^{n,k} \right),$$

como la secuencia de iteraciones  $v_i^{n,k}$  converge a  $v_i^n$  (ver [B06]), se puede pensar  $v_i^{n,k+1} - v_i^{n,k}$  como un término de correlación que se adiciona a  $v_i^{n,k}$  para acercarse al valor exacto de  $v_i^n$ . Entonces surge la posibilidad que la secuencia converja más rápidamente, se incluye un nuevo parámetro de aceleración  $\omega$ , de modo que:

$$v_i^{n,k+1} = v_i^{n,k} + \frac{\omega}{M_{ii}^n} \left( q_i - \sum_{j=1}^{i-1} M_{ij}^n v_j^{n,k+1} - \sum_{j=i+1}^{I-1} M_{ij}^n v_j^{n,k} \right). \quad (29)$$

Cuando  $0 < \omega < 1$ , el algoritmo se conoce como sub relajación, en lugar de sobre relajación que se utiliza para los casos en que  $1 < \omega < 2$ . De esta manera la ecuación (29) es la que se conoce como el algoritmo SOR.

<sup>1</sup>SOR: Successive Over-Relaxation.

<sup>2</sup>Ecuación (28) Método Gauss-Seidel.

La efectividad de este método esta dado por el radio de convergencia, ya que el número de iteraciones necesarias para reducir un error inicial es inversamente proporcional al radio de convergencia, a menor radio espectral mayor tasa de convergencia, como en este caso se depende del parámetro  $\omega$ , este se escogerá de tal forma que el radio espectral se minimice.

Esta demostrado que existe un valor óptimo de  $\omega$ , en el intervalo  $1 < \omega < 2$ , lo que conduce a una convergencia mucho mas rápida que para otros valores de  $\omega$ . Este valor óptimo de  $\omega$  depende de la dimension de la matriz  $M$ . Hay medios para calcular o estimar el valor óptimo de  $\omega$  pero estos implican tantos cálculos que es mas rápido cambiar  $\omega$  en cada paso de tiempo hasta que se encuentra un valor que minimiza el número de iteraciones.

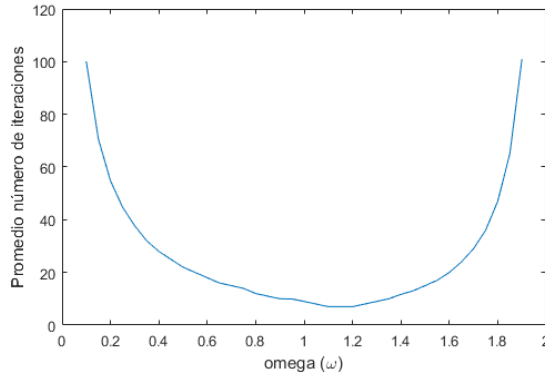


Figura 1: Grafica de  $\omega$  respecto al numero de iteraciones algoritmo SOR con  $S_0 = 50$ ;  $K = 40$ ;  $\sigma = 0, 5$ ;  $\gamma = 0, 1$ ;  $r = 0, 05$ ;  $T = 1$

### 3.3. Método SOR Proyectada

Haciendo algunas modificaciones al algoritmo SOR, obtenemos el algoritmo SOR proyectada, para esto se define una función para nuestro contrato americano con precio de ejercicio variable como,

$$f(t, S) = (S - Ke^{\gamma t})^+ \quad (30)$$

para todo  $S$  en un tiempo  $t$ , se observa que

$$\mathbf{V}^n \geq \mathbf{f}^n \iff \mathbf{M}\mathbf{V}^n \geq \mathbf{q} \quad (31)$$

para  $\mathbf{f}^n = (f_0^n, f_1^n, \dots, f_I^n)'$  con  $f_i^n = f(n\Delta t, i\Delta S)$ , donde se incorporan las condiciones de frontera libre dadas en las dos ultimas ecuaciones del sistema (7), así el algoritmo SOR proyectada halla  $v_i^n$  como el máximo entre SOR y el valor de  $f_i^n$ .

Se comienza con la estimación inicial

$$\begin{aligned} \mathbf{V}^{n,0} &= (\mathbf{V}^{n-1}, \mathbf{f}^n)^+ \\ v_i^{n,0} &= (v_i^{n-1}, f_i^n)^+ \end{aligned} \quad (32)$$

Entonces se genera  $v_i^{n,k+1}$  usando

$$v_i^{n,k+1} = \text{máx} \left\{ v_i^{n,k} + \frac{\omega}{M_{ii}^n} \left( q_i - \sum_{j=1}^{i-1} M_{ij}^n v_j^{n,k+1} - \sum_{j=i+1}^{I-1} M_{ij}^n v_j^{n,k} \right), f_i^n \right\}. \quad (33)$$

Este algoritmo, es construido a partir de las posibles inconsistencias que puede generar el algoritmo *SOR* al no tener en cuenta la condiciones  $S < S_f(t)$ ,  $t \in [0, T]$  y  $S > S_f(t)$ ,  $t \in [0, T]$  de las ecuaciones (1) y (4) del sistema (7). Por lo tanto, con el algoritmo de *SOR proyectada* garantizamos la consistencia interna del sistema (7), lo que nos garantiza encontrar una frontera de ejercicio más precisa y en consecuencia unos valores más exactos del contrato.

### 3.4. Frontera eficiente

La región de ejercicio (al tiempo  $t$ ) es el conjunto de valores de  $S$  tales que satisfacen:

$$V(t, S) = S - Ke^{\gamma t} \quad (34)$$

Se busca aproximar la frontera eficiente  $S_f(t)$  teniendo en cuenta la característica dada por la igualdad:

$$\frac{\partial V}{\partial S}(t, S_f(t)) = 1. \quad (35)$$

Existe un valor especial para  $S$  tal que:

$$\begin{aligned} \text{Si } S \geq S_f(t) &\Rightarrow V(t, S) = S - Ke^{\gamma t} \text{ y } \Delta(t, S) = 1 \\ \text{Si } S < S_f(t) &\Rightarrow V(t, S) = S - Ke^{\gamma t} \text{ y } \Delta(t, S) < 1, \end{aligned}$$

donde la letra griega “Delta” es definida por  $\Delta := \frac{\partial V}{\partial S}$ . Para encontrar la frontera eficiente para cada uno de los valores de  $t$  en la malla:

1. Se aproxima la derivada de  $V$  con respecto a  $S$  ( $\Delta$ ) usando la formula

$$\Delta_i = \frac{V_i - V_{i-1}}{\Delta S}, \text{ donde } \Delta S = S_i - S_{i-1} \text{ con } i = 0, \dots, I \quad (36)$$

La cual es una aproximación por diferencias hacia atrás de orden 1.

2. Se busca el primer indice  $j$  tal que  $\Delta_j = 1 \pm \varepsilon$ , donde  $\varepsilon$  es una tolerancia dada. Entonces  $S_f(t)$  se encuentra entre  $S_{j-1} = (j-1)\Delta S$  y  $S_j = j\Delta S$ .



3. Se calcula  $\Delta_{j-2}$  y  $\Delta_{j-1}$  usando las formulas:

$$\Delta_{j-2} = \frac{V_{j-1} - V_{j-3}}{\Delta S} \quad \text{y} \quad \Delta_{j-1} = \frac{3V_{j-1} - 4V_{j-2} + V_{j-3}}{2\Delta S} \quad (37)$$

Las Cuales son aproximaciones por diferencias hacia atras de orden 2, ver página 162 de [N92].

4. Se encuentran los parámetros de la recta,  $m$  y  $b$ , que pasa por  $(S_{j-2}, \Delta_{j-2})$  y  $(S_{j-1}, \Delta_{j-1})$  los cuales están dados por:

$$m = \frac{\Delta_{j-1} - \Delta_{j-2}}{S_{j-1} - S_{j-2}} \quad \text{y} \quad b = \Delta_{j-1} - mS_{j-1} \quad (38)$$

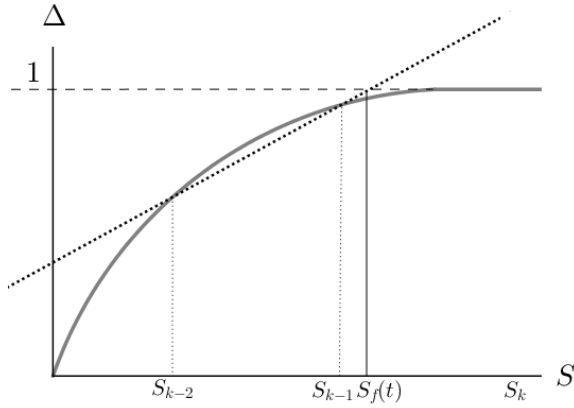


Figura 2: Grafico aproximación de  $S_f(t)$  usando diferencias finitas

5. La aproximación para  $S_f(t)$  es aquel valor para el cual la segunda componente del punto, sobre la recta encontrada en el literal anterior, es 1. Es decir:

$$S_f(t) \approx \frac{1 - b}{m}. \quad (39)$$

## 4. Implementación del método numérico y análisis del contrato

En esta sección comparamos los resultados de la implementación del método de Crank-Nicolson y el algoritmo *SOR*, presentados en la sección anterior, con los resultados que se conocen en la literatura que tiene valores comparativos, en este caso con Prager [P10]. A continuación se comparan los dos algoritmos iterativos *SOR* y *SOR proyectada*, además, se implementa la frontera de ejercicio y se hace un análisis de sensibilidad del valor del contrato con respecto a algunos parámetros.

## 4.1. Comparación con Prager

A continuación en la Tabla 1 se muestran los resultados de la implementación del método numérico y una comparación con Prager [P10] bajo sus parámetros de prueba con el algoritmo *SOR* donde el parámetro  $\omega$  es fijado en  $\omega = 1,1$ , (ver Anexo A y C).

$K$	$\sigma$	$\gamma$	Prager	<i>SOR</i>
10	0,5	0,1	39,49	40,4048
20			29,24	29,3952
30			20,34	20,3695
40			13,59	13,5934
50			8,89	8,8930
40	0,3	0,1	10,34	10,3425
	0,4		11,93	11,9335
	0,5		13,59	13,5934
	0,6		15,26	15,2811
	0,7		16,94	16,9734
	0,8		18,59	18,6501
40	0,5	0,07	14,27	14,2732
		0,1	13,59	13,5934
		0,13	12,91	12,9229
		0,16	12,25	12,2641
		0,19	11,6	11,6179
		0,22	10,96	10,9854

Tabla 1: Comparación con Prager utilizando los parámetros:  $S_0 = 50$ ,  $r = 5\%$  y  $T = 1$ .

En la anterior tabla, podemos observar que el método numérico es comparable con los resultados obtenidos por Prager, aunque difiere en algunos casos, siendo el caso más extremo los valores con  $K = 10$ ,  $\sigma = 0,5$  y  $\gamma = 0,1$ , es decir, Prager: 39,49 y método *SOR*: 40,4048.

Parametros			<i>SOR</i>		<i>SOR proyectada</i>	
<i>K</i>	$\sigma$	$\gamma$	Valor	Tiempo(s)	Valor	Tiempo(s)
10	0,5	0,1	40,40484993	1,1	40,40484993	72,6
20			29,39521099	1,4	29,39521099	186,8
30			20,36953893	1,6	20,36953893	322,1
40			13,59343481	2,2	13,59343481	583,7
50			8,89300244	2,9	8,89300244	1172,2
40	0,3	0,1	10,34259433	2,1	10,34259433	532,8
	0,4		11,93357325	2,3	11,93357325	658,1
	0,5		13,59343481	2,3	13,59343481	743,1
	0,6		15,28113609	2,7	15,28113609	952,4
	0,7		16,97340765	2,9	15,93340765	1067,3
	0,8		18,65010596	3,3	18,65010596	1154,6
40	0,5	0,07	14,27326012	2,4	14,27326012	562,9
		0,1	13,59343481	2,3	13,59343481	591,8
		0,13	12,92299333	2,6	12,92299333	592,2
		0,16	12,26411801	2,6	12,26411801	673,9
		0,19	11,61793665	2,5	11,61793665	674,8
		0,22	10,98543265	2,4	10,98543265	628,4

Tabla 2: Comparación *SOR* y *SOR proyectada* bajo los parámetros:  $\omega = 1,1$ ,  $S_0 = 50$ ,  $r = 5\%$  y  $T = 1$ .

En la Tabla 2 se relaciona el tiempo y el valor entre los dos algoritmos implementados *SOR* y *SOR proyectada* (ver Anexo A, B y C) para los parámetros dados en Prager [P10]. Como se observa en la tabla el algoritmo *SOR proyectada* no genera ningún beneficio computacional al valor del contrato, y toma un tiempo considerablemente mayor para encontrar este, a pesar de ser el método escogido por varios autores para la solución de opciones americanas, véase [B06] y [W06]. Como el algoritmo *SOR* no afecta la consistencia en la valoración del contrato, se trabajara con éste algoritmo para los cálculos siguientes.

## 4.2. Análisis de los contratos Repo

A continuación se muestra un ejemplo ilustrativo del cálculo de la prima necesaria para adquirir un préstamo (ver Anexo: C, D, E y F), este ejercicio se hace para los cuatro tipos de contratos teniendo en cuenta que el valor del préstamo es un porcentaje del colateral en el tiempo inicial ( $t = 0$ ) que llamaremos  $\alpha$ , es decir,  $K = \alpha S_0$ .

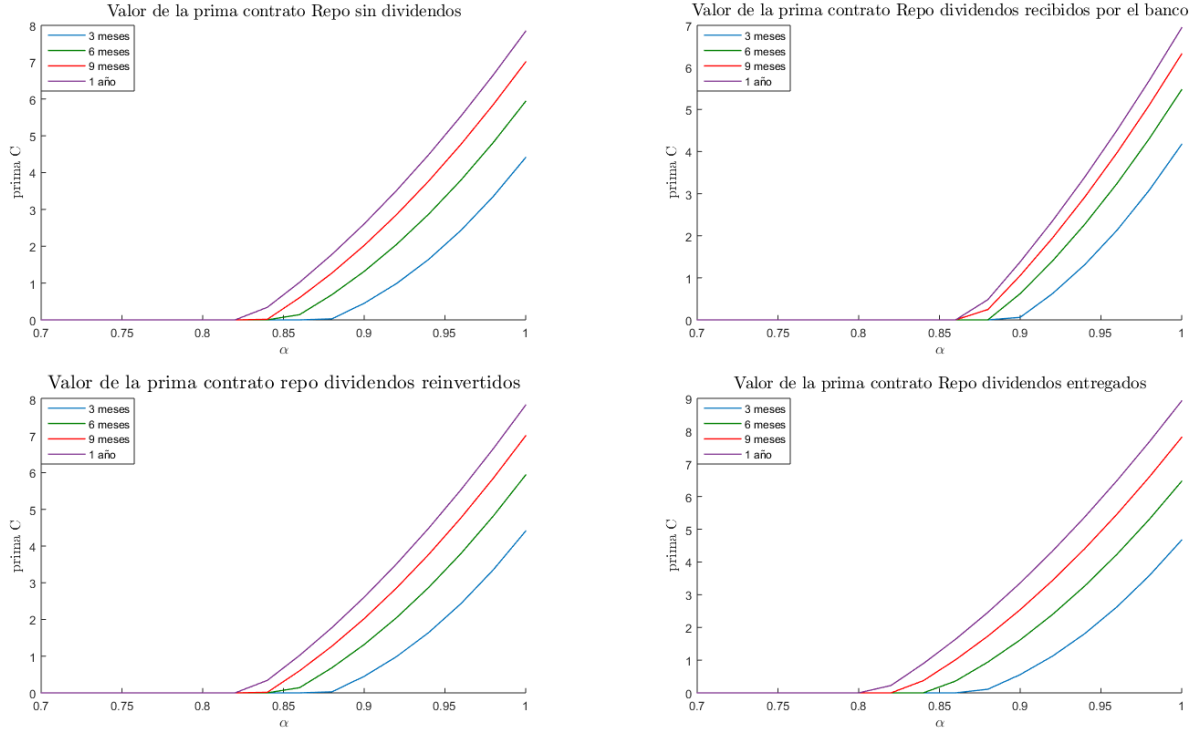


Figura 3: Valor de la prima  $C$  variando  $\alpha$  con  $S_0 = 100$ ,  $\sigma = 0,25$ ,  $\gamma = 0,12$ , y  $\delta = 0,02$ .

En la Figura 3 se puede observar que hay un conjunto de valores  $\alpha$  para los cuales la prima  $C$  es cero en los cuatro contratos, para los valores positivos de  $C$  se puede notar, que a mayor tiempo hay un aumento en el valor de la prima, pues si el tiempo aumenta también aumenta la probabilidad de recuperación de las acciones por parte del cliente (ver figura 5), de esta manera se puede encontrar el mayor valor de  $\alpha$ , para los cuales  $C$  es cero:

Tiempo (meses)	$\alpha$			
	Sin dividendos	Recibidos por el banco	Reinvertidos	Entregados
3	86 %	88 %	86 %	86 %
6	84 %	86 %	84 %	84 %
9	82 %	86 %	82 %	82 %
12	82 %	86 %	82 %	80 %

Tabla 3: Mayores valores de  $\alpha$  para los cuales la prima  $C$  es cero.

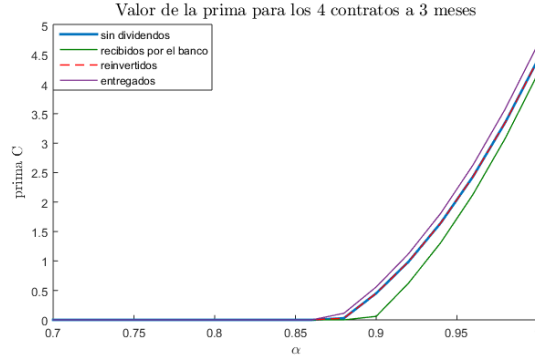


Figura 4: Valor de la prima  $C$  variando  $\alpha$  con  $T = 3$  meses para los cuatro contratos.

Si la duración del contrato Repo es por 3 meses, se puede realizar un préstamo entre el 70 % y el 86 % sobre el valor de  $S_0$  y el valor de la prima para los cuatro contratos es 0, se observa que el contrato con dividendos entregados es el más costoso, pues si el cliente paga el préstamo, el banco le regresara las acciones, los dividendos y sus intereses generados, después encontramos los contratos sin dividendos y con dividendos reinvertidos, que son iguales (ver Ecuación (12)), pues si el cliente paga el préstamo al cliente se le entregarán las acciones mas los dividendos si es el caso, y por último encontramos el contrato con dividendos recibidos por el banco, este tiene menor costo porque el banco se queda con los dividendos como parte de pago del préstamo.

### 4.3. Frontera de ejercicio

Una vez aplicado el algoritmo con el método *SOR* y utilizando el algoritmo de frontera de ejercicio (ver Anexo G), se encuentra el momento en el cual el cliente debería ir al banco a pagar el préstamo y recuperar sus acciones (Frontera de ejercicio), ya que obtendrá un beneficio, la prima  $C$  le da al cliente el derecho de recuperarlas en cualquier momento. El conocimiento de esta frontera beneficia más al cliente, pues el banco ya ha cobrado una prima que lo cubre de este riesgo.

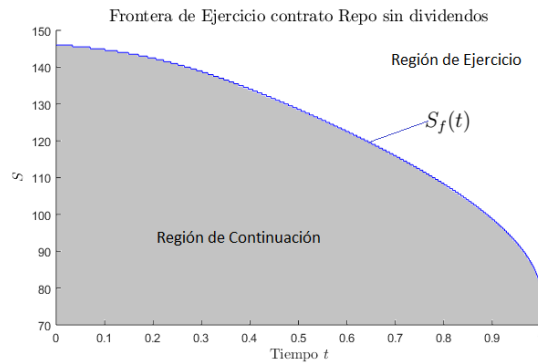
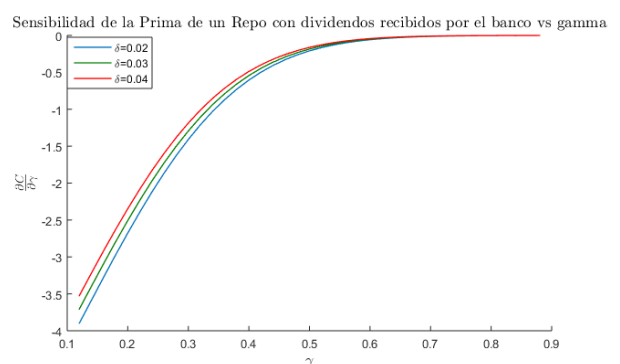
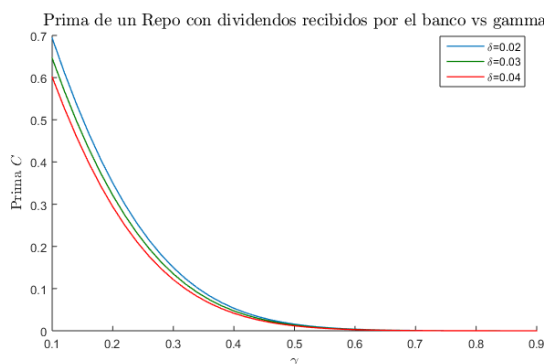
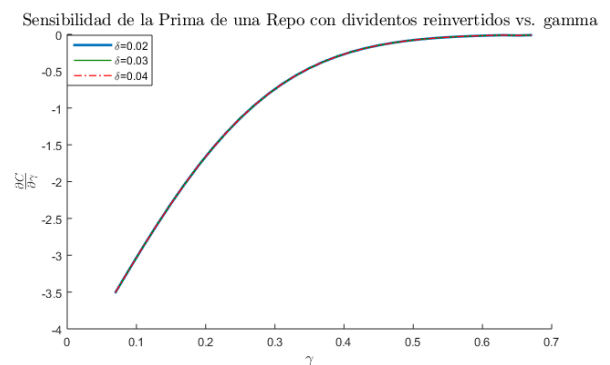
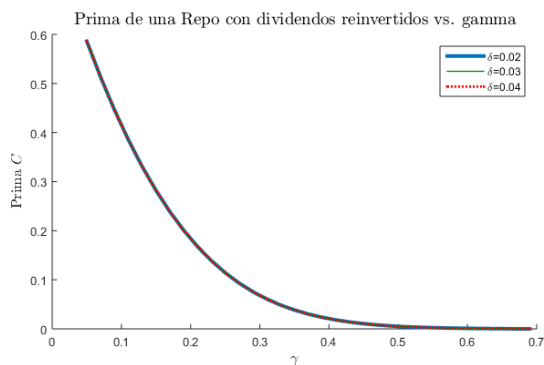
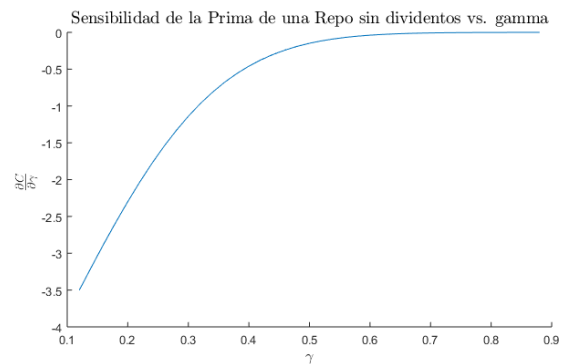
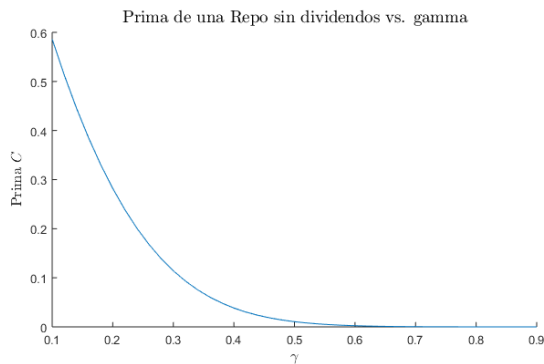


Figura 5: Frontera de ejercicio.

En la figura anterior podemos observar que al encontrar la frontera de ejercicio, dividimos el espacio precio  $S$  vs. tiempo  $t$  en dos regiones, la región de continuación, en la cual el cliente no tiene incentivos para recuperar sus acciones y, la región de ejercicio, en esta región el valor  $S > Ke^{\gamma t}$ , esto quiere decir que en esta región el cliente podría ir a pagar el préstamo más los intereses y así recuperar sus acciones, dándole un beneficio positivo.

#### 4.4. Análisis de sensibilidad

A continuación se hace el análisis de la sensibilidad del valor de la prima  $C$  con respecto a los parámetros  $\gamma$  y  $\delta$  ya que variaciones en estos afectan de manera significativa a la valoración del contrato. Para estos análisis se usaron los siguientes parámetros:  $S_0 = K = 10$ ,  $\sigma = 0,2$   $T = 1$ .



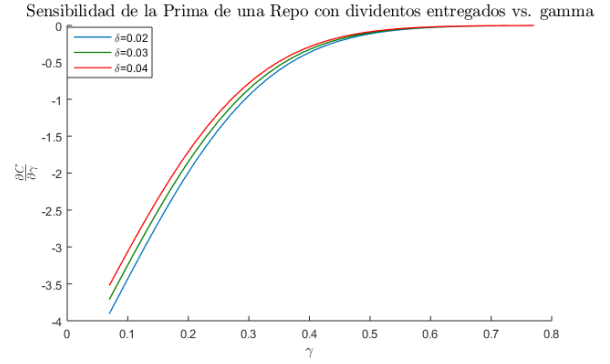
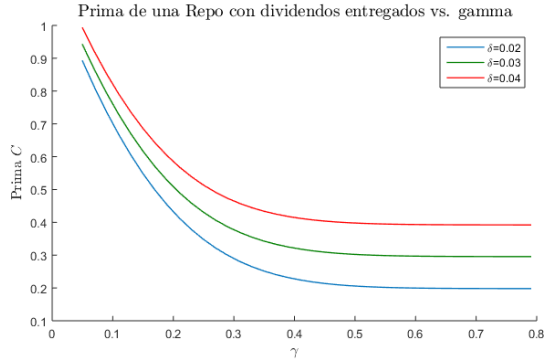
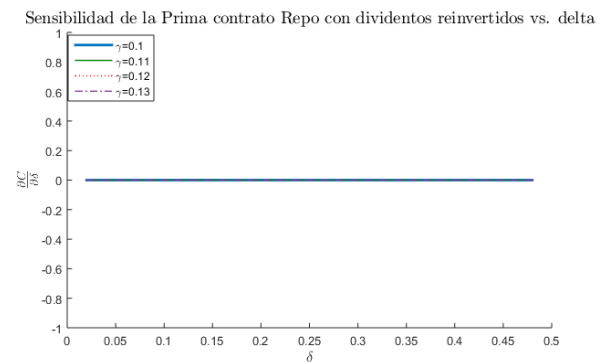
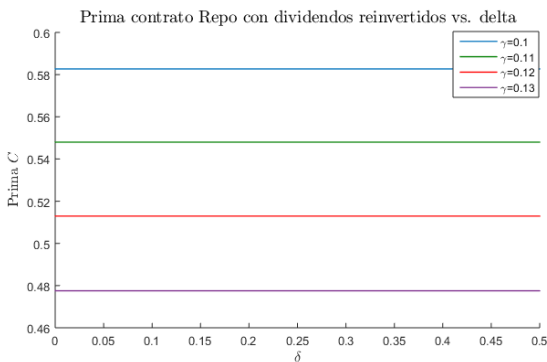


Figura 6: Graficas análisis de sensibilidad con respecto a  $\gamma$ .

En la Figura 6 se puede ver que el comportamiento del contrato cuando hay cambios en la tasa de interés cobrada por el banco ( $\gamma$ ), es similar en los cuatro contratos, si se aumenta la tasa que cobra el banco por el contrato el valor de la prima cobrada al cliente disminuye. También se puede observar que en la gráfica de la sensibilidad con dividendos reinvertidos no cambia cuando se varía la tasa de dividendos  $\delta$ ; además, esta es igual a la gráfica de la sensibilidad sin dividendos, esto ocurre por la transformación hecha en (12) para poder calcular el valor del contrato.

De igual manera, en la siguiente figura se analiza la sensibilidad del valor de la prima con respecto al parámetro  $\delta$  (la tasa de dividendos de la acción), y se muestra el valor de la prima  $C$  variando valores de  $\gamma$ .



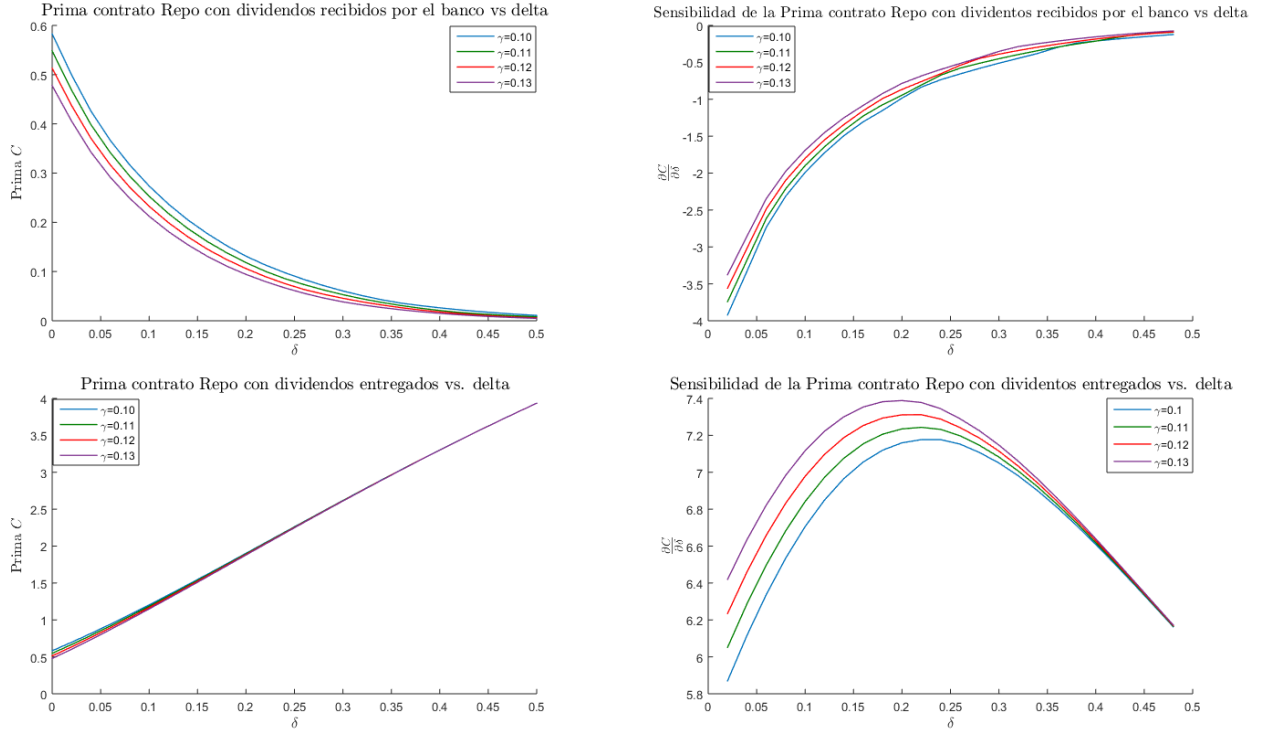


Figura 7: Graficas prima y sensibilidad con respecto a  $\delta$

En la Figura 7 se muestra el resultado para los 3 tipos de contratos que incorporan dividendos, para el contrato de dividendos reinvertidos, los dividendos no afectan el valor del contrato (ver sección 2.3), para cuando los dividendos son recibidos por el banco podemos observar que un aumento en el porcentaje de dividendos disminuirá el valor de la prima  $C$  cobrada al cliente, pues el banco se esta quedando con los dividendos como parte de pago del contrato. Cuando los dividendos son entregados al cliente podemos observar que el valor de la prima siempre aumenta aunque hay un cambio de tendencia en el precio de  $C$ , estos puntos cambian dependiendo del valor de gamma ( $\gamma$ ), como se muestra en la siguiente tabla,

$\gamma$	$\delta$	<i>Sensibilidad</i>
0,10	0,2	7,1769
0,11	0,2	7,2434
0,12	0,2	7,3118
0,13	0,18	7,3885

Tabla 4: Máximos gráfica de sensibilidad contrato Repo con dividendos entregados vs  $\delta$ .

Estos deltas son el punto exacto donde hay un cambio en la velocidad de crecimiento del valor de la prima  $C$ . Si el porcentaje de dividendos es mayor que en la tabla 4 disminuye el porcentaje posible de perdida para el banco, sin embargo, aun el cliente tiene que pagar por



el derecho de recuperar las acciones y los dividendos con los intereses acumulados hasta un tiempo  $t \leq T$ .

## 5. Aplicación al mercado colombiano

Para poder realizar operaciones Repo en Colombia (con pacto de recompra), las acciones deben cumplir los criterios de riesgo y liquidez establecidos por la Bolsa de Valores de Colombia (BVC). Los criterios de elegibilidad dependen de las garantías, el volumen promedio diario de negociación, el porcentaje de frecuencia de marcación de precio y la dispersión del volumen, todo esto en el trimestre inmediatamente anterior, las acciones que son elegibles son las que pertenezcan a los grupos  $A$ ,  $B$  y  $C$ <sup>3</sup>. Las acciones que se tomaron para la aplicación en el caso colombiano se escogieron con respecto a el informe de riesgos de renta variable de la BVC con fecha 31 de julio de 2017, de tal manera que todas son elegibles para realizar el contrato.

A continuación se presentan las 20 primeras acciones que componen el índice COLCAP, por orden alfabético, con su valor por acción, volatilidad y porcentaje de dividendos anuales.

<i>Ticker</i>	$S_0$ (1 agosto)	Volatilidad	Tasa continua de dividendos
<i>BCOLOMBIA</i>	30,860	17,67 %	3,03 %
<i>BOGOTA</i>	68,300	19,73 %	4,47 %
<i>CELSIA</i>	4,720	14,28 %	2,72 %
<i>CEMARGOS</i>	11,860	16,13 %	2,00 %
<i>CORFICOLCF</i>	29,120	24,53 %	1,35 %
<i>ECOPETROL</i>	1,365	23,65 %	1,67 %
<i>EEB</i>	1,985	14,86 %	4,87 %
<i>EXITO</i>	15,340	14,22 %	0,32 %
<i>GRUPOARGOS</i>	21,260	17,08 %	1,45 %
<i>GRUPOSURA</i>	41,740	14,58 %	1,16 %
<i>ISA</i>	13,600	16,18 %	2,84 %
<i>NUTRESA</i>	26,760	12,26 %	1,98 %
<i>PFAVAL</i>	1,320	15,87 %	4,36 %
<i>PFAVH</i>	2,680	36,24 %	2,83 %
<i>PFBCOLOM</i>	32,940	17,92 %	2,84 %
<i>PFCEMARGOS</i>	10,640	13,53 %	2,23 %
<i>PFDAVVNDA</i>	33,780	15,24 %	2,77 %
<i>PFGRUPOARG</i>	19,500	16,65 %	1,58 %
<i>PFGRUPSURA</i>	40,580	14,74 %	1,20 %

Tabla 5: Acciones que generan dividendos.

<sup>3</sup>Modelo de riesgo para Repos y TTVs sobre renta variable, fuente: [www.bvc.com.co](http://www.bvc.com.co)

En la siguiente tabla se muestra la acción que no entrega dividendos:

<i>Ticker</i>	$S_0$ (1 agosto)	Volatilidad
<i>CLH</i>	10,800	25, 21 %

Tabla 6: Acciones que no generan dividendos.

Para calcular la volatilidad anual de cada una de las acciones se tomaron los precios de cierre desde 1 de julio de 2016 al 31 de julio del 2017, se calcularon los retornos logarítmicos, se encontró la desviación estándar de estos datos y se multiplico por  $\sqrt{245}$ . Para la tasa de dividendos anuales se tomo la información de los pagos de dividendos para el año 2017, y se encontró su respectiva tasa continua.

En las siguientes Tablas se presenta el análisis del valor de la prima para diferentes acciones listadas en la BVC, primero se presenta la acción que no genera dividendos.

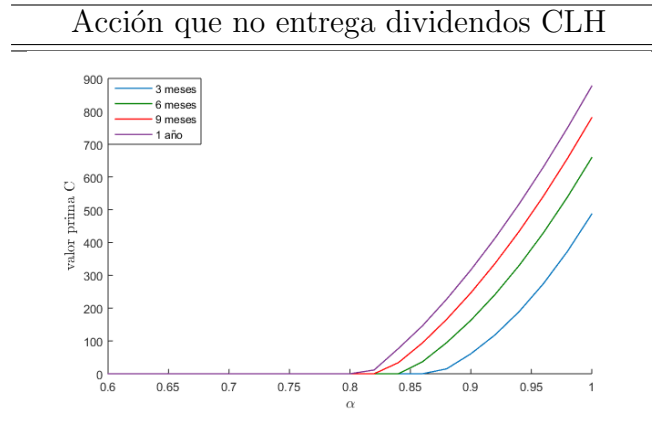


Tabla 7: Valor de la prima  $C$  contrato sin dividendos,  $\gamma = 12\%$ ,  $r = 7,5\%$

Las acciones que se presentan a continuación están organizadas según su porcentaje de participación en el índice COLCAP, observando el valor de la prima por tipo de contrato y tiempo.

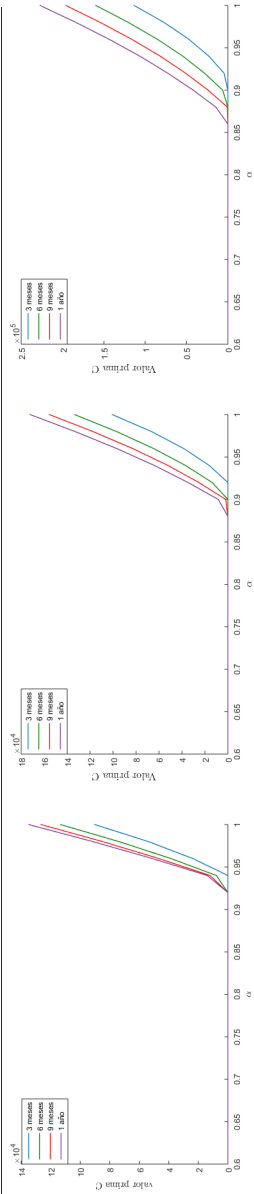
Retornos al cliente

Reinvertidos

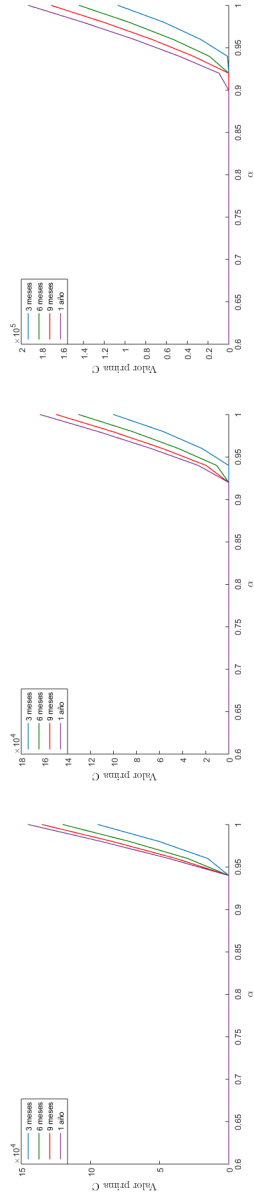
Recibidos por el banco

Ticker

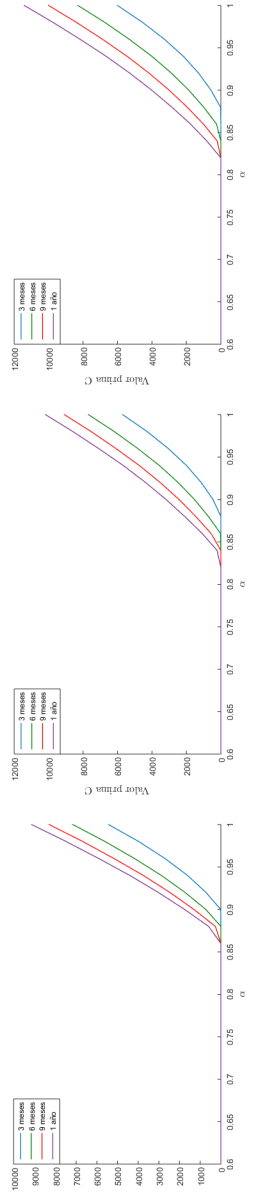
PFBCOLOM



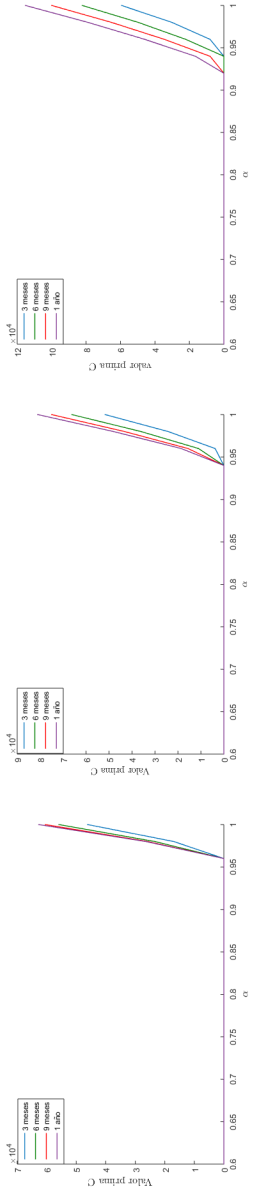
GRUPOSURA



ECOPETROL



NUTRESA



GRUPOARGOS

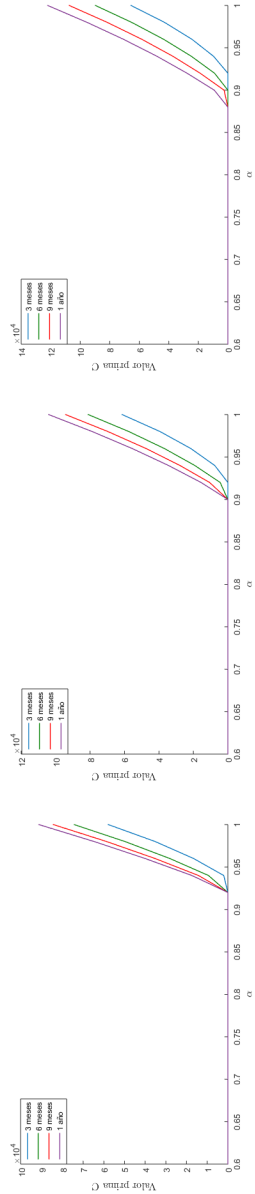


Tabla 8: Valor de la prima  $C$  por contratos,  $\gamma = 12\%$ ,  $r = 7,5\%$

En las tablas anteriores podemos observar que para todas las acciones y con cualquier política de dividendos la prima del contrato siempre es cero para un préstamo por el 70 % del valor del colateral. Esto nos indica que no existe el riesgo de anticipación de pago del préstamo en las condiciones actuales en el mercado colombiano. Por otro lado, en las tablas anteriores podemos observar que se podría aumentar el porcentaje del valor del colateral prestado al cliente y aun así no incurrir en el riesgo de anticipación del pago del préstamo.

También podemos decir que el contrato más conveniente para el cliente y el prestamista sería un contrato Repo con dividendos recibidos por el banco, este es el menos riesgoso, porque el cliente puede recibir un mayor porcentaje de préstamo sobre sus acciones sin cobro de una prima y el prestamista esta recibiendo los dividendos durante el tiempo del contrato.

## 6. Conclusiones

En este trabajo se analizó las operaciones Repo sin pacto de recompra en cuatro tipos de contratos con diferentes tratamientos a los dividendos, analizando el riesgo que asume el prestamista al permitir que el cliente pueda pagar el préstamo en cualquier momento, cuando este esté en la región de ejercicio. Al observar los resultados numéricos y gráficos se puede notar que este riesgo no es fundamental, ya que el valor de la prima  $C$  es cero para la mayor parte de los  $\alpha$ 's en el intervalo  $[0\%, 100\%]$ . A pesar de esto, creemos que la modelación matemática y la implementación numérica del contrato, como se realizó en este trabajo, nos podría permitir hacer un análisis a otro tipo de riesgos, como por ejemplo, el riesgo de no pago por medio del algoritmo de la frontera de ejercicio, los riesgos en el cambio del mercado, cambios en las condiciones del contrato, entre otros.

## Referencias

- [BS73] BLACK, F., Y SCHOLLES M. (1973). *The Pricing of Options and Corporate Liabilities*. Journal of Political Economy, 81(3), 637 - 654.
- [XZ07] XIA, J., Y ZHOU, Y. (2007). *Stock Loans*. Math. Finance , 17(2), 307-317.
- [DX11] DAI, M., Y XU, Z. Q. (2011). *Optimal Redeeming Strategy of Stock Loans with Finite Maturity*. Mathematical Finance, 21(4), 775 - 793.
- [LP15] LU, X. Y PUTRI, E. (2015). *Semi-analytic valuation of stock loans with finite maturity* . Commun Nonlinear Sci Numer Simulat, 27, 206 - 215.
- [GG12] GRASSELLI M. Y GÓMEZ C. (2012). *Stock Loans in Incomplete Market*. Applied Mathematical Finance, 20, 118 - 136.
- [P10] PRAGER D.J. (2010). *Valuation of finite maturity stock loans under regime switching and mean reverting stock models*. Thesis, University of Georgia, Athens.
- [W06] WILMOTT P. (2006), Paul Wilmott on quantitative finance. England: John Wiley & Sons, Ltd.
- [WHD95] WILMOTT P., HOWISON S. Y DEWYNNE J. (1995), The Mathematics of Financial Derivatives. Cambridge University Press.
- [B06] BRANDIMARTE P. (2006). Numerical Methods in Finance and Economics (second ed.). Jhon Wiley & Sons, Inc.
- [N92] NAKAMURA, S. (1992). *Metodos numericos aplicados con software* . Mexico : Prentice-hall hispanoamerica S.A. .

# Anexos

## A. Algoritmo SOR

```
function [ x, nIters, errs ] = SORIter( M, q, v0, omega, tol, maxIters )
% SORIter( M, q, v0, omega, tol,maxIters): ejecutar iteración SOR
%para resolver M*x = q
% omega es el parametro de sobrerelajación ( si omega > 1 ) o
% subrelajación si omega está en (0, 1)
% el algoritmo corre hasta que || v^(n+1) - v^n || < tol o hasta maxIters.
% Si maxIters no está definido, se usa maxIters = 1000

err = tol * 2;
nIters = 0;

if(nargin < 6)
    maxIters = 1000;
end

errs = zeros( maxIters, 1 );

[ m , n ] = size( M );
if( m ~= n )
    error( 'M debe ser cuadrada' )
end

vViejo = v0;
vNuevo = zeros( m, 1 ); % v^{n+1}

D = diag(M); % vector
U = triu( M, 1 ); % triangulo superior de M estrictamente por encima de M
DmasOmegaL = diag(D) + omega * tril(M, -1);
omega_q = omega * q;
T = (1 - omega) * diag(D) - omega * U;

while( err > tol && nIters < maxIters )
    % formula v^{n+1} = ( D + omega * L)^(-1) ( omega * q + ((1 - omega) * D ...
    % - omega * U ) * v^{n} )

    vNuevo = DmasOmegaL \ ( T * vViejo + omega_q );

    nIters = nIters + 1;
```

```

    err = norm( vViejo - vNuevo );
    errs(nIters) = err;
    vViejo(:) = vNuevo;
end

x = vNuevo;
errs = errs( 1 : nIters );
end

```

## B. Algoritmo SOR proyectada

```

function [x,nIters,errs] = SORproyectada(M,q,v0,omega,tol,maxIters,pago)
% SORproyectada( M, q, v0, pago, tol,maxIters): ejecutar iteracion
%SOR proyectada para resolver M*x = q
% pago es un vector columna con I-1 componentes
% el algoritmo corre hasta que || v^(n+1) - v^n || < tol o hasta maxIters.
% Si maxIters no est? definido, se usa maxIters = 1000

err = tol * 2;
nIters = 0;

if(nargin < 7)
    maxIters = 1000;
end

errs = zeros( maxIters, 1 );

[ m , n ] = size( M );
if( m ~= n )
    error( 'M debe ser cuadrada' )
end
vViejo = v0;
vNuevo = zeros( m, 1 ); % v^{n+1}
xsor=zeros(m,1);
D = diag(M);

while( err > tol && nIters < maxIters )
    for i=1:m
        vNuevo(i)=vViejo(i)+(omega/D(i))*(q(i)-M(i,1:i-1)*vNuevo(1:i-1) ...
        ...- M(i,i:m)*vViejo(i:m));
        vNuevo(i)=max(vNuevo(i),pago(i));
    end
end

```

```

    nIters = nIters + 1;
    err = norm( vViejo - vNuevo );
    errs(nIters) = err;
    vViejo(:) = vNuevo;
end

x = vNuevo;
errs = errs( 1 : nIters );
end

```

## C. Contrato Repo sin dividendos

```

function [ precio, VO, S, V, M, SORnIters] = BSCNrepo_nodiv( S0, K, r, ...
    gamma ,T, sigma, dS, dt, omega, SORP )

% Calcular el precio de un opción Repo sin dividendos usando el método
% de Crank-Nicolson
% S0 = Precio inicial del subyacente (puede ser un vector)
% K = Strike
% r = Tasa de interés del subyacente
% y = Tasa de interés del presta,o
% T = Fecha de vencimiento
% sigma = Volatilidad
% dS = Longitud de los subintervalos de la grilla para la variable espacial
% dt = Longitud de los subintervalos de la grilla para la variable temporal
% omega = parámetro para el uso del metodo SOR

tol = 10^-6; % para la iteración SOR

minS = 0;
maxS = 4 * K;

I = round( (maxS - minS)/dS );
S = linspace( minS, maxS, I + 1 )';
N = round( T / dt );
t = linspace( 0, T, N+1);

V = zeros( I+1, N+1);

% Función de pago para una repo en T

```



```

Pago = max( S - K*exp(gamma*T), 0 );
% También usada con condición final
V( :, N+1 ) = Pago;

% Condiciones de frontera para S -> 0 y S -> infinito
Fabajo = 0;
V(1, :) = Fabajo;
Farriba = maxS - K*exp(gamma*t);
V(I+1, :) = Farriba;

% para la ecuación tradicional de Black-Scholes,
% los coeficientes son independientes del tiempo:

a = 0.5 * sigma^2 * S.^2;
b = r * S;
c = -r * ones( I + 1, 1);

nu1 = dt / dS^2;
nu2 = dt / dS;

% construimos los coeficientes para la inducción hacia atrás
% como combinaciones de los anteriores

A = 0.5 * nu1 * a - 0.25 * nu2 * b;
B = - nu1 * a + 0.5 * dt * c;
C = 0.5 * nu1 * a + 0.25 * nu2 * b;

A = A( 2 : I, 1 );
B = B( 2 : I, 1 );
C = C( 2 : I, 1 );

% Los reunimos en una matriz de I-1 x I+1
ImasF = spdiags( [ A (1 + B) C ], [ 0 1 2 ], I-1, I+1 ); %modificar
ImenosF =spdiags( [ -A (1 - B) -C ], [ 0 1 2 ], I-1, I+1 ); %modificar
% Construir matriz cuadrada M = ( I - G ):
% remover primera columna y última fila de I-F
M = ImenosF( : , 2:I ); % resultado es (I-1) x (I-1)
% * Si las condiciones de frontera son de otra forma la construcción
% es más complicada!
% Vector r que usaremos abajo
rvec = zeros( I - 1, 1 );
rvec(1) = -Fabajo * A(1);
SORnIters = 0;
% INDUCCIÓN hacia atrás
for k = N+1:-1:2

```

```

rvec(I-1) = -Farriba(k)* C(I-1);
% Lado derecho del sistema es q = ImasF * V(:,k) - rvec
q1 = ImasF * V(:, k ) - rvec; %%%% RENOMBRO q
% Resolvemos la ecuación
% (I - G) * V(2:I, k-1) + rvec = ImasF * V(:, k),
% es decir,
% M * V( 2:I, k-1 ) = q
% mediante iteración SOR con 'adivinanza' inicial igual
% a V(2:I) de la anterior iteración
if SORP==1
[Vmonho,nIters] = SORproyectada(M,q1, V(2:I, k), omega, tol, 1000,Pago(2:I));
V(2:I, k-1) = Vmonho;
else
[Vmonho, nIters] = SORIter( M, q1, V(2:I, k), omega, tol, 1000 );
V(2:I, k-1) = max( Vmonho, Pago(2:I) );
end

SORnIters = SORnIters + nIters;

end

V0 = V( :,1);

precio = interp1(S , V0 , S0 , 'linear', 'extrap');

end

```

## D. Contrato Repo con dividendos recibido por el banco

```

function [ precio, V0, S, V, M, SORnIters] = BSCNrepo_divbank( S0, K, r, gamma, ...
...T, sigma, dS, dt, omega, delta , SORP)
% BSCNrepo( S0, K, r, y ,T, sigma, d, dS, dt, omega )
% Calcular el precio de un opción Repo sin dividendos usando el método
% de Crank-Nicolson
% S0 = Precio inicial del subyacente (puede ser un vector)
% K = Strike
% r = Tasa de interés del subyacente
% y = Tasa de interés del presta,o
% T = Fecha de vencimiento
% sigma = Volatilidad
% dS = Longitud de los subintervalos de la grilla para la variable espacial

```

```

% dt = Longitud de los subintervalos de la grilla para la variable temporal
% omega = parámetro para el uso del metodo SOR

tol = 10^-6; % para la iteración SOR

minS = 0;
maxS = 4 * K;

I = round( (maxS - minS)/dS );
S = linspace( minS, maxS, I + 1 )';
N = round( T / dt );
t = linspace( 0, T, N+1);

V = zeros( I+1, N+1);

% Función de pago para una repo en T
Pago = max( S - K*exp(gamma*T), 0 );
% También usada con condición final
V( :, N+1 ) = Pago;

% Condiciones de frontera para S -> 0 y S -> infinito
Fabajo = 0;
V(1, :) = Fabajo;
Farriba = maxS - K*exp(gamma*t);
V(I+1, :) = Farriba;

% para la ecuación tradicional de Black-Scholes,
% los coeficientes son independientes del tiempo:

a = 0.5 * sigma^2 * S.^2;
b = (r - delta) * S;
c = -r * ones( I + 1, 1);

nu1 = dt / dS^2;
nu2 = dt / dS;

% construimos los coeficientes para la inducción hacia atrás
% como combinaciones de los anteriores

A = 0.5 * nu1 * a - 0.25 * nu2 * b;
B = - nu1 * a + 0.5 * dt * c;
C = 0.5 * nu1 * a + 0.25 * nu2 * b;

A = A( 2 : I, 1 );

```

```

B = B( 2 : I, 1 );
C = C( 2 : I, 1 );

% Los reunimos en una matriz de I-1 x I+1
ImasF = spdiags( [ A (1 + B) C ], [ 0 1 2 ], I-1, I+1 ); %modificar
ImenosF =spdiags( [ -A (1 - B) -C ], [ 0 1 2 ], I-1, I+1 ); %modificar
% Construir matriz cuadrada M = ( I - G ):
% remover primera columna y última fila de I-F
M = ImenosF( : , 2:I ); % resultado es (I-1) x (I-1)
% * Si las condiciones de frontera son de otra forma la construcción
% es más complicada!
% Vector r que usaremos abajo
rvec = zeros( I - 1, 1 );
rvec(1) = -Fabajo * A(1);
SORnIters = 0;
% INDUCCIÓN hacia atrás
for k = N+1:-1:2
    rvec(I-1) = -Farriba(k)* C(I-1);
    % Lado derecho del sistema es q = ImasF * V(:,k) - rvec
    q1 = ImasF * V(:, k) - rvec; %%%% RENOMBRO q
    % Resolvemos la ecuación
    % (I - G) * V(2:I, k-1) + rvec = ImasF * V(:, k),
    % es decir,
    % M * V( 2:I, k-1 ) = q
    % mediante iteración SOR con 'adivinanza' inicial igual
    % a V(2:I) de la anterior iteración
    if SORP==1
        [Vmonho,nIters] = SORproyectada(M,q1, V(2:I, k), omega, tol, 1000,Pago(2:I));
        V(2:I, k-1) = Vmonho;
    else
        [Vmonho, nIters] = SORIter( M, q1, V(2:I, k), omega, tol, 1000 );
        V(2:I, k-1) = max( Vmonho, Pago(2:I) );
    end

    SORnIters = SORnIters + nIters;

end

V0 = V( :,1);

precio = interp1(S , V0 , S0 , 'linear', 'extrap');

end

```

## E. Contrato Repo con dividendos entregados al cliente

```
function [ precio, VO, S, V, M, SORnIters] = BSCNrepo_diven( S0, K, r, gamma,...
...T, sigma, dS, dt, omega, delta , SORP)
% BSCNrepo( S0, K, r, y ,T, sigma, d, dS, dt, omega )
% Calcular el precio de un opción Repo dividendos entregados al cliente en
% al tiempo de redencion
% S0 = Precio inicial del subyacente (puede ser un vector)
% K = Strike
% r = Tasa de interés del subyacente
% d = Tasa de dividendos del subyacente
% y = Tasa de interés del presta,o
% T = Fecha de vencimiento
% sigma = Volatilidad
% dS = Longitud de los subintervalos de la grilla para la variable espacial
% dt = Longitud de los subintervalos de la grilla para la variable temporal
% omega = parámetro para el uso del metodo SOR
% SORP = si es igual a 1 usa el algoritmo SORProyectada en caso contrario
% usa SORIter

tol = 10^-6; % para la iteración SOR

minS = 0;
maxS = 4 * K;

I = round( (maxS - minS)/dS );
S = linspace( minS, maxS, I + 1 )';
N = round( T / dt );
t = linspace( 0, T, N+1);

V = zeros( I+1, N+1);

% Función de pago para una repo en T
Pago = max( S - K*exp(gamma*T), 0 );
% También usada con condición final
V( :, N+1 ) = Pago;

% Condiciones de frontera para S -> 0 y S -> infinito
Fabajo = 0;
Farriba = maxS - K*exp(gamma*t);
V(1, :) = Fabajo;
V(I+1, :) = Farriba;
```

```

% para la ecuación tradicional de Black-Scholes,
% los coeficientes son independientes del tiempo:
%
a = 0.5 * sigma^2 * S.^2;
b = ( r - delta ) * S;
c = -r * ones( I + 1, 1);
d = delta * dt * S(2:end-1);

nu1 = dt / dS^2;
nu2 = dt / dS;

% construimos los coeficientes para la inducción hacia atrás
% como combinaciones de los anteriores

A = 0.5 * nu1 * a - 0.25 * nu2 * b;

B =      - nu1 * a + 0.5 * dt * c;

C = 0.5 * nu1 * a + 0.25 * nu2 * b;

A = A( 2 : I, 1 );
B = B( 2 : I, 1 );
C = C( 2 : I, 1 );

% Los reunimos en una matriz de I-1 x I+1
ImasF = spdiags( [ A (1 + B) C ], [ 0 1 2 ], I-1, I+1 ); %modificar
ImenosF =spdiags( [ -A (1 - B) -C ], [ 0 1 2 ], I-1, I+1 ); %modificar
% Construir matriz cuadrada M = ( I - G ):
% remover primera columna y última fila de I-F
M = ImenosF( : , 2:I ); % resultado es (I-1) x (I-1)
% * Si las condiciones de frontera son de otra forma la construcción
% es más complicada!
% Vector r que usaremos abajo
rvec = zeros( I - 1, 1 );
rvec(1) = -Fabajo * A(1);
SORnIters = 0;
% INDUCCIÓN hacia atrás
for k = N+1:-1:2
    rvec(I-1) = -Farriba(k)* C(I-1);
    % Lado derecho del sistema es q = ImasF * V(:,k) - rvec
    q1 = ImasF * V(:, k) - rvec + d; %%%% RENOMBRO q
    % Resolvemos la ecuación
    % (I - G) * V(2:I, k-1) + rvec = ImasF * V(:, k),

```

```

% es decir,
%   M * V( 2:I, k-1 ) = q
% mediante iteración SOR con 'adivinanza' inicial igual
% a V(2:I) de la anterior iteración
if SORP==1
[Vmonho,nIters] = SORproyectada(M,q1, V(2:I, k), omega, tol, 1000,Pago(2:I));
V(2:I, k-1) = Vmonho;
else
[Vmonho, nIters] = SORIter( M, q1, V(2:I, k), omega, tol, 1000 );
V(2:I, k-1) = max( Vmonho, Pago(2:I) );
end

SORnIters = SORnIters + nIters;

end

V0 = V( :,1);

precio = interp1(S , V0 , S0 , 'linear', 'extrap');

end

```

## F. Contrato Repo con dividendos reinvertidos

```

function [ precio, V0, S, V, M, SORnIters] = BSCNrepo_divre( S0, K, r, gamma, ...
...T, sigma, dS, dt, omega, delta , SORP)
% BSCNrepo_divre
% Calcular el precio de un opción Repo con dividendos reinvertidos
% S0 = Precio inicial del subyacente (puede ser un vector)
% K = Strike
% r = Tasa de interés del subyacente
% delta = Tasa de dividendos del subyacente
% y = Tasa de interés del presta,o
% T = Fecha de vencimiento
% sigma = Volatilidad
% dS = Longitud de los subintervalos de la grilla para la variable espacial
% dt = Longitud de los subintervalos de la grilla para la variable temporal
% omega = parámetro para el uso del metodo SOR
% SORP = si es igual a 1 usa el algoritmo SORProyectada en caso contrario
% usa SORIter

tol = 10^-6; % para la iteración SOR

```

```

minS = 0;
maxS = 4 * K;

I = round( (maxS - minS)/dS );
S = linspace( minS, maxS, I + 1 )';
N = round( T / dt );
t = linspace( 0, T, N+1);

V = zeros( I+1, N+1);

% Función de pago para una repo en T
Pago = max( S - K*exp(gamma*T), 0 );
% También usada con condición final
V( :, N+1 ) = Pago;

% Condiciones de frontera para S -> 0 y S -> infinito
Fabajo = 0;
Farriba = maxS - K*exp(gamma*t);
V(1, :) = Fabajo;
V(I+1, :) = Farriba;

% para la ecuación tradicional de Black-Scholes,
% los coeficientes son independientes del tiempo:
%
a = 0.5 * sigma^2 * S.^2;
b = r * S;
c = -r * ones( I + 1, 1);

nu1 = dt / dS^2;
nu2 = dt / dS;

% construimos los coeficientes para la inducción hacia atrás
% como combinaciones de los anteriores

A = 0.5 * nu1 * a - 0.25 * nu2 * b;
B = - nu1 * a + 0.5 * dt * c;
C = 0.5 * nu1 * a + 0.25 * nu2 * b;

A = A( 2 : I, 1 );
B = B( 2 : I, 1 );
C = C( 2 : I, 1 );

```



```

% Los reunimos en una matriz de I-1 x I+1
ImasF = spdiags( [ A (1 + B) C ], [ 0 1 2 ], I-1, I+1 ); %modificar
ImenosF = spdiags( [ -A (1 - B) -C ], [ 0 1 2 ], I-1, I+1 ); %modificar
% Construir matriz cuadrada M = ( I - G ):
% remover primera columna y última fila de I-F
M = ImenosF( : , 2:I ); % resultado es (I-1) x (I-1)
% * Si las condiciones de frontera son de otra forma la construcción
% es más complicada!
% Vector r que usaremos abajo
rvec = zeros( I - 1, 1 );
rvec(1) = -Fabajo * A(1);
SORnIters = 0;
% INDUCCIÓN hacia atrás
for k = N+1:-1:2
    rvec(I-1) = -Farriba(k)* C(I-1);
    % Lado derecho del sistema es q = ImasF * V(:,k) - rvec
    q1 = ImasF * V(:, k) - rvec; %%% RENOMBRO q
    % Resolvemos la ecuación
    % (I - G) * V(2:I, k-1) + rvec = ImasF * V(:, k),
    % es decir,
    % M * V( 2:I, k-1 ) = q
    % mediante iteración SOR con 'adivinanza' inicial igual
    % a V(2:I) de la anterior iteración
    if SORP==1
        [Vmonho,nIters] = SORproyectada(M,q1, V(2:I, k), omega, tol, 1000,Pago(2:I));
        V(2:I, k-1) = Vmonho;
    else
        [Vmonho, nIters] = SORIter( M, q1, V(2:I, k), omega, tol, 1000 );
        V(2:I, k-1) = max( Vmonho, Pago(2:I) );
    end

    SORnIters = SORnIters + nIters;

end

V0 = V( :,1);

precio = interp1(S , V0 , S0 , 'linear', 'extrap');

end

```

## G. Algoritmo frontera de ejercicio

```
function [E,t] = FronteraEjerREPO_nodiv( S0, K, r, gamma ,T, sigma, dS, ...
... dt, omega,SORP)
% FronteraEjerREPO_nodiv( S0, K, r, y ,T, sigma, dS, dt, omega) :
%calcula la frontera de ejercicio en una secuencia de tiempos ts
% Los parámetros coinciden con los de la función BSCNrepo_nodiv.m
%
% Salidas:
% t = linspace( 0 , T-dt , nc-1 ) = dt * ( 0 : nc-1 ),
% donde nc es el numero de columnas de V
% E un vector fila de dimension nc - 1 tal que E(k) es
% el valor critico de S para el tiempo k * dt (para k = 0,...,nc-1)
% es decir, el valor S* tal que si S < S* al tiempo k * dt,
% entonces es optimo ejercer la opcion.

%Llama a la funcion de precio de la opcion americana

[ precio, V0, S, V, M, SORnIters] = BSCNrepo_nodiv( S0, K, r, gamma , ...
...T, sigma, dS, dt, omega, SORP );

%Determina filas y columnas de la matriz V
[ Im1 , Nm1 ] = size( V );

%Determina el primer parametro de salida, el vector de tiempo
t = linspace( 0 , T-dt , Nm1-1 );

%Define la tolerancia del metodo
tol = 10^-3;

%Crea el vector de precios criticos para ejercer la opcion
E = zeros(1, Nm1-1);

%Recorrido para generar los precios óptimos
for k=1:Nm1-1
    %Calcula los deltas
    Delta = (V(2:end,k)-V(1:end-1,k))/(dS); % diferencias hacia atras
    %Halla el primer delta con valor 1
    e = find(1-Delta<=tol,1,'first');
    %Calcula dos puntos sobre X (S_e) y Y (delta_e)
    s_e_1=(e-1)*dS;
    s_e_2=(e-2)*dS;
    % diferencias hacia atras de orden 2
    delta_e_1 = (3*V(e-1,k)-4*V(e-2,k)+V(e-3,k))/(2*dS);
```

```
    delta_e_2 = (V(e-1,k)-V(e-3,k))/(dS); % diferencias hacia atras de orden 2
%Calcula la pendiente de la rectas
    m=(delta_e_1-delta_e_2)/(s_e_1-s_e_2);
%Halla el punto de corte de la recta
    b = delta_e_1 - s_e_1*m;
%Halla el precio optimo con el punto de corte y pendiente calculados,
    E(k) = (1-b)/m;
end
end
```