



UNIVERSIDAD ADOLFO IBÁÑEZ

DERIVADOS AVANZADOS

PRICING A BOND WITH MATLAB

Alumno:

Felipe Ignacio Durán Aranda

Profesor:

Jacques Burrus

Ayudantes:

- Alejandro Olguín
- Guillermo Sepúlveda

February 25, 2021

Contents

1	Prefacio	2
2	Pregunta Uno	2
3	Pregunta Dos	3
4	Pregunta Tres	4
5	Pregunta Cuatro	5
6	Pregunta Cinco	5
6.1	getYearFraction(t, T)	5
6.2	getDiscountFactor(t, T, r)	6
6.3	getCoupon(t, i, γ, N)	6
6.4	getBondValue5(t, N, r, γ)	6
7	Pregunta Seis	7
8	Pregunta Siete	7
9	Pregunta Ocho	9
10	Pregunta Nueve	10
11	Pregunta Diez	12

1 Prefacio

Mediante este informe se busca presentar los resultados obtenidos a partir de la actividad "*Pricing A Bond With Matlab*", en la cual se pide obtener el precio de un bono respecto a diversos parámetros, metodologías y requerimientos.

2 Pregunta Uno

Para esta pregunta, se pide calcular el valor de un bono a partir de la siguiente función:

$$V(t) = \sum_{i=1}^N \frac{c_i}{(1+r)^{\frac{T_i-t}{360}}} \quad (1)$$

Donde los valores de la función para los cupones y para el tiempo de pago de estos vienen dados de la siguiente forma:

$$c_i = \begin{cases} \gamma & \text{if } i < N \\ 1 + \gamma & \text{otherwise} \end{cases} \quad (2)$$
$$T_i = 180 \times i$$

Bajo estas ecuaciones, se calcula el valor del bono en base a los siguientes parámetros:

$$\begin{aligned} t &= 0 \\ N &= 4 \\ r &= 0.03 \\ \gamma &= 0.05 \end{aligned}$$

$$VP1 = C1 + C2 + C3 + C4 \quad (3)$$

A partir de las formulas anteriormente presentadas e ingresándolas en *Matlab*, se obtuvieron los siguientes resultados:

$$VP1 = 0.0493 + 0.0485 + 0.0478 + 0.9897$$

$$VP1 = 1.1354$$

En base a los resultados, podemos observar que los cupones del bono tienen un valor decreciente, concordante con el comportamiento que esperaríamos frente a un factor de descuento.

3 Pregunta Dos

De manera similar a la pregunta anterior, se nos pide calcular el valor del bono, utilizando las fórmulas presentadas con anterioridad, sin embargo, utilizando esta vez operaciones matriciales. Para esta pregunta, se utilizó la propiedad de convertibilidad de matrices en escalares a partir de una multiplicación matricial de la forma:

$$[1xN] [Nx1] = [1x1] \quad (4)$$

En la cual, como se puede observar, se obtiene como resultado un escalar. Para aplicar esta forma, se calculó un vector filas y un vector columnas, los cuales contenían el factor de descuento del bono y el valor del cupón respectivamente, utilizando los mismos parámetros de la pregunta anterior, tal como se puede observar de la siguiente manera:

$$Filas = \left[\frac{1}{(1+0.03)^{\frac{180*1-0}{360}}} \quad \frac{1}{(1+0.03)^{\frac{180*2-0}{360}}} \quad \frac{1}{(1+0.03)^{\frac{180*3-0}{360}}} \quad \frac{1}{(1+0.03)^{\frac{180*4-0}{360}}} \right] \quad (5)$$

$$Columnas = \begin{bmatrix} 0.05 \\ 0.05 \\ 0.05 \\ 1.05 \end{bmatrix} \quad (6)$$

De esta forma al multiplicar el vector *Filas* por el vector *Columnas* podemos obtener el escalar que correspondería al valor del bono, el cual, como era de esperar, resultó ser igual al de la pregunta anterior:

$$VP2 = \left[\frac{0.05}{(1+0.03)^{\frac{180*1-0}{360}}} + \frac{0.05}{(1+0.03)^{\frac{180*2-0}{360}}} + \frac{0.05}{(1+0.03)^{\frac{180*3-0}{360}}} + \frac{1.05}{(1+0.03)^{\frac{180*4-0}{360}}} \right] \quad (7)$$

$$VP2 = 1.1354$$

4 Pregunta Tres

El objetivo de esta pregunta consiste en programar un ciclo en *Matlab* que calcule el valor de un bono utilizando los mismos parámetros presentados para la pregunta uno, de la cuál se esperaría obtener el mismo resultado.

El bucle utilizado solamente plantea convertir la sumatoria presentada en la ecuación uno en un ciclo *for*, el cual es presentado a continuación:

```
VP3=0;
for i=1:N
    if (i<N)
        ci=y;
    else
        ci=1+y;
    end
    Ti=180*i;
    aux=(ci)/((1+r)^((Ti-t)/360));
    VP3=VP3+aux;
end
VP3
```

La explicación del ciclo *for* anterior, consiste en calcular los términos de la sumatoria presentada en la ecuación uno como una función que se repite en múltiples pasos, efectivamente según el número de cupones que posea nuestro bono. Luego, guarda y suma estos valores en una variable creada fuera del bucle, retornando finalmente esta misma resultando de la sumatoria de todos los valores del bono según sus cupones y los factores de descuento, tal como se indica en la ecuación uno. Finalmente el valor del bono mediante esta metodología es:

$$VP3 = 1.1354$$

Siendo este igual al valor obtenido en todas las preguntas anteriores, apuntando a que el bucle fue implementado de la manera correcta.

5 Pregunta Cuatro

Para esta pregunta, se solicita crear una función *getBondValue4*(t, N, r, γ), la cuál debería calcular el valor del bono a partir de los argumentos: el tiempo (t), el número de cupones (N), el factor de descuento (r) y el interés del bono (γ). Esta función consiste en implementar simplemente el código del bucle utilizado en la pregunta 4 para parámetros generales, lo cuál se realizó a través de la opción de creación de funciones de *Matlab*, la cuál nos permite realizar dicho cambio solamente trasladando el código del bucle a esta realizando pequeñas modificaciones. La función se puede ver en el código y programa adjunto en la carpeta *Matlab*.

Aplicando esta función para los parámetros utilizados en la pregunta uno, podemos ver que obtenemos el mismo resultado en *Matlab* que para la pregunta uno, siendo el valor:

$$VP4 = \text{getBondValue4}(0, 4, 0.03, 0.05) = 1.1354$$

6 Pregunta Cinco

El objetivo de esta pregunta consiste en crear cuatro funciones que permitan calcular los diversos parámetros de la primera ecuación, presentada con anterioridad, y que estos puedan ser llamados por una función llamada *getBondValue5*, la cuál debe retornar el valor del bono a partir de los parámetros ingresados. Las funciones programadas para cumplir con este objetivo son presentadas a continuación (El código de las funciones implementadas queda adjunto en el archivo de *Matlab*):

6.1 *getYearFraction*(t, T)

Esta función pretende obtener la fracción del año para el calculo del bono. Corresponde al exponente del factor de descuento del bono, y la fórmula para su calculo, expresada de la siguiente manera:

$$\begin{aligned} \text{getYearFraction}(t, T) = y &= \frac{T-t}{360} \\ T &= 180 * i \end{aligned} \tag{8}$$

6.2 getDiscountFactor(t,T,r)

Esta función tiene como objetivo determinar el factor de descuento, incluyendo el cálculo sobre la fracción del año correspondiente, que se obtiene a través de la función *getYearFraction(t, T)*, llamándola dentro de esta nueva función en *Matlab*. La función corresponde a la siguiente:

$$\begin{aligned} y &= \text{getYearFraction}(t, T) \\ \text{getDiscountFactor}(t, T, r) &= d = \frac{1}{(1+r)^y} \end{aligned} \quad (9)$$

6.3 getCoupon(t,i,γ,N)

La función *getCoupon(t,i,γ,N)*, tiene como cometido obtener el valor del cupón correspondiente dependiendo del número del cupón y del interés (γ), tal como se presenta en la ecuación dos. Para esta función en específico, se añadieron una serie de condicionales necesarios para poder obtener el valor del bono correctamente, dado que sería necesario posteriormente para las preguntas siete y ocho. Estos condicionales, principalmente dependen del tiempo (t) y del tiempo de pago de los cupones (T), así como del número del cupón respectivo y el total de cupones del bono.

6.4 getBondValue5(t,N,r,γ)

Finalmente, la función *getBondValue(t,N,r,γ)* utiliza todas las funciones anteriormente programadas para calcular el valor del bono, explicándose someramente de la siguiente manera:

$$\begin{aligned} ci &= \text{getCoupon}(t, i, y, N) \\ Ti &= 180 * i \\ \text{getDiscountFactor}(t, Ti, r) &= d = \frac{1}{(1+r)^y} \\ \text{getBondValue5}(t, N, r, y) &= \sum_{i=1}^N ci * d \end{aligned} \quad (10)$$

Una vez hecho esto, para probar la función final y comprobar que tuviese un correcto funcionamiento, se ingresaron los parámetros estándar planteados para la pregunta uno, para los cuales la función arrojó el siguiente resultado:

$$VP5 = \text{getBondValue5}(0, 4, 0.03, 0.05) = 1.1354$$

Siendo este el resultado esperable y deseado para la función, acorde a lo obtenido de las múltiples otras formas y métodos planteados en este informe.

7 Pregunta Seis

En esta pregunta, se solicita eliminar las variables previas existentes, almacenar las variables de los parámetros estándar, y ejecutar la función obtenida en la pregunta anterior utilizando estos parámetros, para lo cuál se utilizó el siguiente código:

```
clearvars -except t N r y
VP6=getBondValue5(t,N,r,y)
```

Logrando efectivamente borrar las variables almacenadas, exceptuando aquellas indicadas, correspondientes a las variables de los parámetros estándar, consiguientemente llamando la función *getBondValue5(t,N,r,y)* para volver a obtener el valor del bono.

$$VP6 = \text{getBondValue5}(t,N,r,y) = 1.1354$$

En la cual, como se puede ver en el ejercicio presentado, se logra obtener el mismo resultado esperado de la pregunta anterior, así como la conclusión de las otras metodologías planteadas en el resto del informe.

8 Pregunta Siete

En esta pregunta, se nos pide graficar el comportamiento del valor del bono obtenido a través de la función *getBondValue5(t,N,r,y)*, según variaciones en la variable tiempo, planteada como t en esta función, con cada cambio equivalente a un período de 10 días en un rango consistente de un tiempo mínimo (tmin) y tiempo máximo (tmax). Para esto, se nos pide confeccionar una función que, considerando estos nuevos requerimientos; logre graficar el vector de valores obtenidos, a través de la siguiente función *plotBondValue7(tmin,tmax,deltat,r,N,y)*:

```
range=tmin:deltat:tmax;
mlocale=[1:length(range)]; %ok<NBRAK>
auxmlocale=1;
for a=drange(range)
    auxlocale=getBondValue5(a,N,r,y);
    mlocale(auxmlocale)=auxlocale;
    auxmlocale=auxmlocale+1;
end
plotBondValue7=plot(range,mlocale);
```


Cabe mencionar que con esta pregunta nos podemos percatar que el valor del bono tiene una dependencia con el tiempo, específicamente cuando hablamos del valor de los cupones de este, dado que a medida que pasa el tiempo, el bono va perdiendo valor, dado que disminuye la cantidad de cupones que recibimos.

Finalmente a continuación es presentado el gráfico correspondiente al valor del bono en función del tiempo t .

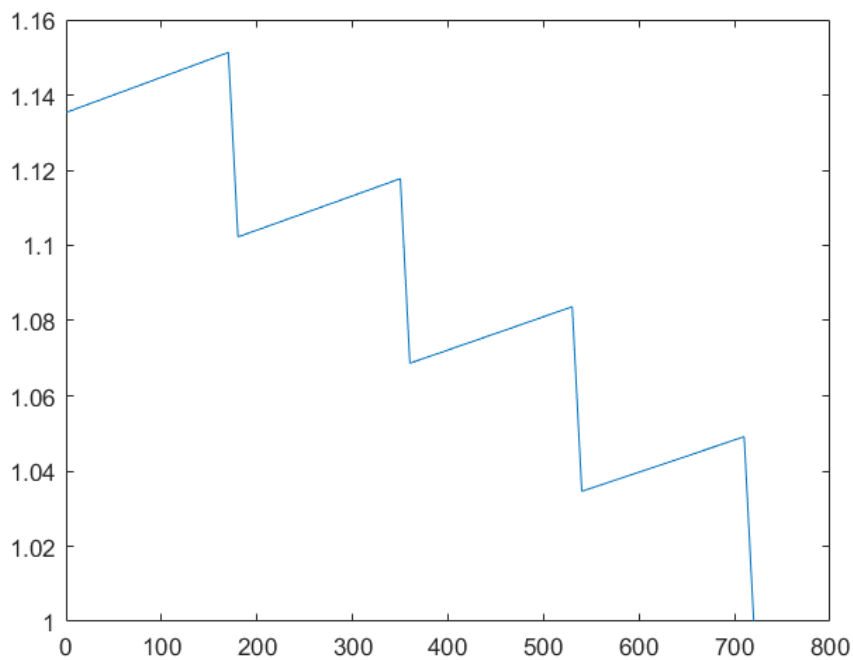


Figure 1: "Valor del Bono en función del tiempo t "

Como podemos observar, el valor del bono tiende a crecer a medida que nos acercamos a la fecha de pago de los cupones, sin embargo, una vez pasado este cupón, podemos ver como el valor del bono disminuye dado que correspondería a un cupón que no percibiríamos del bono.

9 Pregunta Ocho

Esta pregunta tiene como objetivo añadir al gráfico anterior algunos elementos, tales como: título, rótulos en los ejes, y una leyenda, así como de cambiar el tipo de línea y el color de esta. El gráfico resultante con dichos requerimientos se presenta a continuación:

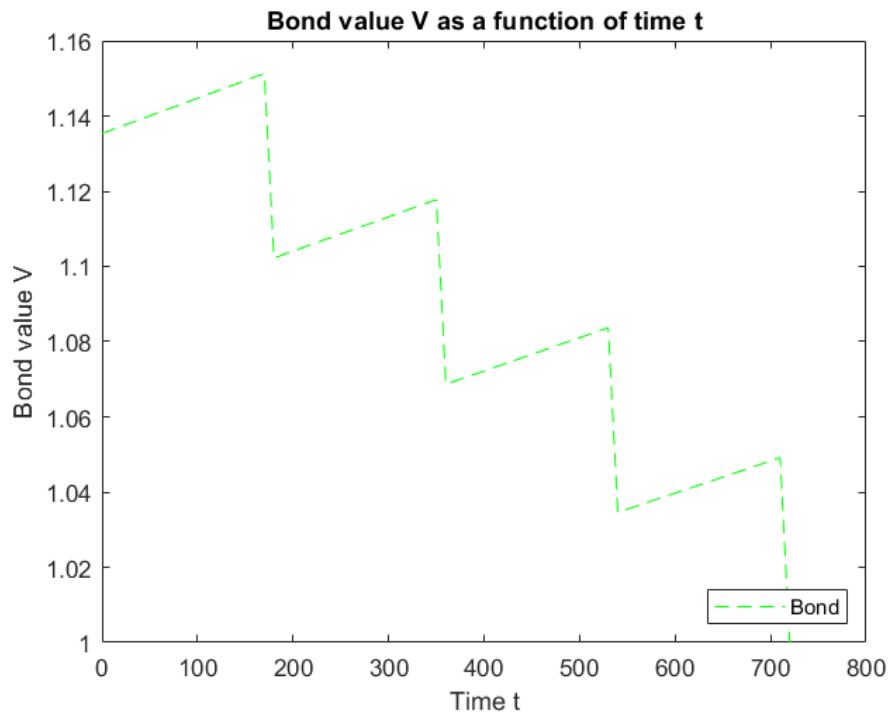


Figure 2: "Bond Value V as a function of time t "

10 Pregunta Nueve

Para esta pregunta, se solicita implementar una función que pueda graficar el comportamiento del valor del bono respecto a variaciones, tanto en el factor de descuento (r), como en el tiempo (t), siendo el último similar a como se hizo en la pregunta siete. Para expresar estas variaciones en dos variables, se utilizó un método similar al empleado para recorrer una matriz, con dos ciclos *for*, o también conocido como un "*for anidado*". La función utilizada para este propósito, *surfaceBondValue9*(*tmin,tmax,deltat,rmin,rmax,deltar,N,y*) es presentada a continuación:

```
range=tmin:deltat:tmax;
ranger=rmin:deltar:rmax;
mlocale=zeros(length(range),length(ranger));
auxmlocale=1;
auxmlcaler=1;
for a=drange(range)
    for b=drange(ranger)
        auxaux=getBondValue5(a,N,b,y);
        mlocale(auxmlocale,auxmlcaler)=auxaux;
        auxmlcaler=auxmlcaler+1;
    end
    auxmlcaler=1;
    auxmlocale=auxmlocale+1;
end
auxsurf=surf(ranger,range,mlocale);
surfaceBondValue9=auxsurf,
title('Bond value V as a function of time t and rate r'),
xlabel('Rate r'),ylabel('Time t'),zlabel('Bond Value');
```

La función plantea el uso de un bucle *for anidado*, para generar y recorrer una matriz de dimensiones iguales a los rangos de las variaciones en los parámetros planteados, mientras hace uso de la función *getBondValue5* para obtener el valor del bono en cada iteración. Utilizando dicho código, se obtiene el siguiente gráfico:

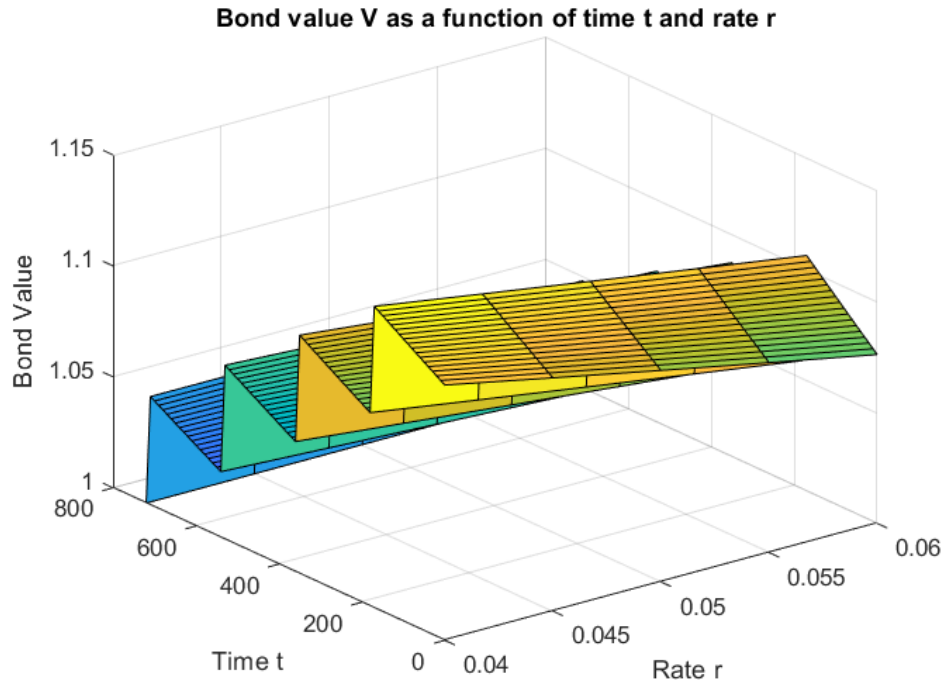


Figure 3: "Bond Value V as a function of time t and rate r"

Como podemos observar, el valor del bono tiende a decrecer a medida que aumenta el factor de descuento (r), y a medida que pasa cada cupón, comportamiento esperable, y que es acorde a lo obtenido en la pregunta anterior.

11 Pregunta Diez

Finalmente, como última actividad, se nos pide encontrar el factor de descuento correspondiente al cual el valor del bono, evaluado con los parámetros estándar presentados en la pregunta uno, sería 1. para obtener este resultado es necesario crear una función que recoja estos parámetros como argumentos, es decir, (γ) , (t) , y (N) , y que sea capaz de encontrar el valor de r como una incógnita en la ecuación, bajo el supuesto que $r > (-1)$, con tal de no indefinir el denominador de la fórmula presentada en la primera ecuación.

Para poder plantear y resolver la ecuación a través de una función en *Matlab*, se utilizó el siguiente código:

```
locale = 1:1:N;
localec = 1:1:N;
syms r;
eqn = 0;
for i = 1:N
    T = 180 * i;
    ci = getCoupon(t, i, y, N);
    yf = getYearFraction(t, T);
    locale(i) = yf;
    localec(i) = ci;
    eqn = eqn + (ci / (1 + r) ^ yf);
end
eqn = eqn - 1;
assume(r, 'real');
getDiscountRate = solve(eqn, r > -1);
%de esta manera la ecuacion no se indefine
```

El código anterior, tiene como finalidad, el declarar a r como una variable simbólica, que resulta el equivalente a declarar una incógnita en una ecuación, luego, utiliza esta variable en un ciclo *for*, con el objetivo de lograr confeccionar la fórmula correspondiente a la ecuación uno, con los parámetros correspondientes. Una vez declarada esta ecuación, se hace uso de un *solver* genérico de *Matlab* para resolver la ecuación, según la restricción impuesta de que el valor de $r > (-1)$, para la cual finalmente se obtiene el siguiente resultado:

$$VP10 = \text{getDiscountRate}(0, 4, 0.05) = r = 0.1025$$

Comparando el resultado obtenido en esta pregunta con el conseguido en la pregunta uno y similares, podemos observar que para llevar el valor del bono a 1 se necesita un factor de descuento (r) más grande. Este comportamiento era esperable, dado que logra disminuir el valor de cada uno de los cupones de este, llevando efectivamente el valor total del bono a 1 cuando nos encontramos con un $r=0.1025$. Adicionalmente, este mismo comportamiento se puede observar en la pregunta nueve, en la cual el valor del bono es decreciente respecto a r , adicional a la tendencia de disminuir su valor a medida que pasa el tiempo.