

Informe Tarea 2: Derivados Avanzados

Martín Palacios Undurraga

En este informe se explicarán los procedimientos y resultados realizados en el documento Excel entregado junto con este documento para desarrollar los ejercicios 1 a 10 de “PRICING A BOND WITH MATLAB”, archivo proporcionado por el profesor.

Al inicio del archivo vemos información acerca de cómo se calcula un bono, la función por tramos de asignación de cupones y cuando se pagan los cupones. Adjunto las fórmulas a continuación ya que estas serán usadas posteriormente para el código.

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

Where the coupon c_i and payment dates T_i are given by:

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

Sumado a esto, se nos proporcionan los parámetros default para la tarea los cuales son:

Default parameters:

Unless specified otherwise, the pricing parameters are:

$$t = 0$$

$$N = 4$$

$$r = 0.03$$

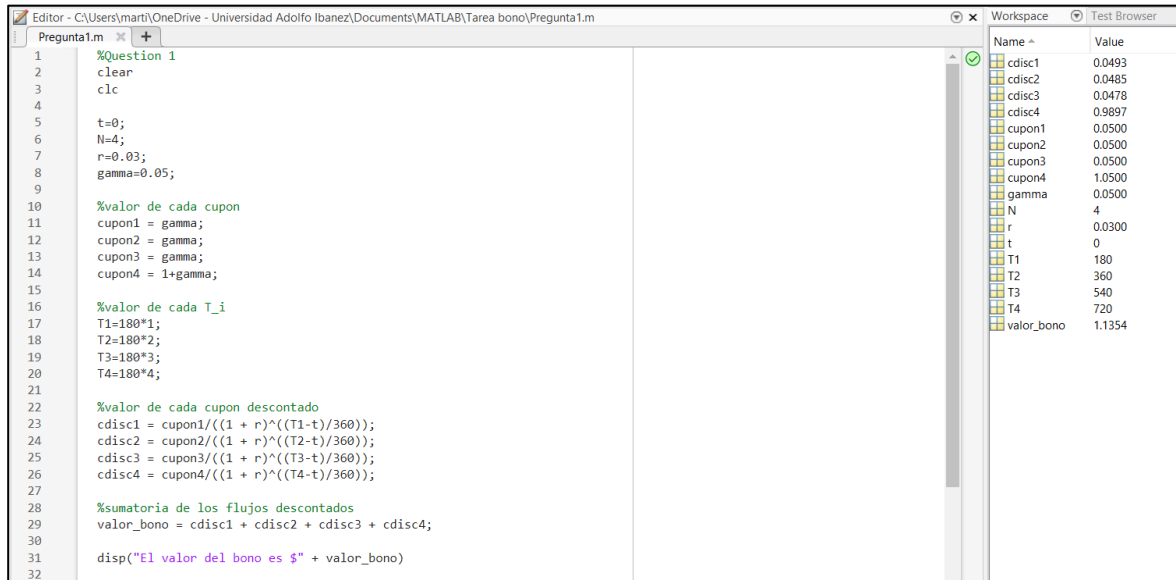
$$\gamma = 0.05$$

A continuación, explicare en detalle las preguntas y como fueron resueltas.

Question 1:

Compute the value V in the Matlab workplace for the aforementioned parameter values.

Para la pregunta 1 utilizamos Matlab para calcular el precio del bono, pero de una forma muy ineficiente, en un script asigné valores para cada componente, luego realicé la operación de descontar cada flujo y finalmente los sumé en una nueva variable llamada valor_bono tal y como se muestra a continuación en el script Preguntal.m:



```

1 %Question 1
2 clear
3 clc
4
5 t=0;
6 N=4;
7 r=0.03;
8 gamma=0.05;
9
10 %valor de cada cupon
11 cupon1 = gamma;
12 cupon2 = gamma;
13 cupon3 = gamma;
14 cupon4 = 1+gamma;
15
16 %valor de cada T_i
17 T1=180*1;
18 T2=180*2;
19 T3=180*3;
20 T4=180*4;
21
22 %valor de cada cupon descontado
23 cdisc1 = cupon1/((1 + r)^((T1-t)/360));
24 cdisc2 = cupon2/((1 + r)^((T2-t)/360));
25 cdisc3 = cupon3/((1 + r)^((T3-t)/360));
26 cdisc4 = cupon4/((1 + r)^((T4-t)/360));
27
28 %sumatoria de los flujos descontados
29 valor_bono = cdisc1 + cdisc2 + cdisc3 + cdisc4;
30
31 disp("El valor del bono es $" + valor_bono)
32

```

Name	Value
cdisc1	0.0493
cdisc2	0.0485
cdisc3	0.0478
cdisc4	0.9897
cupon1	0.0500
cupon2	0.0500
cupon3	0.0500
cupon4	1.0500
gamma	0.0500
N	4
r	0.0300
t	0
T1	180
T2	360
T3	540
T4	720
valor_bono	1.1354

Al correr esta función, se genera un output de texto y numero dada la última línea de código que agregue para que todo este mas claro y presentable de una manera mas limpia. Habiendo entendido como funciona el script podemos ver que el output nos dice que el valor del bono dado los parámetros dados es de \$1.1354 (Para efectos de este archivo el “.” será el separador de decimales, de esta manera seremos consistentes con la notación de Matlab).

Question 2:

Construct the column vectors and of length N in the Matlab workplace and recompute the value V with vectorial operations.

En la pregunta 2 se nos solicita volver a calcular el valor del bono, pero ahora con operaciones vectoriales para los vectores generados. Esta manera es bastante mas optimizada para el calculo del valor del bono, pero aún se puede mejorar. A continuación, dejo el código en el que se observan los vectores y las operaciones vectoriales para llegar al valor del bono que es consistente con el obtenido en la pregunta anterior.

The screenshot shows the MATLAB Editor with a script named 'Pregunta2.m' and the Workspace window. The script calculates the value of a bond using a loop and vector operations. The Workspace window lists the variables and their values.

```

7 N=4;
8 r=0.03;
9 gamma=0.05;
10
11 %vector de cupones
12 vcupones = [gamma, gamma, gamma, 1+gamma];
13
14 %vector T_i
15 vTi = [180*1, 180*2, 180*3, 180*4];
16
17 %calculamos los flujos descontados en forma vectorial
18 vdisc = [0 0 0 0];
19
20 for i = 1:N
21     vdisc(i) = vcupones(i)/((1 + r)^(vTi(i)-t)/360));
22 end
23
24 %calculamos el valor del bono usando operaciones vectoriales
25 V = sum(vdisc);
26
27 disp("el valor del bono es de $" + V)
28

```

Workspace:

Name	Value
cdisc1	0.0493
cdisc2	0.0485
cdisc3	0.0478
cdisc4	0.9897
cupon1	0.0500
cupon2	0.0500
cupon3	0.0500
cupon4	1.0500
gamma	0.0500
N	4
r	0.0300
t	0
T1	180
T2	360
T3	540
T4	720
valor_bono	1.1354

Command Window:

```

El valor del bono es $1.1354
fx >>

```

Podemos verificar que de esta manera las variables estaban almacenadas como vectores en la siguiente imagen:

The screenshot shows the MATLAB Variable Editor with three variables: vTi, vdisc, and vcupones. Each variable is a 1x4 double array.

vTi						
	1	2	3	4	5	6
1	180	360	540	720		
2						

vdisc						
	1	2	3	4	5	6
1	0.0493	0.0485	0.0478	0.9897		
2						

vcupones						
	1	2	3	4	5	6
1	0.0500	0.0500	0.0500	1.0500		
2						
3						

Question 3:

Write a script in a Matlab m file that computes the value V with a loop. Verify the result.

Para la pregunta 3 se nos solicita generar un script (archivo .m) que calcule el valor del bono utilizando un loop, si bien para la pregunta 2 ya usé un for loop procedí a hacerlo nuevamente con algunas variaciones tal y como se muestra a continuación.

```

1 %Question 3
2 clear
3 clc
4
5 %Write a script in a Matlab m file that computes the value V with a loop.
6 t=0;
7 N=4;
8 r=0.03;
9 gamma=0.05;
10
11 %vector de cupones
12 vcupones = [gamma, gamma, gamma, 1+gamma];
13
14 %vector T_i
15 vTi = [180*1, 180*2, 180*3, 180*4];
16
17 %calculamos los flujos descontados en forma vectorial
18 vdisc = [0 0 0 0];
19
20 for i = 1:N
21     vdisc(i) = vcupones(i)/((1 + r)^((vTi(i)-t)/360));
22 end
23
24 %calculamos el valor del bono usando un loop que suma los componentes
25
26 V=0;
27
28 for i = 1:N
29     V = V + vdisc(i);
30 end
31
32
33 disp("el valor del bono es de $" + V)

```

Name	Value
gamma	0.0500
i	4
N	4
r	0.0300
t	0
V	1.1354
vcupones	[0.0500,0.0500,0.0500,1.0500]
vdisc	[0.0493,0.0485,0.0477,0.0469]
vTi	[180,360,540,720]

Podemos ver que para este método de cálculo utilizo loop y vectores para facilitar el calculo dentro de un mismo script.

Question 4:

Write a function `getBondValue4(t, N, r, γ)` that implements the previous script for general parameter values. Compute the value V with the previous parameters.

Para la pregunta 4 genere una función en un script aparte que implementara lo programado en el script anterior, pero a diferencia que al estar en un script de función uno lo puede llamar para poder tener un código más limpio en el archivo main.m que es donde se desarrolla la mayoría de la tarea (a excepción de las preguntas que solicitan explícitamente generar otro script). La función queda de la siguiente forma:

```
Editor - C:\Users\marti\OneDrive - Universidad Adolfo Ibanez\Documents\MATLAB\Tarea bono\pregunta4.m
Pregunta3.m x pregunta4.m x main.m x +
1 %pregunta 4
2 function V = pregunta4(t,N,r,gamma)
3
4 vcupones = [gamma, gamma, gamma, 1+gamma];
5
6 %vector T_i
7 vTi = [180*1, 180*2, 180*3, 180*4];
8
9 %calculamos los flujos descontados en forma vectorial
10 vdisc = [0 0 0 0];
11
12 for i = 1:N
13     vdisc(i) = vcupones(i)/((1 + r)^((vTi(i)-t)/360));
14 end
15
16 %calculamos el valor del bono usando un loop que suma los componentes
17
18
19 V=0;
20
21 for i =1:N
22     V = V + vdisc(i);
23 end
24
25 disp("el valor del bono es de $" + V)
```

Podemos ver que el código es básicamente el mismo que en la pregunta 3 solo que ahora simplemente se puede usar de una manera distinta.

Question 5:

- Write a function `getYearFraction(t,T)` that computes $y = (T - t)/360$.
- Write another function `getDiscountFactor(t,T,r)` that computes the discount factor $\frac{1}{(1+r)^y}$. Hint: This function must call the function `getYearFraction(t,T)`.
- Write a function `getCoupon(t, i, γ , N)` that returns the correct value of the coupon c_i . NB: There is no coupon after the payment date T_i .
- Write a function `getBondValue5(t, N, r, γ)` that computes the bond value V using all previous functions. Check the result with the previous parameters.
Hint: Use one m file per function.

Para la pregunta 5 se nos solicita generar distintas funciones las cuales deben cumplir un cierto rol, deajo a continuación las funciones generadas con un breve comentario en el archivo de código.

```
Editor - C:\Users\marti\OneDrive - Universidad Adolfo Ibanez\Documents\MATLAB\Tarea bono\getYearFraction5.m
main.m x getYearFraction5.m x +
1 %Question 5: YearFraction
2 function y = getYearFraction5(t,T_i)
3 y=(T_i-t)/360;
4 end
5 %La función getYearFraction5() toma dos argumentos de entrada: t y T_i.
6 %La función calcula la fracción de año entre t y T_i y devuelve el
7 %resultado como y.
8 %La fracción de año es un número entre 0 y 1 que representa la
9 %porción del año que ha transcurrido desde T_i.
```

```
Editor - C:\Users\marti\OneDrive - Universidad Adolfo Ibanez\Documents\MATLAB\Tarea bono\getDiscountFactor5mod.m
main.m x getDiscountFactor5mod.m x +
1 %Question 5: Discount factor modified for vectorial operations
2 function dfact5 = getDiscountFactor5mod(t, T, r)
3 % Raise the matrix to the power of the vector
4 dfact5 = (1 / (1+r)) .^ getYearFraction5(t, T);
5 end
6
7 %La función getDiscountFactor5mod() calcula el factor de descuento para
8 % cada vencimiento en un vector. La función es útil para calcular el
9 % valor presente de un flujo de efectivo que vence en diferentes fechas.
```

```
Editor - C:\Users\marti\OneDrive - Universidad Adolfo Ibanez\Documents\MATLAB\Tarea bono\getCoupon5.m
main.m x getCoupon5.m x +
1 %Question 5: Get coupon
2 function coup5 = getCoupon5(t, i, gamma, N)
3 T_i= 180*i;
4 if i < N
5 coup5 = (t<T_i) * gamma;
6 else
7 coup5 = (t<T_i) * (gamma + 1) + (t == T_i);
8 end
9 end
10
11 %La función getCoupon5() calcula el cupón para una fecha determinada,
12 % número de cupón, cantidad de cupón y número de cupones.
13 %La función primero calcula la fecha de vencimiento para el i-ésimo cupón,
14 % T_i. Luego, verifica si la fecha actual es menor a la fecha de
15 % vencimiento. Si la fecha actual es menor a la fecha de vencimiento,
16 % la función devuelve la cantidad de cupón gamma. Si la fecha actual es
17 % igual a la fecha de vencimiento, la función devuelve la cantidad de cupón
18 % gamma más 1.
```

```
Editor - C:\Users\marti\OneDrive - Universidad Adolfo Ibanez\Documents\MATLAB\Tarea bono\getBondValue5
main.m x getCoupon5.m x plotBondValue7.m x getBondValue5.m x +
1 %Question 5: Get Bond Value
2 function V = getBondValue5(t,N, r,gamma)
3 V=0;
4 for i = 1:N
5     T_i=180*i;
6     V = V + getCoupon5(t,i,gamma,N).*getDiscountFactor5mod(t,T_i, r);
7 end
8 end
9 %La función getBondValue5() calcula el valor de un bono para una fecha
10 % determinada, número de cupones, tasa de descuento y cantidad de cupón.
11 %La función primero calcula el valor presente de cada cupón utilizando la
12 % función getDiscountFactor5mod(). Luego, suma el valor presente de los
13 % cupones para obtener el valor del bono.
14
```

En esta pregunta no se nos solicita calcular el valor del bono, sino que simplemente generar las funciones en distintos scripts de función que conversen entre si para poder usarlos más tarde.

Question 6:

Write a main file `main.m` that clears all existing variables, stores the parameter values, calls the `getBondValue5` function, and prints the result in the workspace.

La pregunta 6 nos pide escribir un `main.m` que permita obtener el valor del bono con el trabajo realizado anteriormente. La limpieza de las variables se realiza con el comando `clear` y posteriormente usamos el comando `clc` para limpiar el terminal. Cabe destacar que al igual que en los cálculos anteriores, al calcular el valor del bono con la nueva función generada en la pregunta 5 el resultado es totalmente consistente con el de las preguntas anteriores. A continuación, dejo la parte del código `main.m` dedicada a la pregunta 6:

```
12 %answers question 6
13 clear
14 clc
15
16 t=0;
17 N=4;
18 r=0.03;
19 gamma=0.05;
20
21 auxp6 = getBondValue5(t,N,r,gamma);
22 disp("El valor del bono es de $" + auxp6);
```

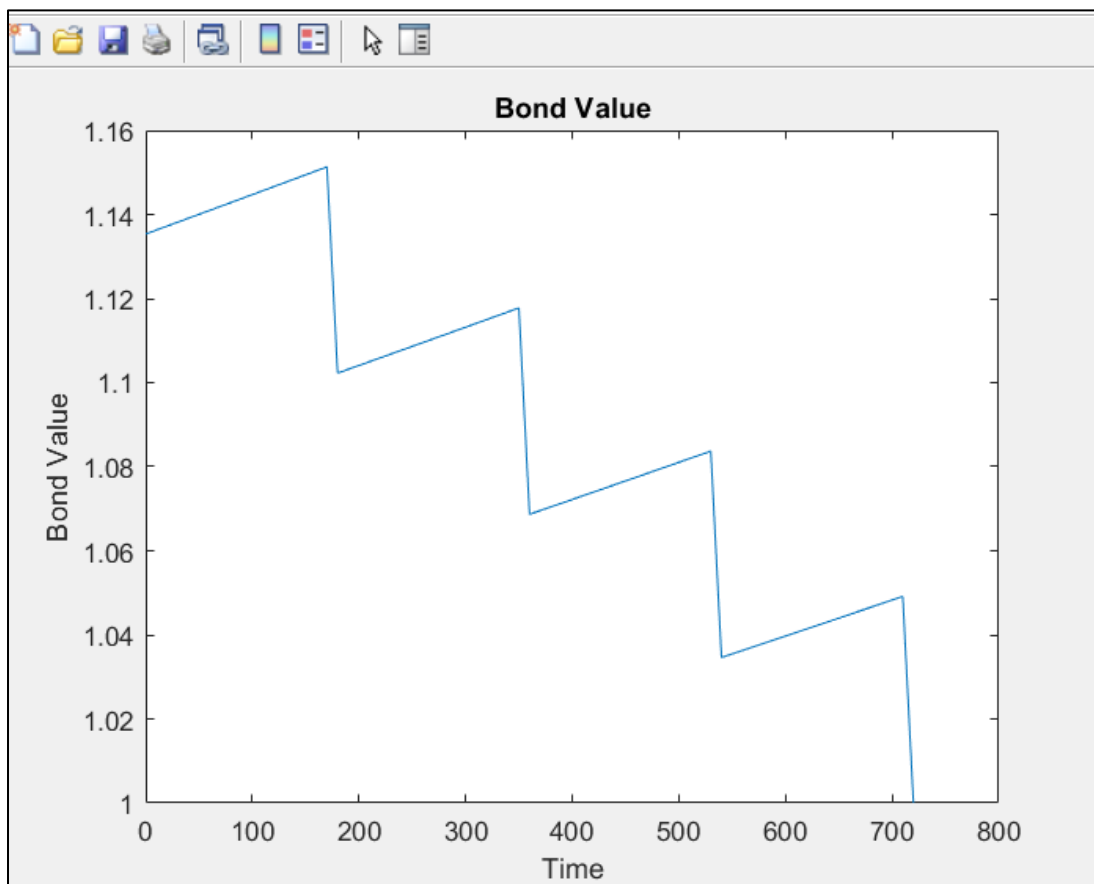
Question 7:

Write a function `plotBondValue7(t_{min} , t_{max} , Δt , r , N , γ)` that plots the bond value with time t ranging from t_{min} to t_{max} with a time step Δt . This function must reuse `getBondValue5`. Make the plot with $t_{min}=0$, $t_{max}=720$, and $\Delta t=10$ from the main file.

De acuerdo con lo solicitado, desarrollo el código de la siguiente manera:

```
Editor - C:\Users\marti\OneDrive - Universidad Adolfo Ibanez\Documents\MATLAB\Tarea bono\plotBondValu
main.m x getCoupon5.m x plotBondValue7.m x getBondValue5.m x +
1 %question 7
2 function plotBondValue7(tmin, tmax, deltat, r, N, gamma)
3 %
4 t = linspace(tmin, tmax, (tmax-tmin)/deltat +1);
5
6 bondValues = getBondValue5(t,N,r,gamma);
7
8 %graficamos
9 plot(t,bondValues);
10 title("Bond Value")
11 xlabel("Time")
12 ylabel("Bond Value")
13 end
14
```

Como se puede observar, para este código generé el vector t (tiempos) con la función `linspace(x1, x2, n)` propia de Matlab que lo que hace es generar un vector con 2 valores límites y un número de componentes, la gracia de este vector es que tiene la particularidad de que sus componentes son equidistantes entre si lo que lo hace perfecto para casos como este donde debemos generar un gráfico limpio que represente bien la situación. Posteriormente se calculan los valores del bono para cada t y luego se grafican dando origen al siguiente gráfico:



En el gráfico podemos ver que a medida que t aumenta, el valor del bono aumenta hasta el momento crítico donde se paga un cupón, este pago produce una caída drástica en el valor del bono generando un nuevo mínimo dado que los flujos futuros del bono ahora son menores, este proceso se repite hasta que no quedan flujos y el valor del bono se desploma.

Question 8:

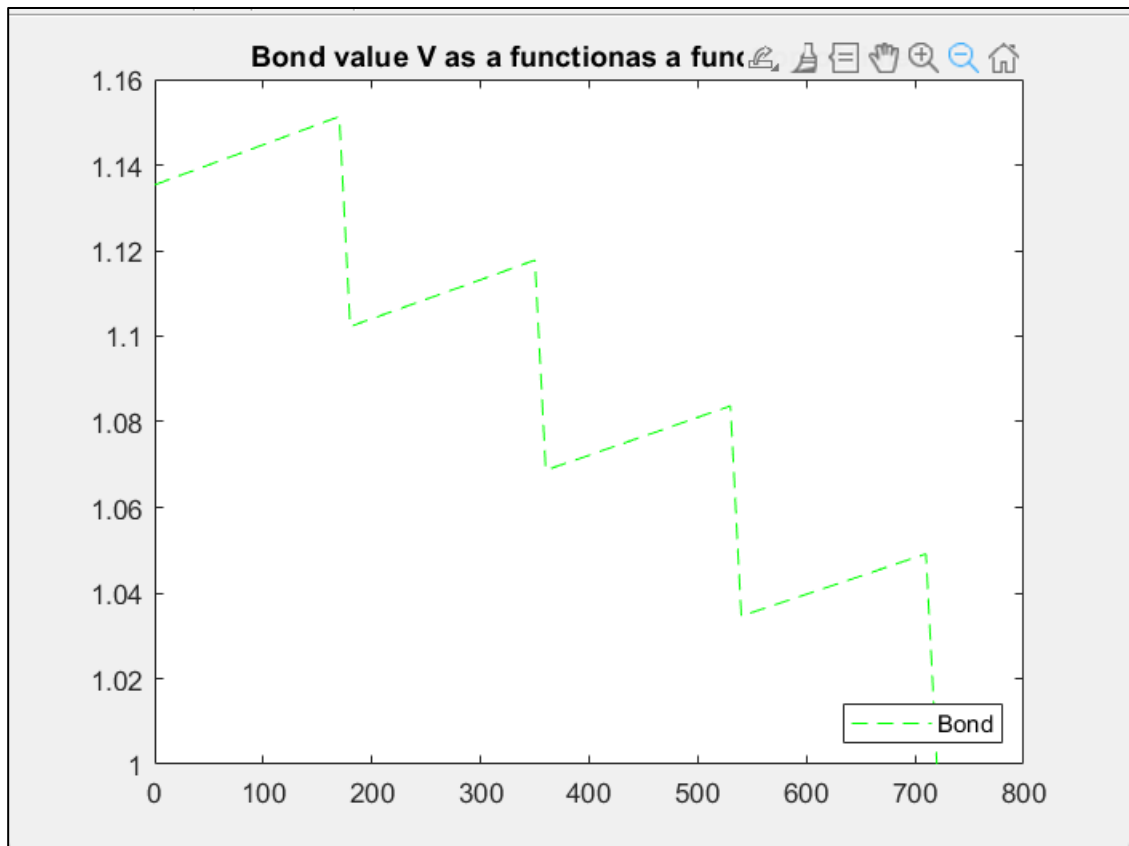
Write a function `plotBondValue8(t_{min} , t_{max} , Δt , r , N , γ)` that plots the bond value with a green dashed line, labels the axes as 'Time t ' and 'Bond value V ', writes the title 'Bond value V as a function of time t ', and has legend 'Bond' in the lower right corner.

Make the plot with $t_{min}=0$, $t_{max}=720$, and $\Delta t=10$ from the main file.

En esta pregunta se solicita generar un gráfico muy similar con diferencias netamente estéticas, lo cual se puede controlar con los argumentos añadidos a continuación:

```
main.m x plotBondValue8.m x +
1 %Question 8
2 function plotBondValue8(tmin, tmax, deltat, r, N, gamma)
3
4 % Initialize the vector of time values
5 t = linspace(tmin, tmax, (tmax - tmin) / deltat + 1);
6
7 % Calculate the bond values
8 bondValues = getBondValue5(t, N, r, gamma);
9
10 % Plot the bond values with a green dashed line
11 plot(t, bondValues, 'g--')
12
13 % Add a legend
14 title("Bond value V as a function of time t")
15 legend('Bond', Location = 'southeast')
16
17 end
18
```

Podemos ver que a la función `plot` simplemente le añadimos el argumento “g—” para generar la “Green dashed line” y para la leyenda añadimos la función `legend` dentro de la función de generación del gráfico 8, esto nos genera un plot de la siguiente forma:

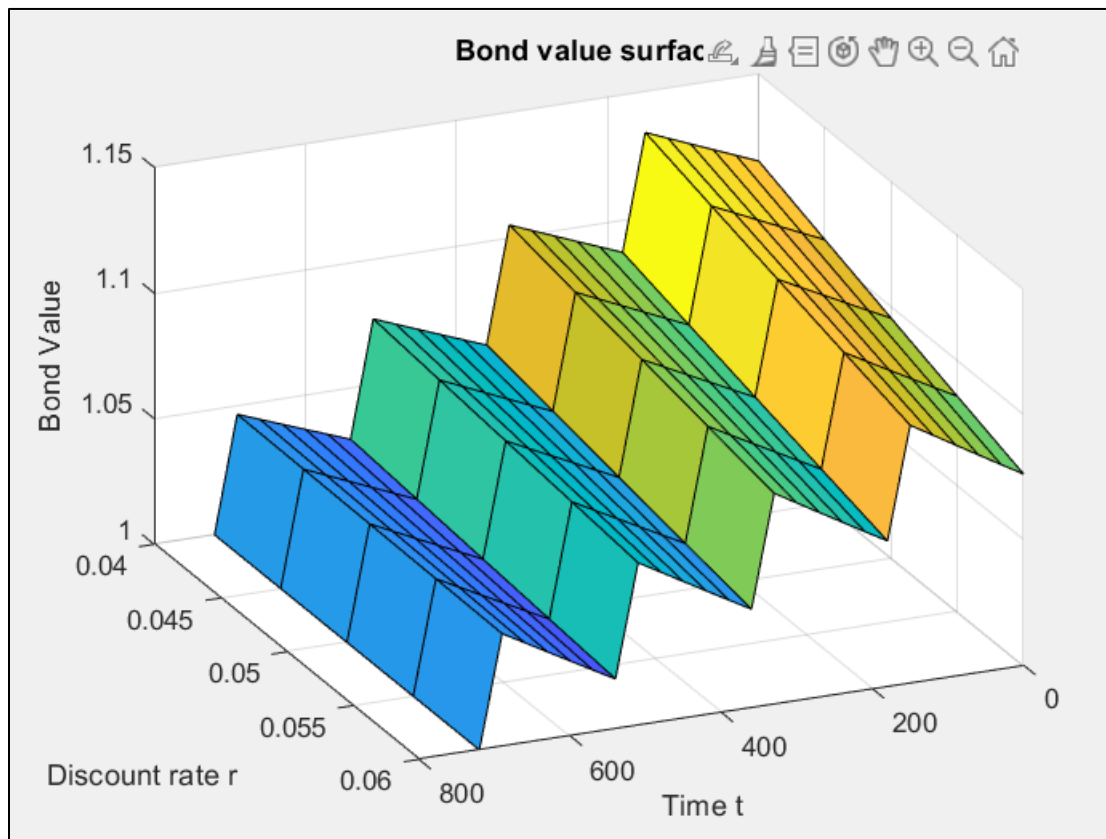


Question 9:

Write a function `surfaceBondValue9(tmin, tmax, Δt, rmin, rmax, rmin, Δr, N, γ)` that plots the surface of the bond value with time t ranging from t_{min} to t_{max} with a time step Δt and discount rate r ranging from r_{min} to r_{max} with a spacing Δr .

Make the surface with $t_{min}=0$, $t_{max}=720$, $\Delta t=30$, $r_{min}=0.04$, $r_{max}=0.06$, $\Delta r=0.005$ from the main file.

Para la pregunta 9 nos solicitan desarrollar un gráfico en forma de superficie que represente las interacciones entre el valor del bono, t y la tasa r . Para esto primero generé una matriz de ceros e inicié las variables j e $i = 0$ que usare posteriormente para los loops. En el loop de cálculo que genere se van calculando los valores del bono para cada combinación de r y t que se permite usando la función `getBondValue5` que generamos en la pregunta 5. Es importante que por la forma en la que condicionamos los loops, los valores de la variable aumentan de acuerdo con las deltas solicitadas (Δt y Δr). Posteriormente generé los vectores correspondientes a los ejes y graficamos con la función `surf` (proveniente de `Surface`). Este código genera el siguiente output.



Question 10:

Find the value of the discount rate r so that $V(r) = 1$ with the default parameters.
Write the function `getDiscountRate(t, N, γ)` that implements the previous script for general parameter values.

La ultima pregunta solicita generar una funcion que encuentre la tasa de descuento necesaria para que el valor del bono sea igual a 1 usando como inputs (t, N, gamma). Para esto, parti con igualar r a 0 y generar una variable auxiliar que tomara el valor de un bono con una tasa dada, posteriormente hay un while loop que iterará aumentando r en 0.005 hasta que el valor de la variable auxiliar sea igual a 1 (target value). Cuando esta condicion se cumpla el loop se rompera y entregara que r es el discount rate necesario. Adjunto el codigo.

```

1 %Question 10: getDiscountRate
2 function DiscountRate = getDiscountRate(t, N, gamma)
3     r=0;
4     targetvalue=1;
5     deltar = 0.005;
6     bondaux = getBondValue5(t, N, r, gamma);
7
8     while bondaux > targetvalue
9         r = r + deltar;
10        bondaux = getBondValue5(t,N,r,gamma);
11    end
12
13    DiscountRate = r;
14

```

Finalmente, al correr esta función el output tomara la siguiente forma:

Command Window

El valor del bono es de \$1.1354

The discount rate that makes V=1 is DiscountRate =0.105

fx >>