Examen Final

- Prender cámaras.
- Puede utilizar sus apuntes.
- Puede utilizar los notebooks del curso.
- La alumna o alumno que sea sorprendido solicitando ayuda a cualquier persona, sea ella compañera o no, será calificada con un 2.
- Cada pregunta vale un punto.
- La nota mínima es un 2.
- La nota se determinará con la siguiente fórmula $nota = \min\left(puntos + 2, 7\right)$
- Todas las funciones necesarias para resolver los ejercicios están ya importadas al notebook.

Librerías

In [1]:
```
from finrisk import examen_final as ef
from scipy.interpolate import interp1d
import modules.hull_white as hw
import pandas as pd
import numpy as np
import textwrap
import random
import math
```

Curva Cero Cupón

Los plazos están en días y las tasas en convención exp act/365.

In [2]:
```
df_curva = pd.read_excel('data/20201012_built_sofr_zero.xlsx')
```

In [3]:
```
df_curva.head()
```

Out[3]:

	plazo	tasa	df
0	1	0.000811	0.999998
1	7	0.000841	0.999984
2	14	0.000780	0.999970
3	21	0.000774	0.999955
4	33	0.000781	0.999929

In [4]:
```python
curva = interp1d(df_curva['plazo'], df_curva['tasa'], 'linear', fill_value='ex
trapolate')
```

Cálculo Opción

Para identificar si es una Call o una Put se usa un `enum` .

In [5]:
```python
c_p = hw.CallPut.CALL
c_p = hw.CallPut.PUT
```

Ver toda la documentación:

In [6]:
```python
print(hw.zcb_call_put.__doc__)
```

```
    Calcula el valor de una call o una put sobre un bono cero cupón en el mod
elo de HW.

    params:

    - c_p: indica si es la opción es Call o Put. Es un `enum` de tipo CallPu
t. Ejemplo, c_p = CallPut.CALL.
    - strike: es el strike de la opción. Se ingresa como número. Un strike de
l 90% se ingresa como .9.
    - r0: es la tasa corta al momento de valorizar la opción (t = 0).
    - to: instante de tiempo en que vence la opción, expresado en años.
    - tb: instante de tiempo en que vence el bono subyacente, expresado en añ
os. Debe ser tb > to.
    - zo: factor de descuento de mercado a tiempo t = 0 hasta to.
    - zb: factor de descuento de mercado a tiempo t = 0 hasta tb.
    - gamma: parámetro gamma del modelo HW.
    - sigma: parámetro sigma del modelo HW.

    return:

    -  el valor de la opción.
```

Valores `gamma` y `sigma`

In [7]:
```python
gamma = 1
sigma = .005
```

In [8]:
```python
strikes = [.98, .981, .982, .983, .984, .985, .99, .991, .992, .993]
for k in strikes:
    print(k, hw.zcb_call_put(
        hw.CallPut.CALL,
        k,
        curva(.0001),
        1,
        2,
        math.exp(-curva(365)),
        math.exp(-curva(730) * 730 / 365.0),
        gamma,
        sigma
    ))
```

```
0.98 0.019553575799963863
0.981 0.0185542779311344
0.982 0.017554980062305048
0.983 0.016555682193475696
0.984 0.015556384324646233
0.985 0.014557086455816881
0.99 0.0095605971116699
0.991 0.008561299242840548
0.992 0.007562001374011196
0.993 0.006562703505181733
```

Preguntas

In [9]:
```python
try:
    q = ef.get_questions()
except Exception as e:
    print(str(e))
```

In [10]:
```
for qq in q:
    print(f'{qq[0]}\n')
    print(textwrap.fill(f'{qq[1]}', 80))
    print('\n')
```

```
Pregunta 1:

Explique la diferencia entre valorizar un payoff g(r(T), T) por simulación de
Montecarlo utilizando el modelo de Hull-White en la medida libre de riesgo y
el
modelo de Hull-White en la medida T-forward.


Pregunta 2:

Con la curva cero cupón, interpolando linealmente en tasa, calcule la tasa
forward entre 282 y 454 días.


Pregunta 3:

Usando la curva cero cupón entregada y parámetros gamma = 1.0 y sigma = 0.5%
valorice una call a 1Y sobre un bono cero cupón a 2Y con un strike de 98.3%.
Entregue el resultado 8 decimales y suponiendo un nocional de 1.


Pregunta 4:

Calcule la TNA con ICP0 = 10,000.00 e ICP365 = 10,049.00 (365 días después).


Pregunta 5:

Considere un swap a 1Y con cupones semestrales de 1.8% (lineal). Si la tasa
cupón cero a 0.5Y es 1.7% (exp), calcule la tasa cero a 1Y.


Pregunta 6:

Ceteris paribus una opción sobre una activo con reversión a la media vale men
os
que una opción sobre un activo sin reversión a la media. Comente.


Pregunta 7:

Calcule un paso de simulación de Montecarlo para el modelo de Vasicek usando
gamma = 1.0, sigma = 0.5%, dt = 1/264, r0 obtenido de la curva cupón cero
entregada y número aleatorio N(0, 1) igual a .1629.
```

Respuestas

Pregunta 1

Pregunta 1: Explique la diferencia entre valorizar un payoff g(r(T), T) por simulación de Montecarlo utilizando el modelo de Hull-White en la medida libre de riesgo y el modelo de Hull-White en la medida T-forward.

RESPUETA:

Para la medida forward, esta se simula las tasas forward con un drift diferente quese llama theta gorro = theta - B(t,T)*sigma^2. Una vez simulada las tasas se procede a calcular el VALOR ESPERADO DEL PAYOFF como el promedio de los valores del payoff para todos los escenarios simulados y que finalmente se optiene el valor del instrumento multiplicandolo por el factor descuento.

En cambio en la medida de libre de riesgo se utiliza otro tipo de drift theta y este valor del instrumento se calcula primero descontando los valores de cada payoff simulado para luego sacar el valor esperado. Ademas este descuento se hace por cada dia y con el factor descuento del periodo pasado, digamos que estamos en el dia 264 entonces tenemos que descontar 263 para llegar a t=0. (a diferencia de la medida forward que se descuenta con la tasa del mismo periodo) 1

Pregunta 2

In [38]:
```
#Pregunta 2:
# Con la curva cero cupón, interpolando linealmente en tasa, calcule la tasa
# forward entre 282 y 454 días.
curva2 = interp1d(df_curva['plazo'], df_curva['tasa'], 'linear')

t1 = 482
t2 = 654
r1 = curva2(t1)
r2 = curva2(t2)
df1 = math.exp(-r1*t1/365)
df2 = math.exp(-r2*t2/365)
tasa_forward = (df1/df2-1)*360/(t2-t1)



print(f'tasa_forward: {tasa_forward}')
```

```
tasa_forward: 0.00041054252846l6809
```
1

Pregunta 3

In [48]:
```
#Usando la curva cero cupón entregada y parámetros gamma = 1.0 y sigma = 0.5%
#valorice una call a 1Y sobre un bono cero cupón a 2Y con un strike de 98.3%.
#Entregue el resultado 8 decimales y suponiendo un nocional de 1.

# strikes = [.98, .981, .982, .983, .984, .985, .99, .991, .992, .993]
# for k in strikes:
#     print(k, hw.zcb_call_put(
#         hw.CallPut.CALL,
#         k,
#         curva(.0001),
#         1,
#         2,
#         math.exp(-curva(365)),
#         math.exp(-curva(730) * 730 / 365.0),
#         gamma,
#         sigma
#     ))


ValorCall = hw.zcb_call_put(
        hw.CallPut.CALL,
        0.983,
        curva(.0001),
        1,
        2,
        math.exp(-curva(365)),
        math.exp(-curva(730) * 730 / 365.0),
        gamma,
        sigma
    )

print(f'Valor de la Call a 1Y :{ValorCall}')
```

```
Valor de la Call a 1Y :0.016555682193475696
```

1

Pregunta 4

In [49]:
```
#Pregunta 4:

#Calcule la TNA con ICP0 = 10,000.00 e ICP365 = 10,049.00 (365 días después).

ICP365 = 10049.00
ICP0 = 10000.00
d = 365

print(((10.049/10.000)-1)*(360/365))
print(f'TNA:{((ICP365/ICP0)-1)*(360/d): .4}')
```

```
0.0048328767123286724
TNA: 0.004833
```

.8

El redondeo es a 4 -> 0.48%. No conversa el código con el output.

Pregunta 5

In [ ]:

Pregunta 6

Ceteris paribus una opción sobre una activo con reversión a la media vale menos que una opción sobre un activo sin reversión a la media. Comente.

Respuesta:

En un modelo de activo con reversion a la media por ejemplo en el modelo de Hull-White la tasa instantanea en la medida de riesgo es:

$$dr_t = (\theta_t - \gamma^* r_t)\, dt + \sigma dX_t$$

La cual esta funcion theta, que representa el nivel de reversion a la media hace que se calibre perfectamente el valor de mercado de los bonos cupon cero. Es por esto mismo que se dice que el modelo Hull-White Calibran perfectamente la curva cupón cero de mercado a diferencia del modelo de Vasicek. Y por lo tanto la opcion del activo sin reversion a la media es mayor que el de reversion a la media.

Activo sin reversion > Activo con reversion Por lo tanto es verdadero, dado que el modelo con reversion a la media converge a valores acotados y la sin reversion no. (entonces hay mayor riesgo en activos sin reversion y por tanto estas opciones se vuelven mas caras)

Pregunta 7

In [54]:

```python
#Calcule un paso de simulación de Montecarlo para el modelo de Vasicek usando
#gamma = 1.0, sigma = 0.5%, dt = 1/264, r0 obtenido de la curva cupón cero
#entregada y número aleatorio N(0, 1) igual a .1629.
from typing import List, Tuple
import scipy.optimize as opt
import plotly.express as px
import pandas as pd
import numpy as np
import math

frmt = {'tasa': '{:.4%}', 'df': '{:.6%}'}
```

In [55]:
```python
def vasicek_path(r0: float,
                 gamma: float, r_: float,
                 sigma: float,
                 num_dias: int = 263,
                 dias_agno: int = 264,
                 pasos_dia: int = 1) -> List[Tuple[float, float]]:
    """
    Retorna un camino de simulación del modelo de Vasicek.

    params:

    - r0: tasa inicial (t = 0) de la simulación
    - Parámetros del modelo:
      - gamma: velocidad de reversión
      - r_: tasa de largo plazo
      - sigma: volatilidad
    - num_dias: número de días en la trayectoria, incluyendo el instante t = 0
    - dias_agno: número de días hábiles por año
    - pasos_dia: número de pasos de simulación en 1 día

    return:

    - Un `list` donde cada elemento es una `tuple` con los valores del tiempo
 y la tasa simulada.
    """
    dt = 1 / (dias_agno * pasos_dia)
    dt_gamma_r_ = dt * gamma * r_
    sigma_sqdt = sigma * math.sqrt(dt)
    result = [(0, r0),]
    gamma_dt = gamma * dt

    r = r0
    for i in range(1, (num_dias + 1) * pasos_dia):
        r = dt_gamma_r_ + (1 - gamma_dt) * r + sigma_sqdt * np.random.normal()
# Discretización de Euler
        result.append((i * dt, r))
    return result
```

In [59]:
```python
sim = vasicek_path(curva(.0001), gamma, 0.005, sigma, num_dias=264)

sim = pd.DataFrame(sim, columns = ['t', 'tasa'])

print(sim)
```

```
            t                  tasa
0    0.000000  0.0008062112176069023
1    0.003788           0.000219027
2    0.007576           0.000388828
3    0.011364            0.00101643
4    0.015152           0.000915147
..        ...                   ...
260  0.984848            0.00442397
261  0.988636            0.00434654
262  0.992424            0.00451577
263  0.996212            0.00481984
264  1.000000            0.00448693

[265 rows x 2 columns]
```

In [ ]:

In [ ]: