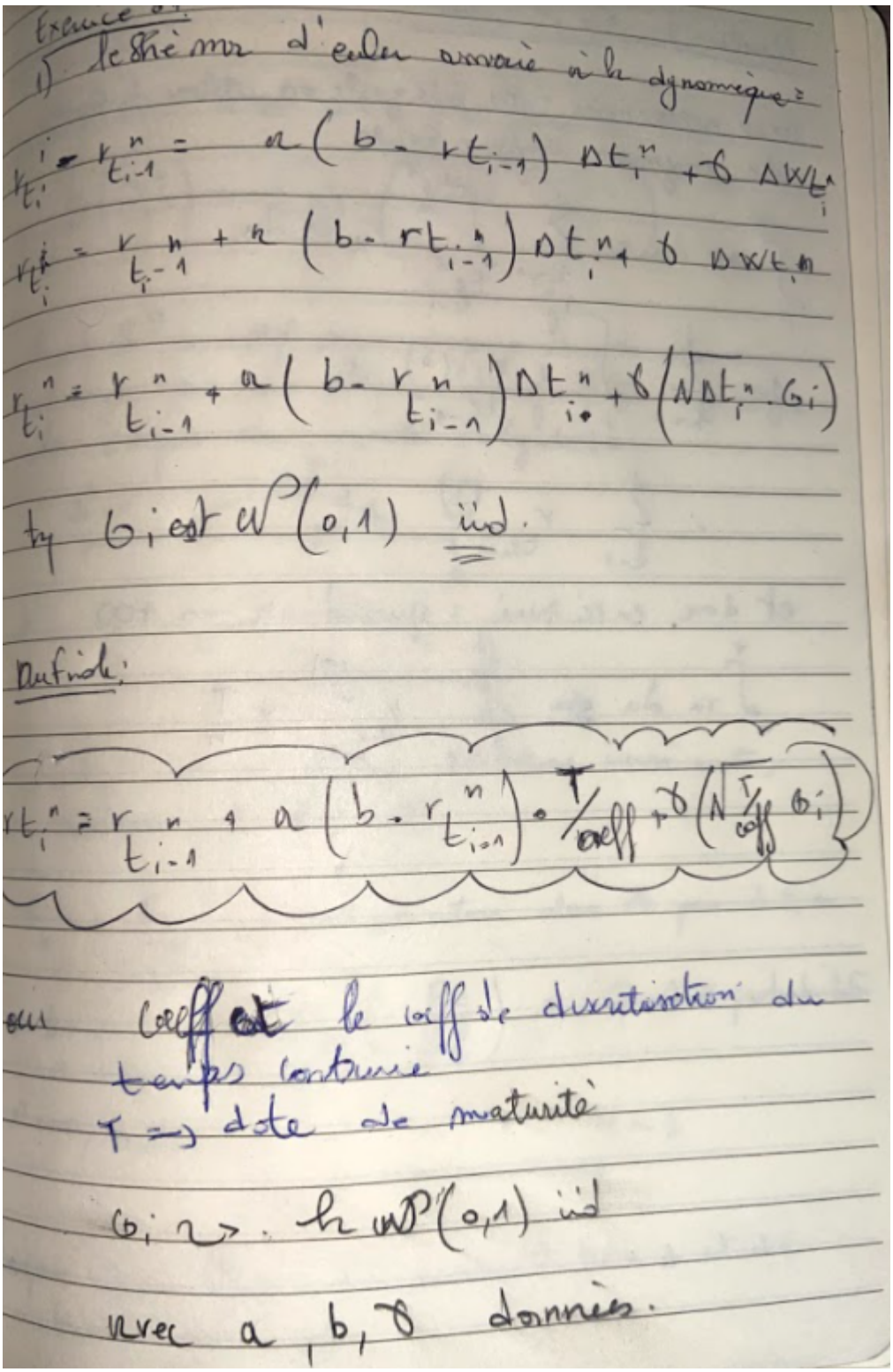
Projet examen finance des marchés

Houssem HAMROUN

Question 1 :



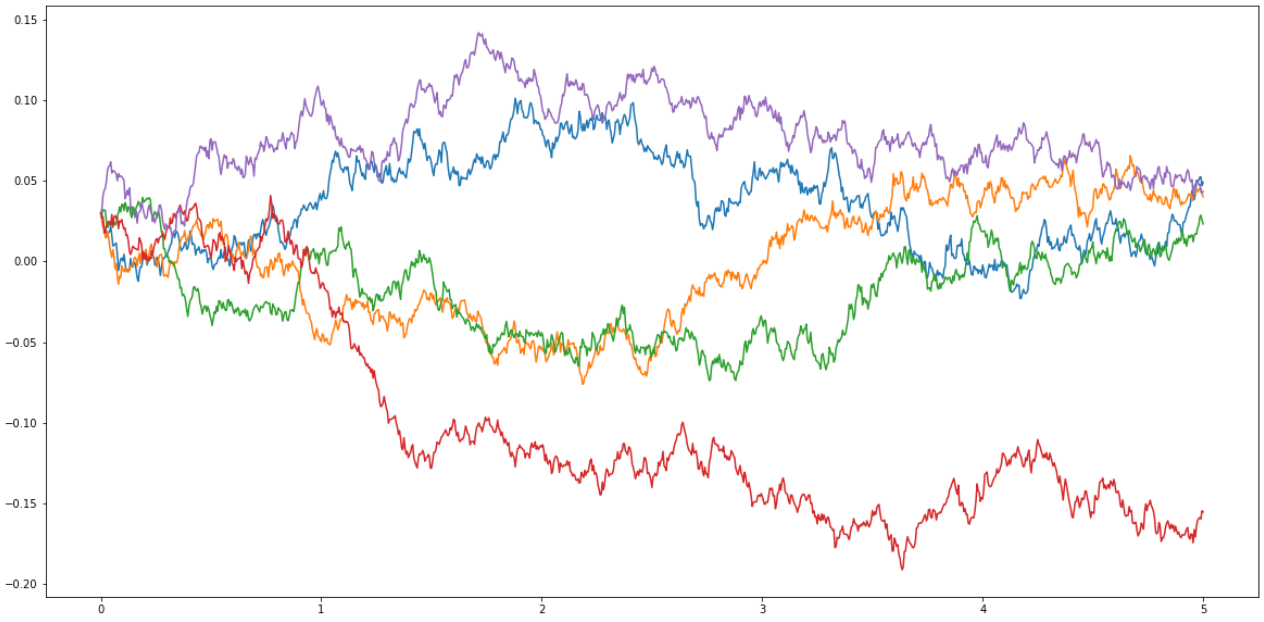
Question 2 :

In [114…

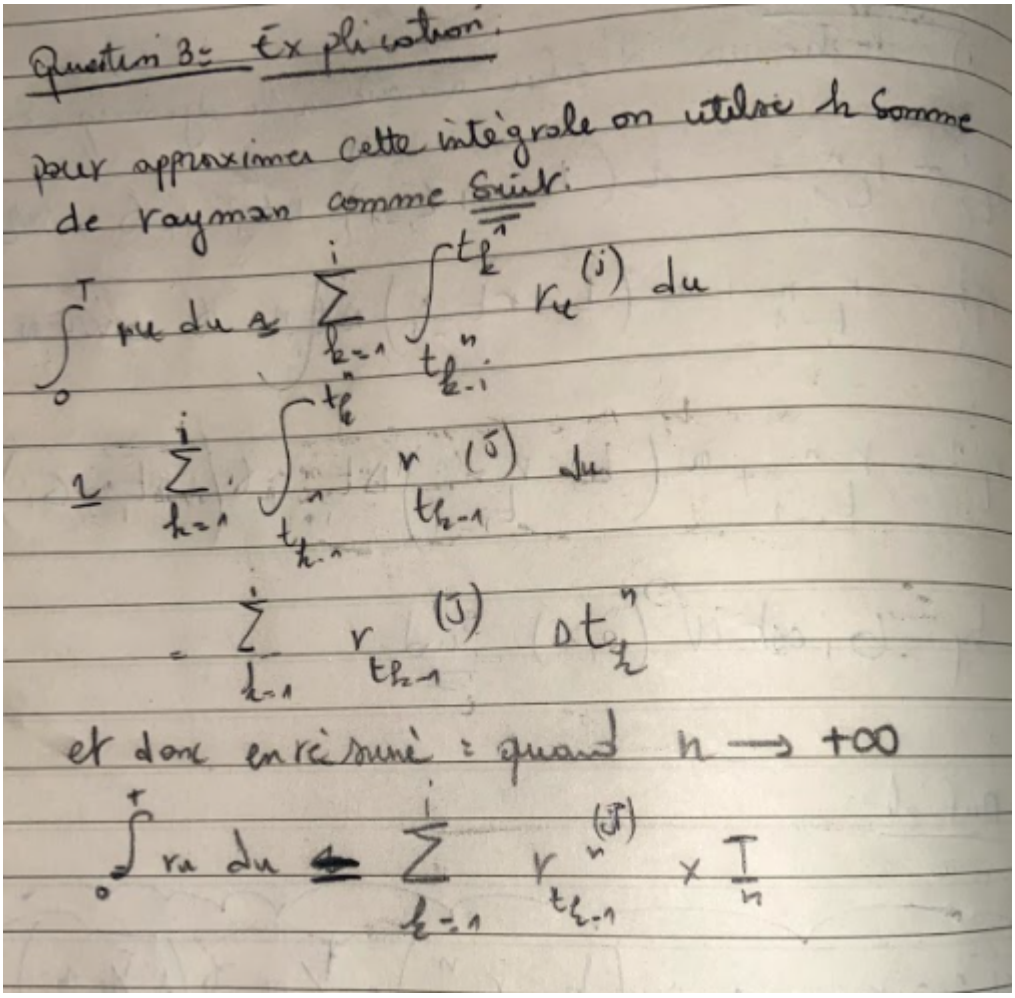
```
import numpy as np
import numpy.random as sim
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams['figure.figsize'] = (20.0, 10.0)
```

In [115…

```
n =1000
r0 =3/100
alpha = 0.1
beta =1/100
theta = 5/100
T=5
pas = T/n
N=5
r=np.ones((n+1,N))*r0

for j in range(N):
    for i in range(1,n+1):
        deltaWt = np.sqrt(pas)*sim.randn()
        r[i,j] = r[i-1,j] + alpha*(beta-r[i-1,j])*pas+ theta*deltaWt

dates=np.linspace(0,T,n+1) # n+1 dates
graph=plt.plot(dates,r)
plt.show()
```



Question 3 :



In [116…

```
n =1000
r0 =3/100
alpha = 0.1
beta =1/100
theta = 5/100
T=5
pas = T/n
N=5
r=np.ones((n+1,N))*r0
integ=np.zeros((n+1,N))*r0
for j in range(N):
    for i in range(1,n+1):
        deltaWt = np.sqrt(pas)*sim.randn()
        r[i,j] = r[i-1,j] + alpha*(beta-r[i-1,j])*pas+ theta*deltaWt
        integ[i,j] = pas* np.sum(r[0:i-1,j]) #ici on a calculé l'intégrale pour chaque

print(integ)
```

```
[[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00]
 [ 1.50000000e-04  1.50000000e-04  1.50000000e-04  1.50000000e-04
   1.50000000e-04]
 ...
 [-2.66025954e-01 -1.66335087e-01 -7.92488926e-02  2.42862488e-01
   2.53967123e-01]
 [-2.66571362e-01 -1.66422262e-01 -7.93945190e-02  2.43004285e-01
```

```
  2.53719508e-01]
[-2.67137784e-01 -1.66519203e-01 -7.95460164e-02  2.43114941e-01
  2.53486992e-01]]
```

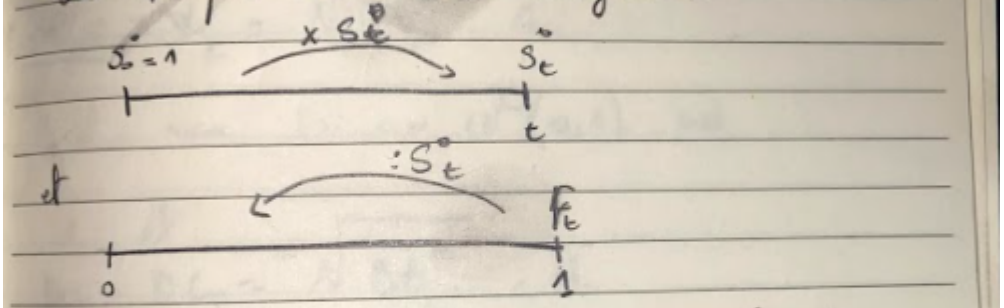
Question 4 :

Question 4:

Explication d'où vient la formule

$$D(0,T) = E_Q\left(e^{-\int_0^T r_u\,du}\right)$$

* Soit S^0 un actif sans risque dans le sens où il n'y a pas de risque de faillite mais il y'a le risque du marché obligataire

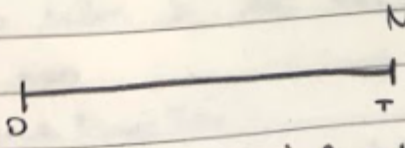


et

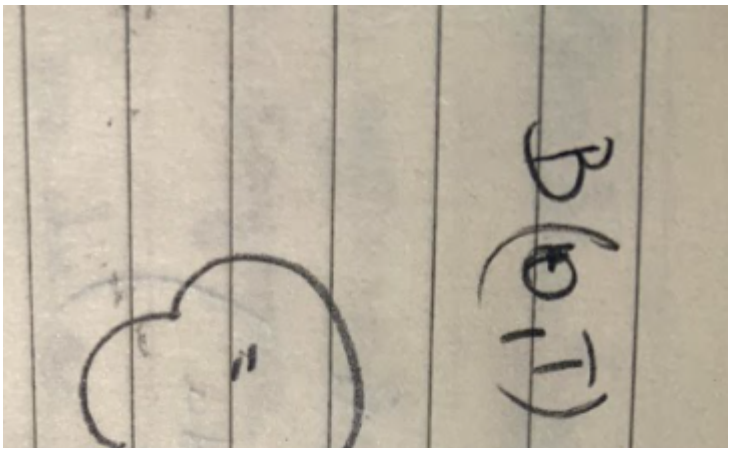
si F_t est un flux aléatoire versé en t, sa valeur actualisée est

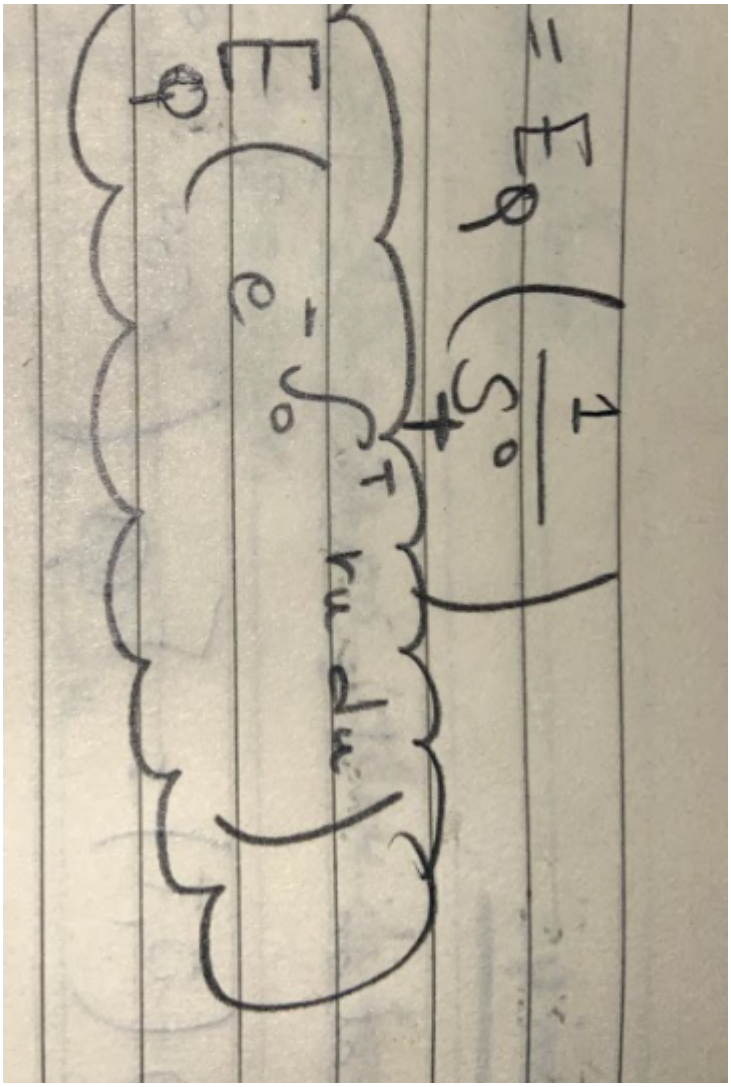
$\tilde{f}_t = \frac{F_t}{S_t^0}$ qui est aléatoire donc le prix de f_t en $t=0$ est: $E_Q\left(\frac{F_t}{S_t^0}\right)$ où Q est la probabilité de risque neutre

Ainsi, pour:



le prix d'un ZC de nominal N = 1 et de maturité T est =

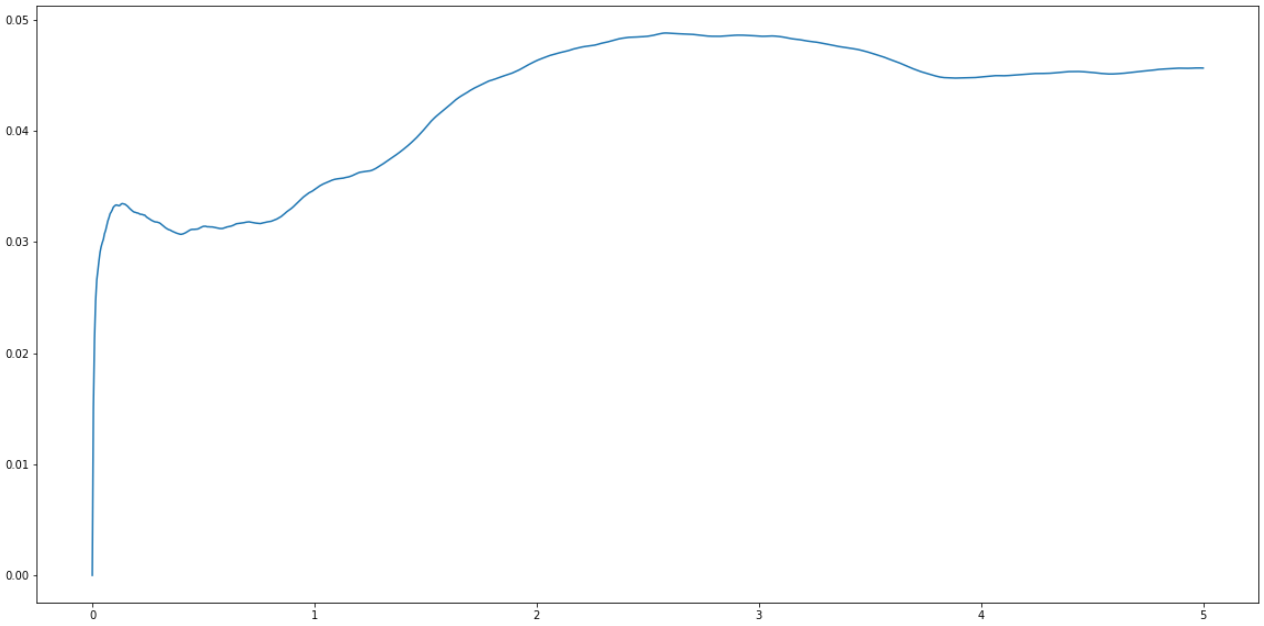




In [117…

```
n =1000
r0 =3/100
alpha = 0.1
beta =1/100
theta = 5/100
T=5
pas = T/n
N=5
r=np.ones((n+1,N))*r0
integ=np.zeros((n+1,N))*r0
Bt=[1] # le vecteur qui stock les Bt(i,n)
R=[]
CoeffActualisation= np.ones((n+1,N)) # Cette variable sert a stocké l'esperence de l'ex
for j in range(N):
    for i in range(1,n+1):
        deltaWt = np.sqrt(pas)*sim.randn()
        r[i,j] = r[i-1,j] + alpha*(beta-r[i-1,j])*pas+ theta*deltaWt
        integ[i,j] = pas* np.sum(r[0:i-1,j]) #ici on a calculé l'intégrale pour chaque
        CoeffActualisation[i,j] = np.exp(-integ[i,j])

 # 2eme etape sommer les Bt(i,n)
for i in range(1,n+1):
    moyenneBt=np.mean(CoeffActualisation[i,:])
    Bt.append(moyenneBt)
    R.append(moyenneBt**(-1/(pas*i))-1)
```

```
R.append(R[-1])
# le vecteur ou on va stocké les valeur du taux Zc R(t)
#on a mis une valeur a cause du problème de n+1 car la longeur de R est n et le rajout

dates=np.linspace(0,T,n+1) # n+1 dates
graph=plt.plot(dates,R)
plt.show()
```



In [118…

```
print("les prix du B0,T sont : ")
print(Bt)
```

```
les prix du B0,T sont :
[1, 1.0, 0.9998500112494375, 0.9996836412954458, 0.9995111987501784, 0.9993440753710885,
0.9991868665838941, 0.9990225990150771, 0.9988527749673641, 0.9986878722528838, 0.998525
0781807806, 0.9983616242289471, 0.9981835166857371, 0.9980135209750587, 0.9978307468662
5, 0.9976462341383392, 0.9974672654964989, 0.9972804772041612, 0.9971069776273389, 0.996
9281510010063, 0.9967427806592302, 0.9965716380951763, 0.9963992780745425, 0.99623529890
85475, 0.9960757622833187, 0.995916464557645, 0.9957480384746509, 0.9955695317444688, 0.
9954012531422327, 0.9952403709943128, 0.9950805512736685, 0.9949239735878536, 0.99477423
92730516, 0.9946230011627659, 0.9944801736430662, 0.9943349019832135, 0.994188960402082
6, 0.9940419368352362, 0.9938977649610736, 0.993745315308004, 0.993589907437299, 0.99343
83330398762, 0.9932817765856375, 0.9931316603848716, 0.992985186719398, 0.99282780173786
11, 0.9926719939532319, 0.9925249383087345, 0.9923657975002941, 0.9922292861196599, 0.99
20969940922406, 0.9919545807051765, 0.9918117719418195, 0.9916736625944148, 0.9915356671
676621, 0.9913940848542862, 0.991250341545266, 0.9911068781051917, 0.9909581829092076,
0.9908042009202355, 0.9906572958382387, 0.990511795046108, 0.9903715361149775, 0.9902419
759772766, 0.9901120038152074, 0.9899821035117421, 0.9898565840431669, 0.989732367665978
3, 0.9896010786562123, 0.9894687015990714, 0.9893322139893733, 0.9891900386755765, 0.989
0593682046613, 0.9889258262191639, 0.9887909815855433, 0.9886605613802708, 0.98852192549
57383, 0.9883869768905209, 0.9882487022887417, 0.9881072607718115, 0.9879718705581281,
0.9878251529615166, 0.9876669103907731, 0.9875072275977217, 0.9873407087152376, 0.987170
2034643723, 0.9869936574578455, 0.9868126808968792, 0.9866373420581475, 0.98646585479285
33, 0.9862998716884673, 0.9861409135693819, 0.9859897534087875, 0.9858362255617694, 0.98
56817557224767, 0.9855306478942613, 0.9853686120878222, 0.9851937359782758, 0.9850137471
521702, 0.9848383705974353, 0.9846660363991301, 0.9844965517239302, 0.9843437861028065,
0.9841886760809924, 0.9840515564169975, 0.9839087325211174, 0.983753287895837, 0.9836096
191392262, 0.9834584028979672, 0.9833071063637859, 0.9831658797543714, 0.983027476978928
2, 0.9828848782743845, 0.9827480591486729, 0.9825987645081602, 0.9824594577588954, 0.982
3242475060405, 0.9821757551293976, 0.9820218431053547, 0.9818567718655666, 0.98168178595
02077, 0.9815171034950311, 0.9813396340642454, 0.9811761077242858, 0.9810065940419538,
```

```
0.9808440445688069, 0.9806832879274727, 0.9805109762576603, 0.9803335391530899, 0.980149
0479003434, 0.9799656170958937, 0.9797909908538562, 0.9796295302131484, 0.97946769453049
85, 0.9793108130357091, 0.9791497739869026, 0.9789838536427432, 0.9788243600942955, 0.97
86573680905342, 0.9784839956315006, 0.9783199487441323, 0.9781536668275057, 0.9779983821
388409, 0.9778562763301046, 0.9777210210838867, 0.9775778035163281, 0.9774390737873292,
0.9772989307245901, 0.9771565425183061, 0.9770132148220858, 0.976871958714223, 0.9767286
996970658, 0.9765790877487113, 0.9764143215125989, 0.976240317799592, 0.976072786563528
9, 0.9759073118439818, 0.9757383615378549, 0.9755633404879699, 0.9754000053833256, 0.975
2327467272088, 0.9750732842977092, 0.9749014381616702, 0.9747225909000242, 0.97453470127
14166, 0.9743503699401144, 0.974167053843115, 0.9739816365945089, 0.9737846968874436, 0.
9735936719388647, 0.9733894317514936, 0.9731815353268622, 0.9729610068325971, 0.97274012
88829256, 0.9725138758424707, 0.972281989829883, 0.9720549860789701, 0.9718342763477699,
0.9716205937770173, 0.9714057621729062, 0.9711847813075913, 0.9709582040872675, 0.970724
5304567355, 0.9704853835067049, 0.9702431187338008, 0.9699990110664307, 0.96975065975172
91, 0.9695042900379598, 0.9692492958568719, 0.9690051007494265, 0.9687585946403126, 0.96
85194045155926, 0.9682885457842332, 0.9680552576579812, 0.967820690203687, 0.96759083902
4659, 0.967364456659286, 0.9671544832793948, 0.9669447715368717, 0.966727681087203, 0.96
64980636670173, 0.9662686017638403, 0.9660329135455574, 0.9658033522897147, 0.9655727236
751724, 0.965343782721277, 0.9651079514490393, 0.9648876593651898, 0.9646666603052505,
0.9644518440146598, 0.9642473779864116, 0.9640344417541067, 0.9638228149322394, 0.9636006
436398837, 0.9633889917407498, 0.9631792919647791, 0.9629696804434105, 0.96277002946200
4, 0.9625679944272386, 0.9623682827713147, 0.9621778891208406, 0.9619901393132851, 0.961
8062376128954, 0.9616259766290396, 0.9614462198985461, 0.9612627952711355, 0.96108171468
35174, 0.9608959392132753, 0.9606955554682154, 0.9605044331766569, 0.9603147341573652,
0.9601203539105019, 0.9599315243986037, 0.9597266509069884, 0.959515457949925, 0.9593072
505753317, 0.9590900789166856, 0.9588696934691274, 0.9586462542228903, 0.958423462239427
5, 0.958201782726498, 0.9579882628625915, 0.9577812431843148, 0.9575918323129837, 0.9574
05852504318, 0.9572150273590522, 0.9570160922822464, 0.9568270505728019, 0.9566438294607
348, 0.9564650936299884, 0.9562857577916377, 0.9560978862054401, 0.9558939920405202, 0.9
55684411613495, 0.9554584623037397, 0.9552321510819016, 0.954996807187403, 0.95475864025
68671, 0.9545067645887038, 0.9542545816333001, 0.954007399602234, 0.9537649232673135, 0.
9535120497516386, 0.9532641418933465, 0.9530081299283243, 0.9527456642257037, 0.95248868
88583758, 0.9522279057479632, 0.9519801962623611, 0.9517212582094743, 0.951459799731351
4, 0.951203728421824, 0.9509443737364988, 0.9506786945762492, 0.9504046773928936, 0.9501
405072012504, 0.9498748182559928, 0.9495985021396359, 0.9493278550493294, 0.949055067744
9313, 0.9487746265457891, 0.9484948869588958, 0.9482152052425115, 0.9479372017398722, 0.
9476457808540918, 0.9473554754255046, 0.9470646692290778, 0.946767796692963, 0.946467093
7526127, 0.9461646024006045, 0.9458636221411586, 0.9455566306193408, 0.9452493134598413,
0.9449304424419939, 0.9446052933848739, 0.9442766100381457, 0.9439536879049044, 0.943636
1058644941, 0.9433006637798844, 0.942955939558377, 0.9426086520464716, 0.942265116261964
6, 0.9419164035001663, 0.9415545142196509, 0.9412100467485824, 0.9408591861784625, 0.940
528826226274, 0.9401988209509066, 0.9398764718220141, 0.9395501760422578, 0.939228248459
1232, 0.9389130246670092, 0.93860653457174, 0.9382975632643129, 0.9379862166219748, 0.93
76743486030519, 0.937368440601835, 0.9370549296301508, 0.9367418443383182, 0.93643325785
09631, 0.9361218972141238, 0.935805466410868, 0.9354908565860288, 0.9351822102452182, 0.
9348659177607546, 0.934539871188006, 0.934213708485844, 0.9338939822684609, 0.9335700913
49298, 0.9332624845656949, 0.9329552493557657, 0.9326567223444651, 0.9323577327041035,
0.9320683459559588, 0.9317857772726272, 0.9315047906304212, 0.9312207987242127, 0.930926
4633231248, 0.930633565309053, 0.930347267366032, 0.9300411167274116, 0.929743900861893
2, 0.9294552059730797, 0.9291627936557376, 0.9288777624048775, 0.9286038850107238, 0.928
3285465879274, 0.9280522430443296, 0.9277738842394989, 0.9274934541956515, 0.92721773483
07752, 0.9269366595677339, 0.9266612586424527, 0.9263867569746662, 0.9261129778554187,
0.9258349849156587, 0.9255555928930642, 0.9252826498396572, 0.9250109571169853, 0.924754
8592665392, 0.9245074562611958, 0.9242594078494548, 0.9240098886604026, 0.92375650638314
85, 0.9234933014040078, 0.9232369001298049, 0.9229768888821015, 0.9227124109204791, 0.92
24480523005323, 0.9221821341033808, 0.9219210653067865, 0.9216664975950302, 0.9214091031
371323, 0.9211435711291026, 0.9208872756290644, 0.9206369298467066, 0.9203826832538372,
0.9201310047643447, 0.9198661479626811, 0.9195969399801847, 0.9193208378422636, 0.919043
3403911106, 0.9187645538597042, 0.918487487564571, 0.9182032242507929, 0.917915673415599
2, 0.9176286019341502, 0.9173271040365943, 0.9170250806707478, 0.9167131503245122, 0.916
4016760647368, 0.916103224973724, 0.9158137132932591, 0.915520071019686, 0.9152222376919
65, 0.9149253745723657, 0.9146337975931489, 0.9143385501512343, 0.9140455772102142, 0.91
37508717330179, 0.9134591184193217, 0.9131709353254506, 0.9128866059705609, 0.9126146227
64008, 0.9123346340932852, 0.9120500801816316, 0.9117780392762752, 0.9115015476391195,
0.9112210170377131, 0.9109410019255948, 0.9106605392667616, 0.9103873942722466, 0.910117
```

```
7298804861, 0.909852824558469, 0.9095925634058991, 0.9093335486743825, 0.909085166384802
1, 0.9088280469145769, 0.9085658832340548, 0.9083051213769913, 0.908040552856386, 0.9077
798666593798, 0.9075246159597203, 0.9072683347333372, 0.907008137012053, 0.9067625561633
456, 0.9065091182573353, 0.9062617190324888, 0.9060117932411453, 0.9057587528553446, 0.9
054946231526699, 0.905220899714353, 0.9049540385303505, 0.904681215584185, 0.90441550618
6961, 0.9041594263111392, 0.9039034036036483, 0.9036450979464148, 0.903398966033674, 0.9
031466246643584, 0.902891829142544, 0.9026343692061637, 0.9023823773914111, 0.9021302935
219829, 0.9018983428858947, 0.9016639344228853, 0.9014332405952198, 0.9011941410649958,
0.9009604440724767, 0.9007286676687899, 0.9005024756517969, 0.9002669147258071, 0.900023
5194445084, 0.8997664756090245, 0.8995153420955486, 0.8992575315410363, 0.89899303346305
45, 0.8987175767861924, 0.8984550612241401, 0.8981883240290305, 0.8979301146850551, 0.89
76726614367354, 0.8974209169954674, 0.8971700048616281, 0.8969225408611556, 0.8966647096
727961, 0.8963982183632684, 0.8961338484906278, 0.8958653479790583, 0.8955964596007953,
0.8953360025678638, 0.8950693397646605, 0.8948033705664953, 0.8945350097972005, 0.894273
1473596852, 0.8940126028530724, 0.8937570941815045, 0.8935103618063851, 0.89326728018639
77, 0.893024351181934, 0.892776857106764, 0.8925342275076963, 0.892297248434027, 0.89206
24741957084, 0.8918301627382592, 0.8916121282479438, 0.8914010868037938, 0.8911873798523
352, 0.8909660326464195, 0.8907407760943732, 0.8905226445497961, 0.8902949299652961, 0.8
900635611059039, 0.8898408397105021, 0.8896234765453489, 0.8894093149491702, 0.889188285
368858, 0.8889587606484692, 0.8887345391408665, 0.88851132561062, 0.8882838444379836, 0.
8880448446638581, 0.8877999149257153, 0.8875591537185299, 0.8873141777800122, 0.88707068
4398994, 0.8868145538323574, 0.8865484251050706, 0.8862868474788332, 0.8860271132064824,
0.8857651071493929, 0.8855056093298863, 0.8852460717975216, 0.8849880575548363, 0.884741
5488590983, 0.8844999330609994, 0.8842688251696199, 0.8840488919472651, 0.88384169744639
76, 0.883645682607369, 0.8834554527229663, 0.8832612973267, 0.8830619837600903, 0.882856
85801798, 0.8826503795985399, 0.8824519855581565, 0.8822571959920575, 0.882064351275441
9, 0.8818666110371147, 0.8816720507423164, 0.8814706615864057, 0.8812695474338629, 0.881
0735090242329, 0.8808792142283239, 0.8806732020905421, 0.8804796825781412, 0.88028085645
7917, 0.8800791804023174, 0.8798720673805894, 0.879670762655611, 0.8794648576958126, 0.8
792726549611208, 0.8790847433003371, 0.8789029772858576, 0.8787183988036954, 0.878536489
5652981, 0.8783504990704799, 0.8781615592294818, 0.8779811120028954, 0.8777962725532127,
0.8776208555346974, 0.8774435533779034, 0.8772623425383801, 0.8770826583371548, 0.876897
6176743708, 0.8767143575375711, 0.8765225733697788, 0.8763281286658724, 0.87612578215574
08, 0.8759238164887609, 0.8757188804756628, 0.8755140590902144, 0.8753073542794585, 0.87
50995869856816, 0.8748960195207115, 0.8746895624262245, 0.8744813810127516, 0.8742596578
200382, 0.8740385939899762, 0.8738142276924459, 0.8735865276909923, 0.8733688238953222,
0.8731533821725549, 0.8729308704220763, 0.8727105885613724, 0.8724867328580878, 0.872254
9361777882, 0.8720239410479017, 0.8717903258605798, 0.8715715512965518, 0.87135011067556
15, 0.8711345802531459, 0.870921713877064, 0.8707136560121084, 0.8705075017074831, 0.870
2981418534652, 0.8700987316104556, 0.869900185646121, 0.8697039567315923, 0.869510981836
4569, 0.8693183354588969, 0.8691267122743547, 0.868934070927447, 0.868733143903372, 0.86
85392438017697, 0.8683393716434091, 0.8681555870907992, 0.8679645308800561, 0.8677769482
859853, 0.8675863188155455, 0.8674084781164645, 0.8672364201678251, 0.8670487445374102,
0.8668509550566872, 0.8666411106143137, 0.8664363159618114, 0.8662249171608047, 0.866011
7969304071, 0.865791841381905, 0.8655730047712703, 0.865356874918309, 0.865139329028362
3, 0.8649266305401853, 0.8647181044356487, 0.8645197038276692, 0.8643318576712105, 0.864
143385235636, 0.8639602973527939, 0.8637812460254726, 0.8636067456116313, 0.863434161832
1123, 0.8632586241595913, 0.8630891336148181, 0.8629290516006416, 0.8627772390407928, 0.
862624422943463, 0.8624714937272518, 0.8623285405661425, 0.8621739229791029, 0.862014732
5240515, 0.8618589233432333, 0.8616939232980645, 0.8615289630780504, 0.861357928216461,
0.8611896810249846, 0.8610105804087806, 0.8608335933110757, 0.8606596530481857, 0.860492
2573453451, 0.8603389668285825, 0.860181522590055, 0.8600275375224756, 0.859868985062783
6, 0.8597075261313611, 0.8595536242826164, 0.8593975069544699, 0.8592368043837635, 0.859
073444329389, 0.8589064449740196, 0.8587417296481077, 0.8585660317601318, 0.858390245827
759, 0.8582166432570301, 0.8580555890053294, 0.8578952973426484, 0.8577427043729353, 0.8
575938619595924, 0.8574478670057072, 0.8572924741318, 0.8571413297164844, 0.856995485180
5381, 0.8568459442413635, 0.8566927515233411, 0.8565504149118739, 0.8564051191398668, 0.
8562545355280022, 0.8561003539464783, 0.8559519230429654, 0.8558104209670233, 0.85567528
09673055, 0.8555355317382937, 0.8553978470796728, 0.8552593648027429, 0.855105479206435
6, 0.8549458121113374, 0.854785474864609, 0.8546345467760494, 0.8544746327485487, 0.8543
146427135466, 0.8541576662154465, 0.8540035888610003, 0.8538497253155709, 0.853693553776
2922, 0.8535432568064522, 0.8533827707700189, 0.8532288686143069, 0.8530722812895892, 0.
8529148627755511, 0.8527615098813097, 0.8526141219332019, 0.8524753474699829, 0.85233730
60036503, 0.8522059272288901, 0.852080761507936, 0.8519548977680979, 0.851829393413485,
0.851709312273924, 0.8515971246457614, 0.851474945262584, 0.8513534265302518, 0.85124134
```

```
89299963, 0.8511304825716935, 0.8510217832853121, 0.8509176181504744, 0.850806653680668,
0.8507013682055623, 0.8506015460221995, 0.8505049503071757, 0.8504105676381302, 0.850310
2304279608, 0.8501992448453469, 0.8500915282172862, 0.8499842733563719, 0.84988705117268
7, 0.849791377580862, 0.8497025780981693, 0.8496243122264293, 0.849545125290183, 0.84947
16842742436, 0.8493848532244639, 0.8492971286124351, 0.8492090937592149, 0.8491245754046
016, 0.8490391897491048, 0.8489463153400685, 0.8488623545594198, 0.8487770741317021, 0.8
487041714983052, 0.8486358666189606, 0.848572625103688, 0.8485082326336573, 0.8484456716
697757, 0.8483844881544689, 0.8483285138614945, 0.8482653862675484, 0.8482038046916298,
0.8481433589978227, 0.8480774142101757, 0.8480158762928747, 0.8479508785389742, 0.847876
9483505315, 0.8478025606538535, 0.8477291716105537, 0.8476581809852426, 0.84758529309843
76, 0.8475085349543608, 0.8474297543945276, 0.8473404707051962, 0.8472389437986451, 0.84
71369260514076, 0.8470400250986401, 0.8469488463367579, 0.8468566526355261, 0.8467600267
184547, 0.8466631313454831, 0.8465661732392405, 0.8464698555088643, 0.8463776341450121,
0.8462810020893619, 0.846181093554763, 0.8460806041966235, 0.8459729194252065, 0.8458663
160595894, 0.8457575218916498, 0.8456394200055118, 0.8455039417115804, 0.845359195537664
8, 0.8452196735866412, 0.8450657999608436, 0.8448963296259837, 0.8447296603006341, 0.844
5706348906674, 0.8444105942236042, 0.8442491295595023, 0.8440803447216287, 0.84390506497
64972, 0.843721089177332, 0.8435345753405861, 0.8433475618515688, 0.8431554185058564, 0.
8429646388526258, 0.8427707959331018, 0.8425703530524444, 0.8423710181953652, 0.84217880
981665, 0.841993982407973, 0.841806772456254, 0.8416172053353337, 0.841429960594599, 0.8
412459812878266, 0.841049928259881, 0.8408477029498622, 0.8406540882509799, 0.8404601587
60641, 0.8402717716117796, 0.8400730610511481, 0.8398679162442617, 0.8396530211077181,
0.8394393308758655, 0.8392251037663158, 0.8390013990021433, 0.8387886523305266, 0.838568
5338285738, 0.8383500731205384, 0.838131697096216, 0.8379111039632219, 0.83769854624307
4, 0.8374704524041825, 0.8372551609088046, 0.8370391684978788, 0.8368309700184474, 0.836
6248998161396, 0.8364182310219462, 0.8362124978910034, 0.8360087326981706, 0.83581933641
03862, 0.835634686568239, 0.8354526852283923, 0.8352745036652497, 0.8351041750995586, 0.
83492586664161, 0.8347495386548353, 0.8345646367699026, 0.8343728755165307, 0.8341766144
050531, 0.8339742579594296, 0.8337678716547068, 0.8335584172748896, 0.8333426195448836,
0.8331332572901096, 0.8329324576385069, 0.8327216559201105, 0.8325115721435312, 0.832298
052031536, 0.8320862870460675, 0.8318696665130997, 0.8316517113758133, 0.831436759841300
4, 0.8312256123888249, 0.831016303733765, 0.8308140806588054, 0.8306058974514009, 0.8304
038376832665, 0.8301982619712514, 0.8299923648100549, 0.8297843763142033, 0.829575055378
3959, 0.8293689167687273, 0.8291598796269424, 0.8289587471275357, 0.8287634646966524, 0.
8285727324923366, 0.8283883453567755, 0.828210962148289, 0.8280280178947905, 0.827842093
1297384, 0.8276541326848393, 0.827465878091488, 0.8272786916102023, 0.8270990670280396,
0.8269181147743396, 0.8267179139501983, 0.8265110570055336, 0.8263078197096119, 0.826106
9770203188, 0.8259008331089721, 0.8256889231299904, 0.8254719120983166, 0.82525654958254
93, 0.8250419912002032, 0.8248286462741745, 0.8246137987528733, 0.8243997029777457, 0.82
41788023229759, 0.823952274501837, 0.8237287921670194, 0.8235187990679871, 0.82331018672
46724, 0.8230990169970808, 0.8228926682346867, 0.8226967443545966, 0.822507120782304, 0.
8223160879513483, 0.8221332649791007, 0.8219477843266756, 0.8217666635238089, 0.821573954
3220313, 0.821388895534953, 0.8212001197870 53, 0.8210213699279107, 0.8208512296119375,
0.8206856777327196, 0.820526090057174, 0.8203642471705784, 0.8202044401710828, 0.8200493
839634285, 0.8198899065981496, 0.819736842599761, 0.8195836877209896, 0.819440656822488
7, 0.8192845404842704, 0.8191350010869151, 0.8189893582088832, 0.8188433729401658, 0.818
7065961639872, 0.8185581445416261, 0.8184199611213568, 0.8182831404171527, 0.81813851436
1147, 0.8179973035064598, 0.8178476314608407, 0.8177039346870222, 0.8175605970558667, 0.
8174065061389164, 0.8172457284822332, 0.8170799922165614, 0.816913179738021, 0.816737162
1991404, 0.8165546098128915, 0.8163658571917475, 0.8161852175257142, 0.8159931003941641,
0.8157894506781325, 0.8155820847672569, 0.8153807900240807, 0.8151807313250229, 0.814970
7896023008, 0.8147737641192009, 0.8145674678227932, 0.8143583864876478, 0.81414367627320
84, 0.8139257754574798, 0.8137095375569713, 0.8134883884511627, 0.813278164085322, 0.813
0549772056772, 0.8128355769396745, 0.8126146476770622, 0.812393594975427, 0.812179277159
027, 0.8119677438742372, 0.8117526536124717, 0.8115280285370894, 0.8112993958381095, 0.8
110725033457026, 0.810842125457049, 0.8106140801766483, 0.8103908770723898, 0.8101788003
941396, 0.8099643265139326, 0.8097512440592431, 0.80952872412685, 0.8093095216601798, 0.
809097214856793, 0.8088820951063029, 0.8086628919434107, 0.808443252156667, 0.8082226425
051868, 0.8079984895513677, 0.8077616610945754, 0.8075284980393379, 0.8073077111856849,
0.807095311134494, 0.8068869026416756, 0.8066788557521232, 0.8064759854094714, 0.8062741
651514148, 0.8060721329552999, 0.8058679477468816, 0.8056524604792813, 0.805438305062304
3, 0.8052260274156063, 0.8050147749756118, 0.8048093233609036, 0.8046092257962234, 0.804
4152324280395, 0.8042270915620742, 0.8040356421415249, 0.8038409545195897, 0.80366204529
20639, 0.8034923827622299, 0.8033296878761298, 0.8031536645819497, 0.8029801345167196,
0.8028122307996215, 0.8026304770521516, 0.8024469834489644, 0.8022679678752335, 0.802090
```

```
1298955753, 0.8019107850799754, 0.801719104932511, 0.8015152087438748, 0.801305172099901
8, 0.8011135063364015, 0.8009213121243729, 0.8007293537264685, 0.8005425134939064, 0.800
3583829463764, 0.8001906768772423, 0.800026164312343, 0.7998630742835893]
```

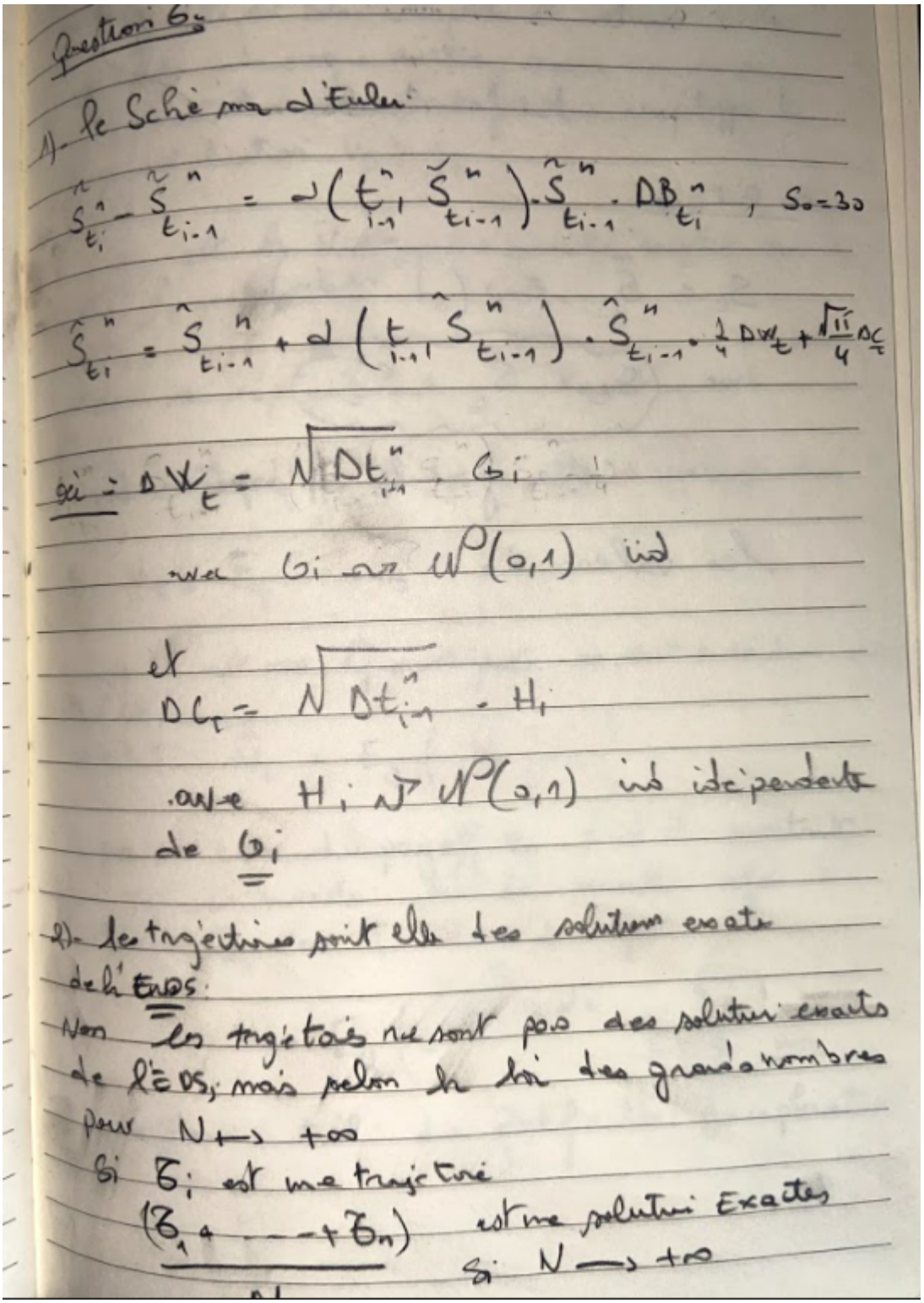
Question 5 :

Les marchés obligataire et l'actif S ne sont pas corrélés car la volatilitée de l'actif S dans l'EDS qui est : $B_t = \frac{1}{4}W_t + \frac{\sqrt{15}}{4}C_t$ où Ct est un mouvement brownien standard independant. On sait que la mouvement brownien du marché et le Wt qui est utilisé dans l'EDS du taux instantané rt. or la volatilité de l'actif S et constitué aussi du movement Ct qui est indépendant du mouvement du marché wt.

donc l'actif S et le marché ne sont pas corrélés.

Question 6 :

Donnez le schéma d'euler :



Question 6 :

1) le Schéma d'Euler :

$$\tilde{S}^n_{t_i} - \tilde{S}^n_{t_{i-1}} = \alpha\left(t_{i-1}, \tilde{S}^n_{t_{i-1}}\right)\cdot \tilde{S}^n_{t_{i-1}} \cdot \Delta B^n_{t_i}, \quad S_0 = 30$$

$$\hat{S}^n_{t_i} = \hat{S}^n_{t_{i-1}} + \alpha\left(t_{i-1}, \hat{S}^n_{t_{i-1}}\right)\cdot \hat{S}^n_{t_{i-1}} \cdot \frac{1}{4}\Delta W_t + \frac{\sqrt{11}}{4}\Delta C_t$$

où : $\Delta W_t = \sqrt{\Delta t^n_{i+1}} \cdot G_i$

avec $G_i \sim \mathcal{N}(0,1)$ iid

et

$\Delta C_t = \sqrt{\Delta t^n_{i+1}} \cdot H_i$

avec $H_i \sim \mathcal{N}(0,1)$ iid indépendante de G_i

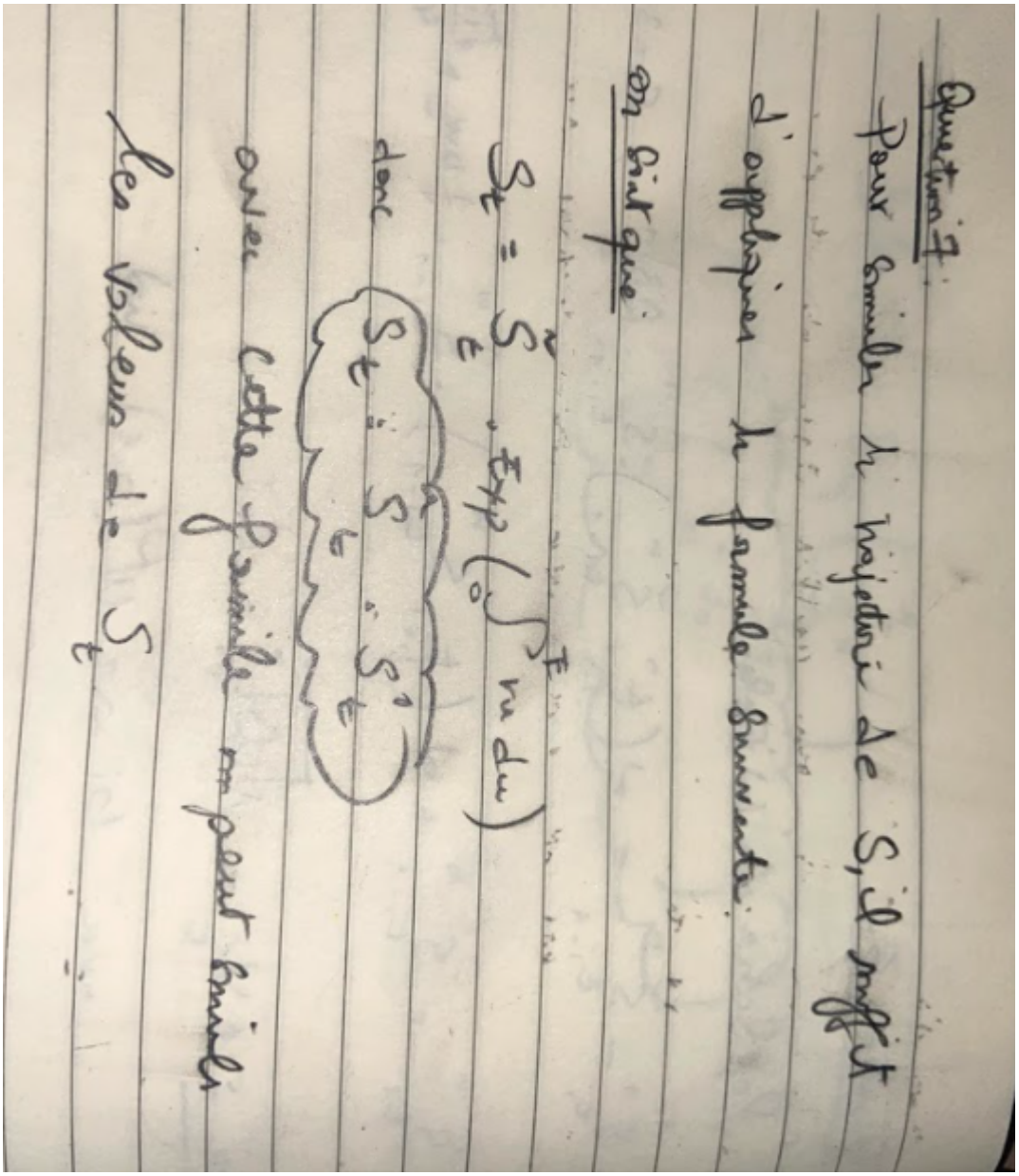
2) les trajectoires sont elles des solutions exacte de l'EDS :

Non les trajectoires ne sont pas des solutions exacts de l'EDS, mais selon la loi des grands nombres pour $N \to +\infty$

Si G_i est une trajectoire

$\frac{(G_1 + \dots + G_n)}{n}$ est une solution Exacte Si $N \to +\infty$

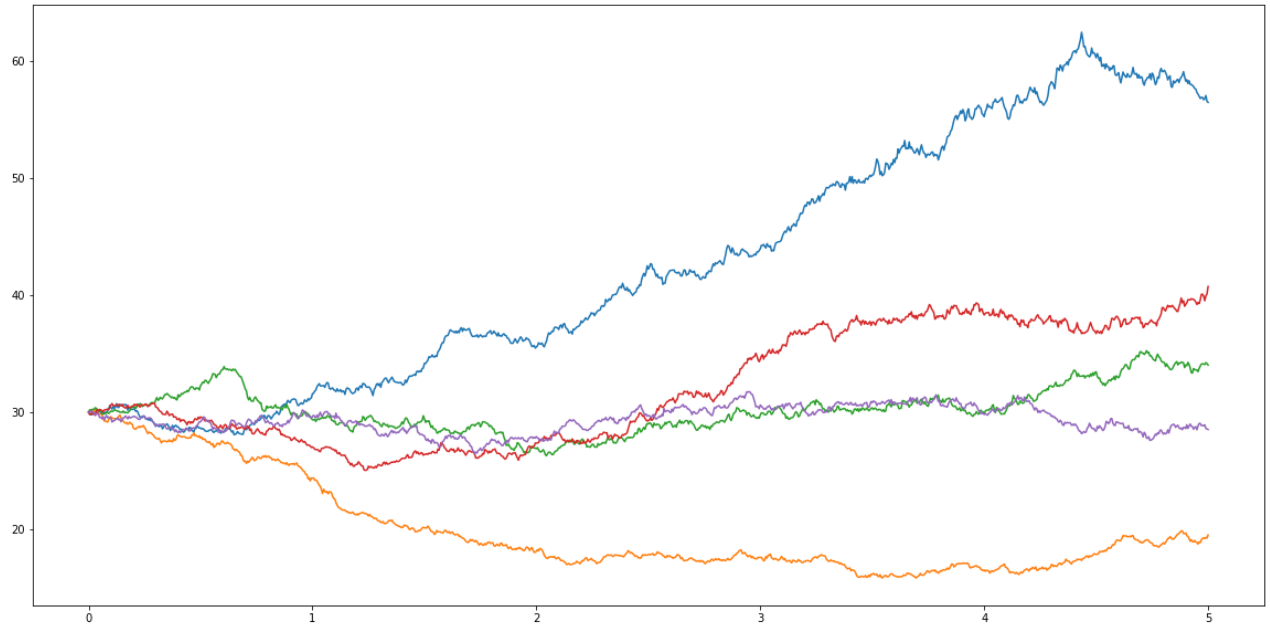
Question 7 :



In [119…

```
def sig(t,x):
    return 0.2* (1+ (np.sqrt(t)/(2*np.sqrt(T))) + (x/ (1+x**2)))


n =1000
r0 =3/100
alpha = 0.1
beta =1/100
theta = 5/100
T=5
pas = T/n
N=5
r=np.ones((n+1,N))*r0
integ=np.zeros((n+1,N))*r0

S0=30
TildeS=S0*np.ones((n+1,N))
S=S0*np.ones((n+1,N))

for j in range(N):
```

```
    for i in range(1,n+1):
        deltaWt = np.sqrt(pas)*sim.randn()
        r[i,j] = r[i-1,j] + alpha*(beta-r[i-1,j])*pas+ theta*deltaWt
        integ[i,j] = pas* np.sum(r[0:i-1,j]) #ici on a calculé l'intégrale pour chaque

        deltaCt=np.sqrt(pas)*sim.randn() # ici on simule le 2e mouvement brownien
        # on utilise l'ancien mouvment brownien Wt
        TildeS[i,j]=TildeS[i-1,j]+sig(pas*(i-1),TildeS[i-1,j])*TildeS[i-1,j]*0.25*delta
        #ici on simule l'actif S selon la formule déduite du tildeS
        S[i,j]=TildeS[i,j]*np.exp(integ[i,j])

dates=np.linspace(0,T,n+1) # n+1 dates
graph=plt.plot(dates,S)
plt.show()
```



Question 8

Question 6

Si le marché est complet, il existe une infinité de probabilité de risque neutre, on n'a pas le prix de réplication mais on peut obtenir un prix de sous réplication $V_t \geq g_T$

où g_T est la V.A dite marché complet qui est $\mathcal{F}_t$ mesurable

et donc sa réplication minimale V_0 est

$$V_0 = \left\{ \sup : E_Q\left(\tilde{g}_T\right) \quad \hat{g}_T \Rightarrow g_T \text{ actualisé} \right.$$

$$\text{et } V_t = \xi_T$$

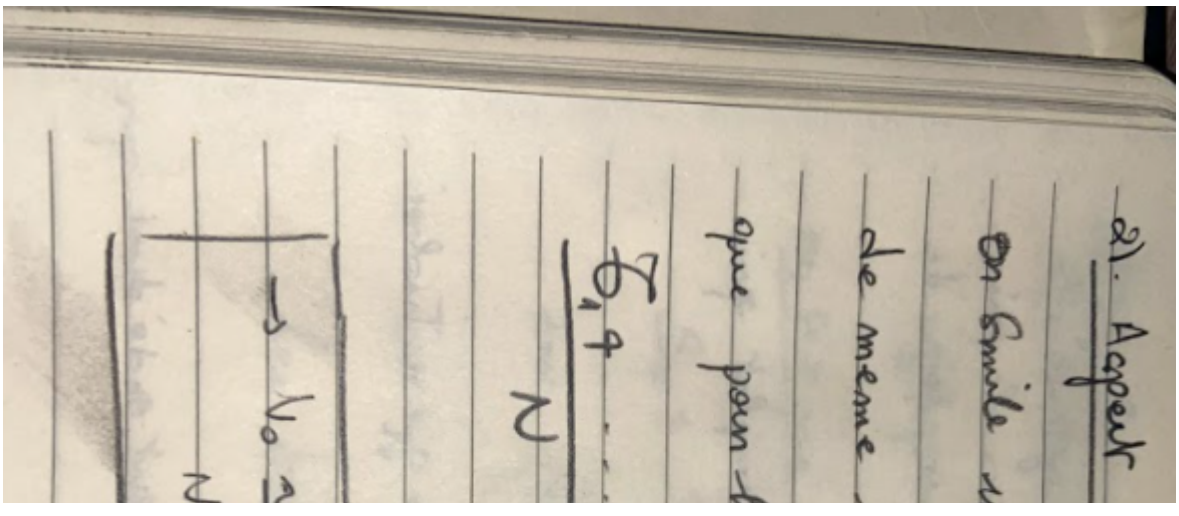
comme $\hat{S}$ est une Q martingale on peut en déduire que V l'est aussi.

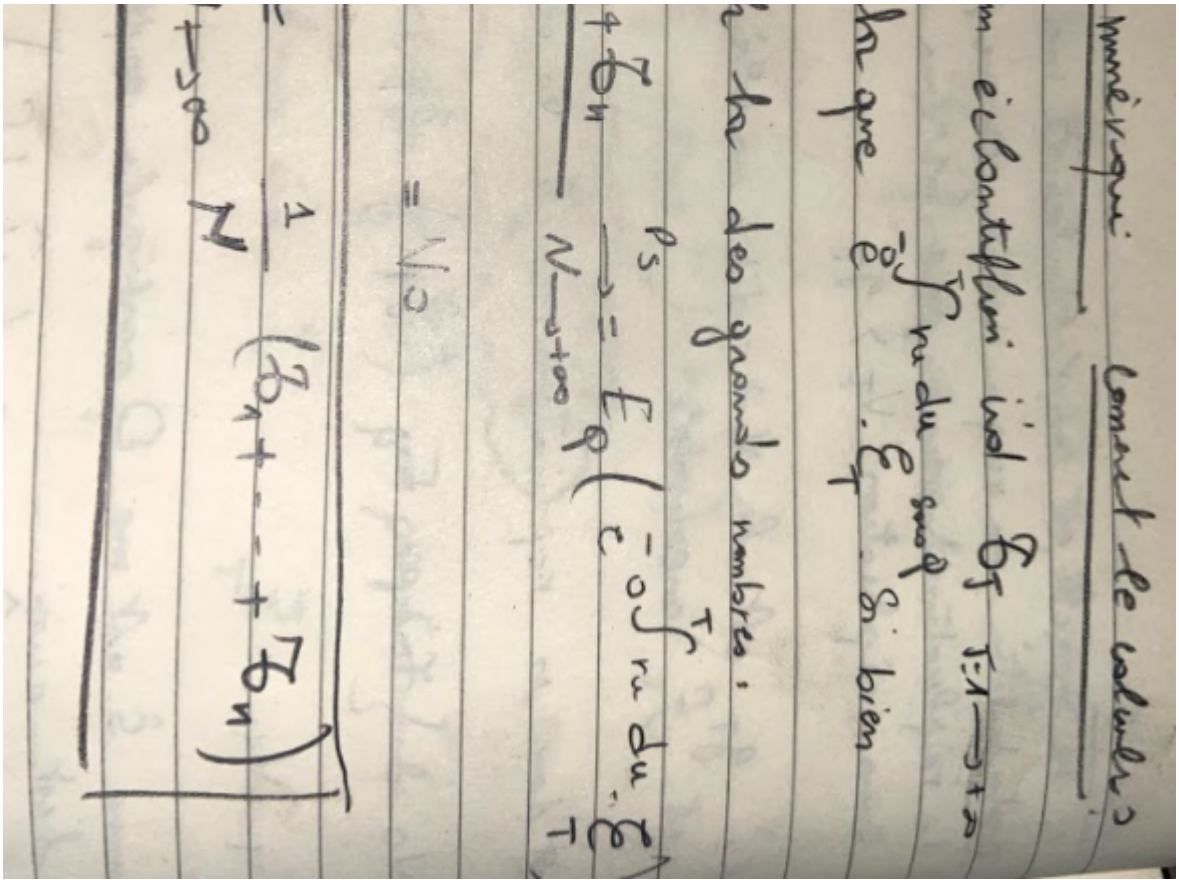
$$\forall t \leq T \quad \hat{V}_t = E_Q\left(\hat{V}_T / \mathcal{F}_{t_1}\right)$$

le prix en $t=0$ du payoff ξ_T c'est le montant que je dois investir pour le marché afin d'obtenir ξ_T en T donc il s'agit

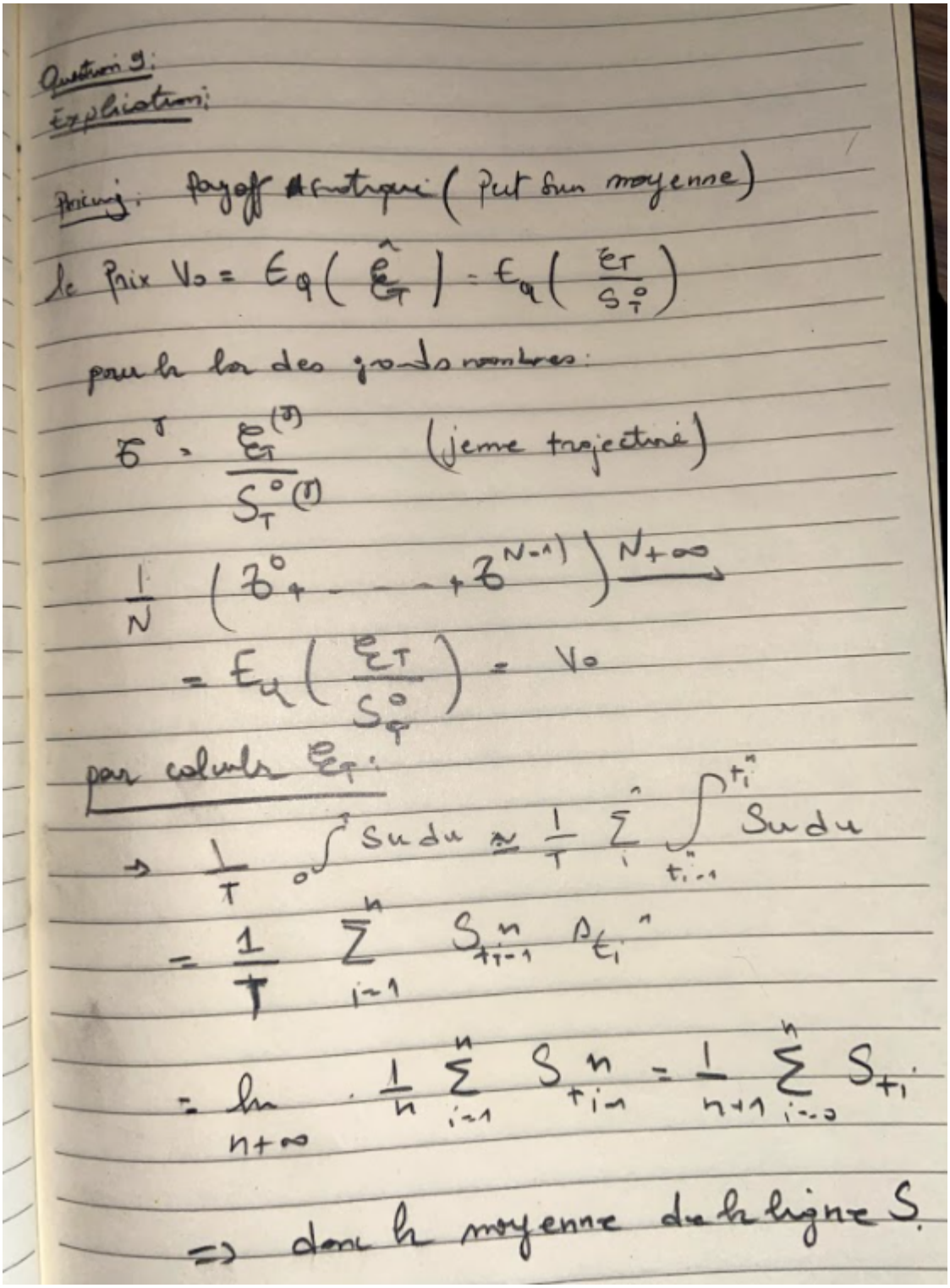
$$\Rightarrow V_0 = E_Q\left(e^{-\int_0^T r_u du} \cdot \xi_T\right)$$

Valeur initial du portfeuille de réplication





Question 9



In [120…

```
def pricingEu(k):
    def sig(t,x):
        return 0.2* (1+ (np.sqrt(t)/(2*np.sqrt(T))) + (x/ (1+x**2)))

    def call_function(x,k):
        return max(x-k,0)

    n =1000
    r0 =3/100
    alpha = 0.1
    beta =1/100
```

```
    theta = 5/100
    T=5
    pas = T/n
    N=100
    r=np.ones((n+1,N))*r0
    integ=np.zeros((n+1,N))*r0

    ListpayoffActualise=[] #cette liste contient toutes les possibilité des payoff poss

    S0=30
    TildeS=S0*np.ones((n+1,N))
    S=S0*np.ones((n+1,N))

    for j in range(N):
        for i in range(1,n+1):
            deltaWt = np.sqrt(pas)*sim.randn()
            r[i,j] = r[i-1,j] + alpha*(beta-r[i-1,j])*pas+ theta*deltaWt
            integ[i,j] = pas* np.sum(r[0:i-1,j]) #ici on a calculé l'intégrale pour cha

            deltaCt=np.sqrt(pas)*sim.randn() # ici on simule le 2e mouvement brownien
            # on utilise l'ancien mouvment brownien Wt
            TildeS[i,j]=TildeS[i-1,j]+sig(pas*(i-1),TildeS[i-1,j])*TildeS[i-1,j]*0.25*d
            #ici on simule l'actif S selon la formule déduite du tildeS
            S[i,j]=TildeS[i,j]*np.exp(integ[i,j])

        payoffActualisee =np.exp(-r*T)* call_function(S[n,j],k)
        ListpayoffActualise.append(payoffActualisee)

    V0Approx=np.mean(ListpayoffActualise)
    print("pour K = ",k)
    print("La valeur apoproximative du prix V0 de l'option Européenne: ", V0Approx)
    print("------------------")
```

In [121…

```
def pricingAsiatique(k):
    def sig(t,x):
        return 0.2* (1+ (np.sqrt(t)/(2*np.sqrt(T))) + (x/ (1+x**2)))

    def put_as(x,k):
        return max(k-x,0)

    n =1000
    r0 =3/100
    alpha = 0.1
    beta =1/100
    theta = 5/100
    T=5
    pas = T/n
    N=10
    r=np.ones((n+1,N))*r0
    integ=np.zeros((n+1,N))*r0

    PayoffAct=[]
    MoyS=[]

    S0=30
    TildeS=S0*np.ones((n+1,N))
    S=S0*np.ones((n+1,N))

    for j in range(N):
```

```
        for i in range(1,n+1):
            deltaWt = np.sqrt(pas)*sim.randn()
            r[i,j] = r[i-1,j] + alpha*(beta-r[i-1,j])*pas+ theta*deltaWt
            integ[i,j] = pas* np.sum(r[0:i-1,j]) #ici on a calculé l'intégrale pour cha
            deltaCt=np.sqrt(pas)*sim.randn() # ici on simule le 2e mouvement brownien
            # on utilise l'ancien mouvment brownien Wt
            TildeS[i,j]=TildeS[i-1,j]+sig(pas*(i-1),TildeS[i-1,j])*TildeS[i-1,j]*0.25*d
            #ici on simule l'actif S selon la formule déduite du tildeS
            S[i,j]=TildeS[i,j]*np.exp(integ[i,j])

        MoyS.append(np.mean(S[:,j]))
        aux=put_as(MoyS[-1],k)
        aux1=aux*np.exp(-integ[n,j])
        PayoffAct.append(aux1)

    V0Approx=np.mean(PayoffAct)
    print("pour K = ",k)
    print("La valeur apoproximative du prix V0 de l'option  Asiatique : ", V0Approx)
    print("------------------")
```

Etude du prix : pour l'option Europenne

In [122…

```
pricingEu(5)
pricingEu(19)
pricingEu(29)
pricingEu(31)
pricingEu(35)
pricingEu(60)
pricingEu(150)
```

```
pour K =  5
La valeur apoproximative du prix V0 de l'option Européenne:  26.36580410691194
------------------
pour K =  19
La valeur apoproximative du prix V0 de l'option Européenne:  14.747737232857975
------------------
pour K =  29
La valeur apoproximative du prix V0 de l'option Européenne:  7.774640420305349
------------------
pour K =  31
La valeur apoproximative du prix V0 de l'option Européenne:  8.140216742637497
------------------
pour K =  35
La valeur apoproximative du prix V0 de l'option Européenne:  7.5713407679087394
------------------
pour K =  60
La valeur apoproximative du prix V0 de l'option Européenne:  1.3703515046106738
------------------
pour K =  150
La valeur apoproximative du prix V0 de l'option Européenne:  0.0
------------------
```

Etude du prix : pour l'option asiatique

In [123…

```
pricingAsiatique(5)
pricingAsiatique(19)
pricingAsiatique(29)
pricingAsiatique(31)
```

```
pricingAsiatique(35)
pricingAsiatique(60)
pricingAsiatique(150)
```

```
pour K =  5
La valeur apoproximative du prix V0 de l'option  Asiatique :  0.0
------------------
pour K =  19
La valeur apoproximative du prix V0 de l'option  Asiatique :  0.0
------------------
pour K =  29
La valeur apoproximative du prix V0 de l'option  Asiatique :  0.7378871368446392
------------------
pour K =  31
La valeur apoproximative du prix V0 de l'option  Asiatique :  0.6208763500875791
------------------
pour K =  35
La valeur apoproximative du prix V0 de l'option  Asiatique :  4.0608439556842955
------------------
pour K =  60
La valeur apoproximative du prix V0 de l'option  Asiatique :  21.73099877725498
------------------
pour K =  150
La valeur apoproximative du prix V0 de l'option  Asiatique :  111.5483005383735
------------------
```