Range Option and Modified Sampling

Jingjie Zhou

1.Introduction

In this project, I use Monte Carlo integration to estimate the range option value from pseudorandom and quasirandom sampling. Comparing the difference of two random sampling methods. Also, I find the best importance sampling shift for pseudorandom variates.

2.Function Used

The assignment specifications require the following 6 functions:

- MakeBSPaths(S0,sigma,r,t,q,fixingTimes,fixingValues,samples,shift=0,integrationType='strong') uses the usual "step forward" method of forming Black-Scholes model price paths and return a dictionary with 'Paths' and 'Weights'

- MCRangePayoff(path, r, t, fixingTimes, finalT, fixingValues, KLow, KHigh, coupon) This function should return the time-t present value of the range option.

- MCRangeValues(S0,KLow, KHigh, coupon,sigma, r, t, q,fixingTimes, finalT, fixingValues, samples=1,shift=0, integrationType='strong',). This function should return a dictionary with 'Samples' and 'Weights'.

- MCRange(S0,KLow, KHigh, coupon,sigma, r, t, q,fixingTimes, finalT, fixingValues, samples=1,shift=0, integrationType='strong',) This function should return a dictionary with'Mean','Std Err' and 'Weights'

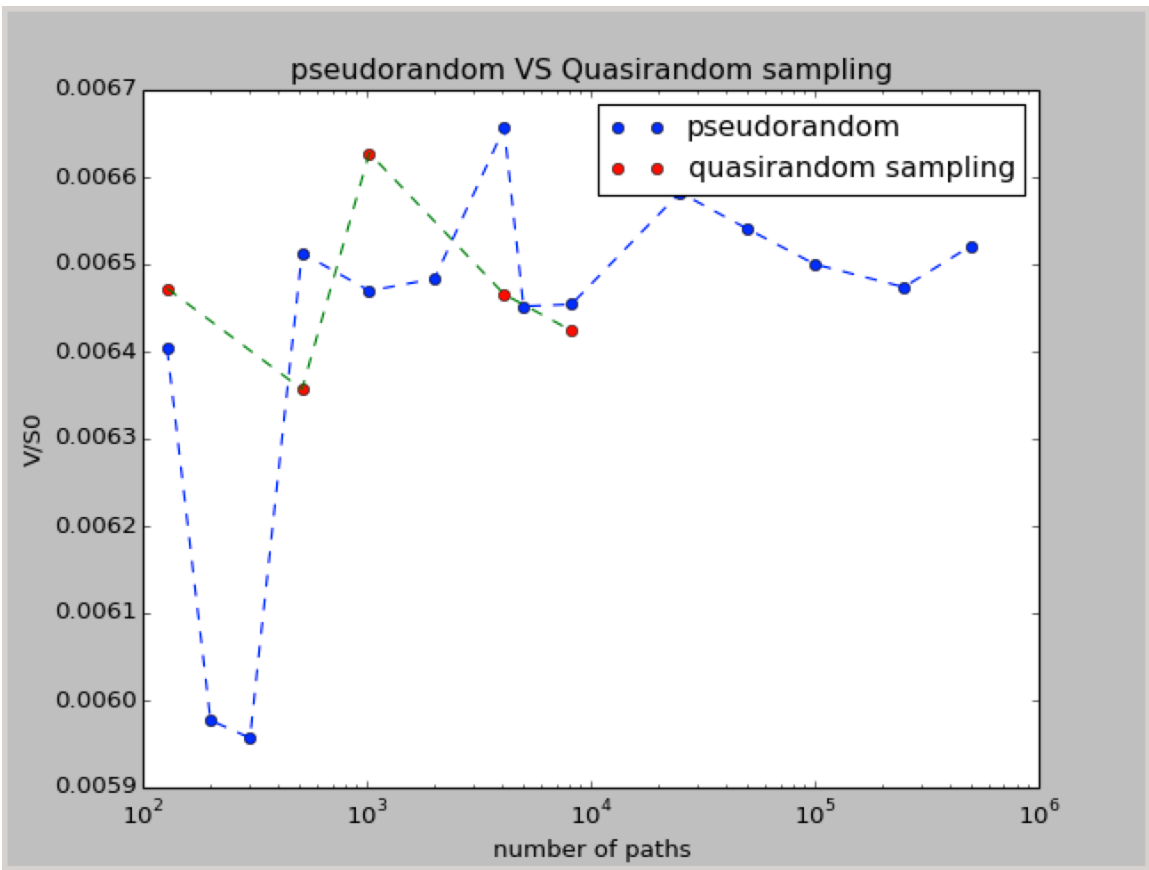
3.Test Suite

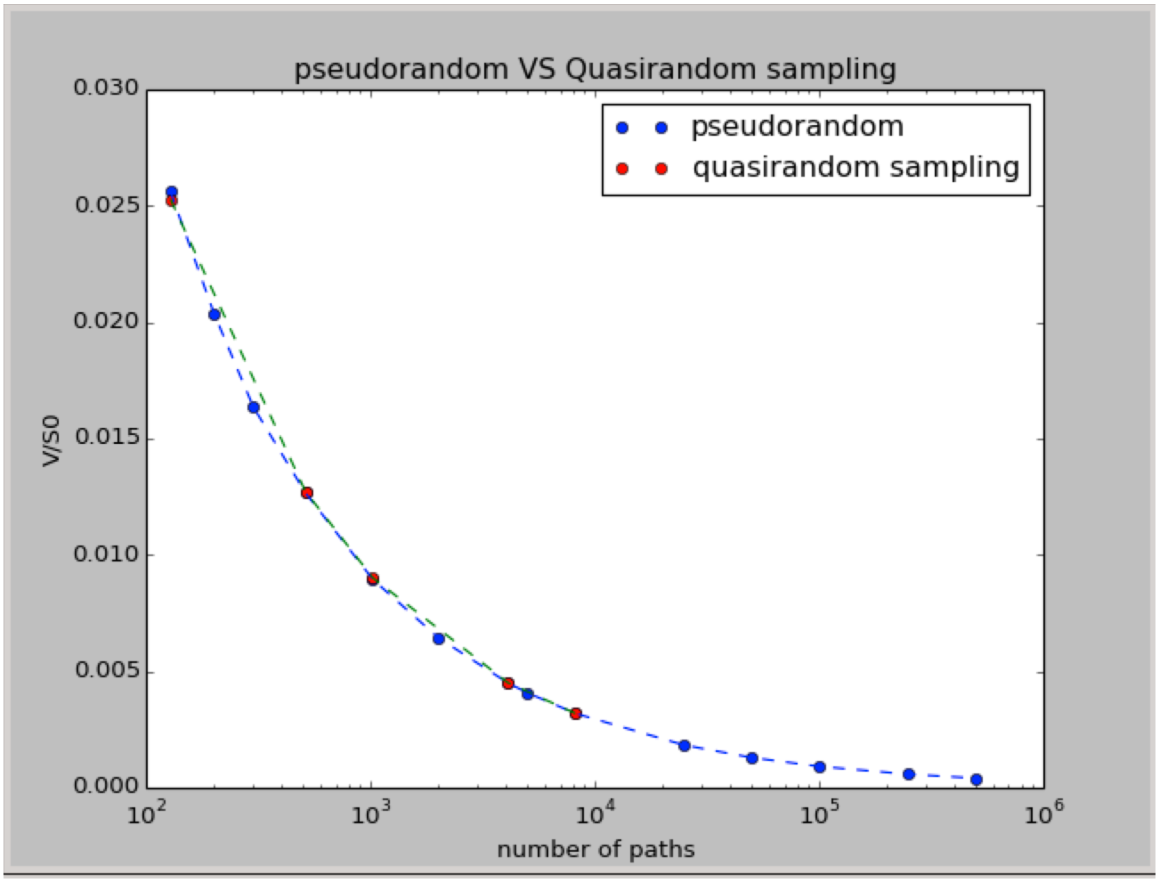
I pass all the tests.

4.Range of Input

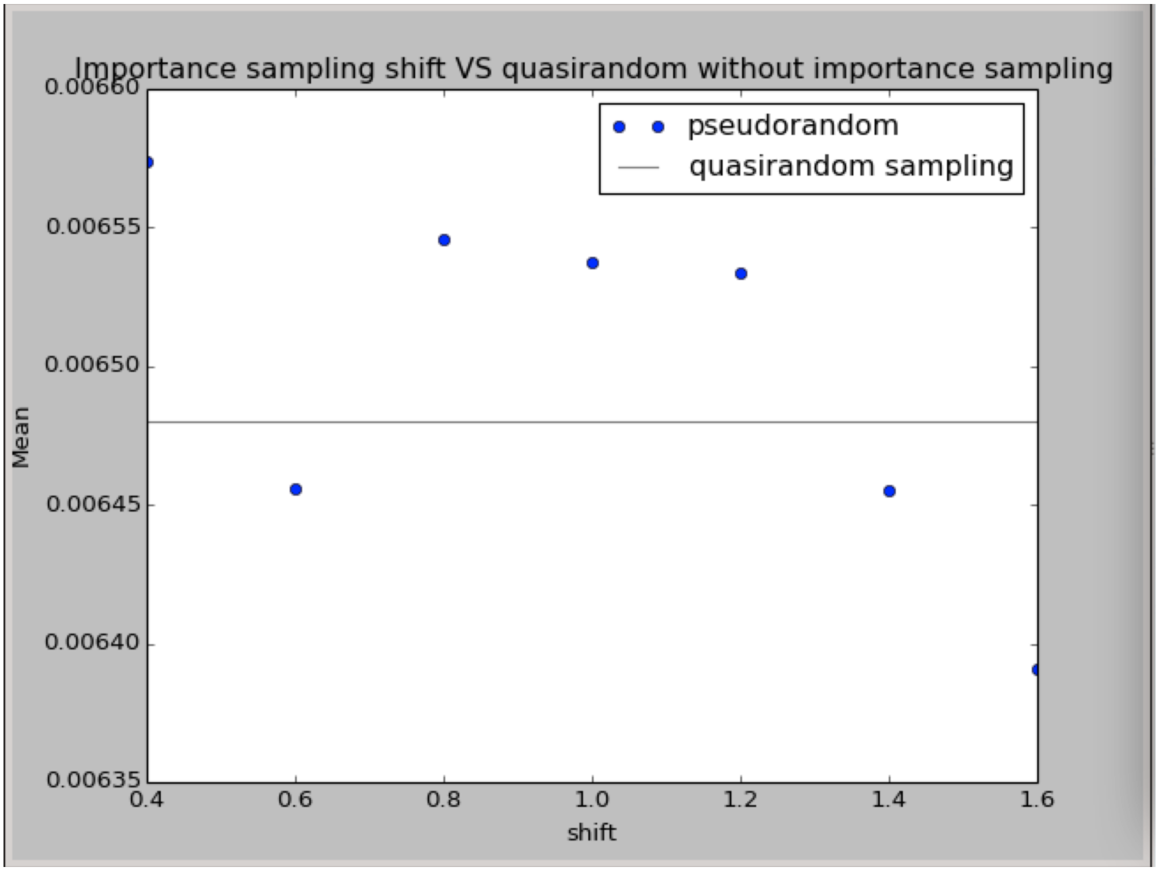
For the “interesting value”, r=0.03, t=0.0, S0=30, fixingTimes=[0.1,0.4,0.8], KLow=0.7*S0, KHigh=2.4*S0, shift=1.3, finalT=1.0, coupon=0.1*S0, sigma=0.75, q=0.01

5.Analysis

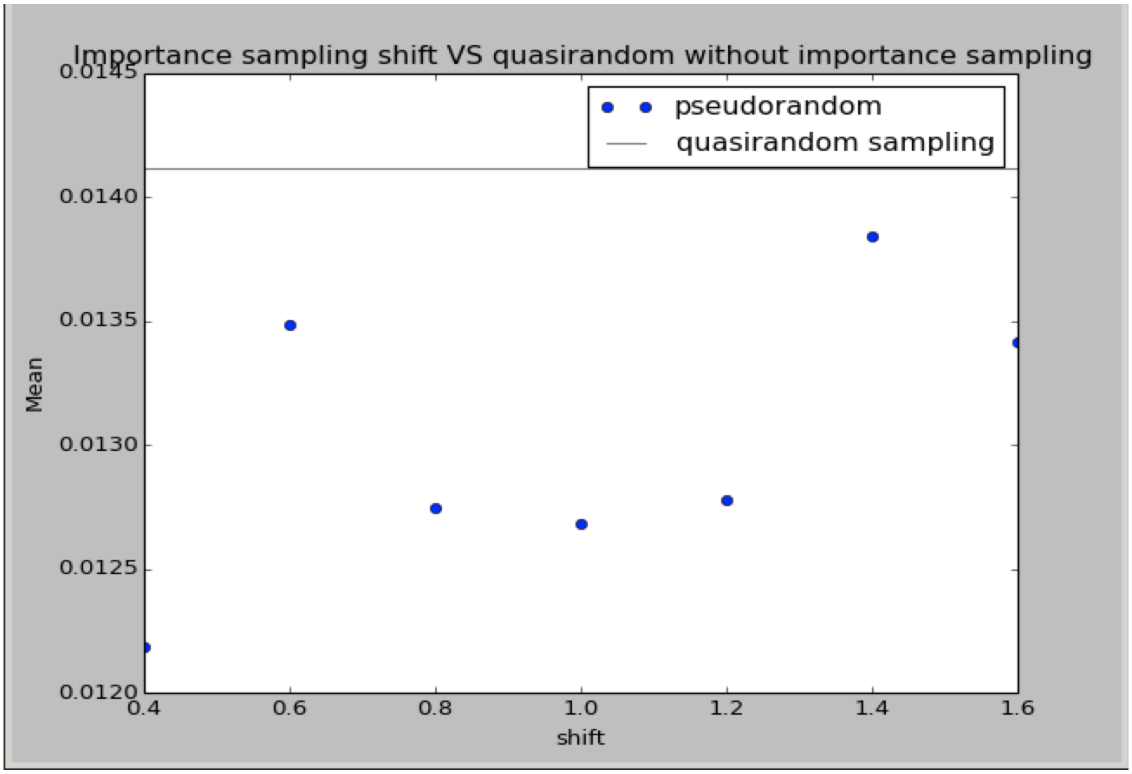




This graph shows the estimate of option value suing standard Monte Carlo integration from pseudorandom and quasirandom sampling. From this graph, we can see that quasirandom sampling method is more stable and converge to a stable state which is 0.0065. Pseudorandom sampling need sample counts large enough to converge to a stable state and it takes long time to get the value.



I use 2000000 as the sample count, which is M. And I get the 0.00648014280152 for the V/S0,which I take it as the ersatz exact value. Then I tried 0.4,0.6,0.8,1.0,1.2,1.4,1.6 as the shift ,and I find that using 1.4 as the shift is closer to the ersatz exact value.



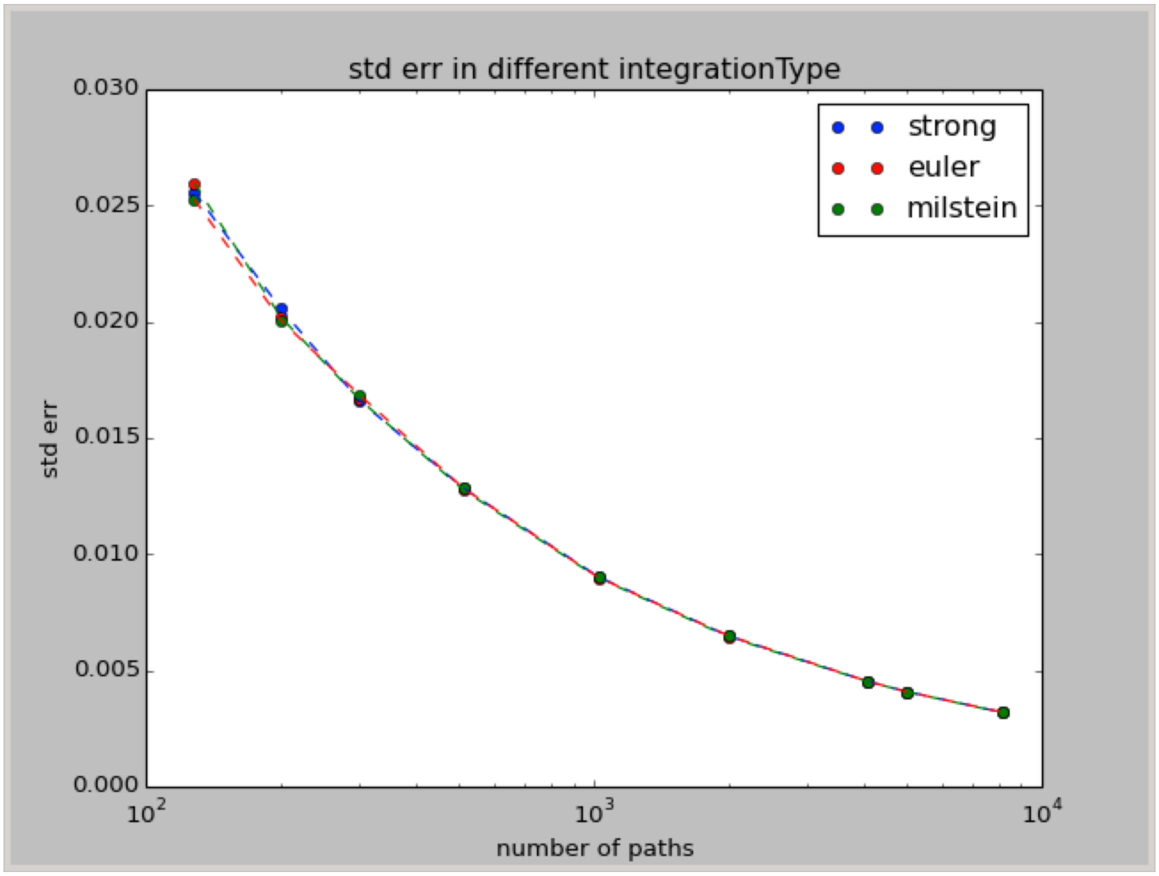
From this graph, I found that shift 1.4 is much closer to quasirandom sampling.

Then I tried an other option for KLow=0.7S0, KHigh=2.4*S0, shift=1.3, fixingTimes=[0.1,0.4,0.8], finalT=1.0.I calculated the std err using pseudorandom and quasirandom sampling.

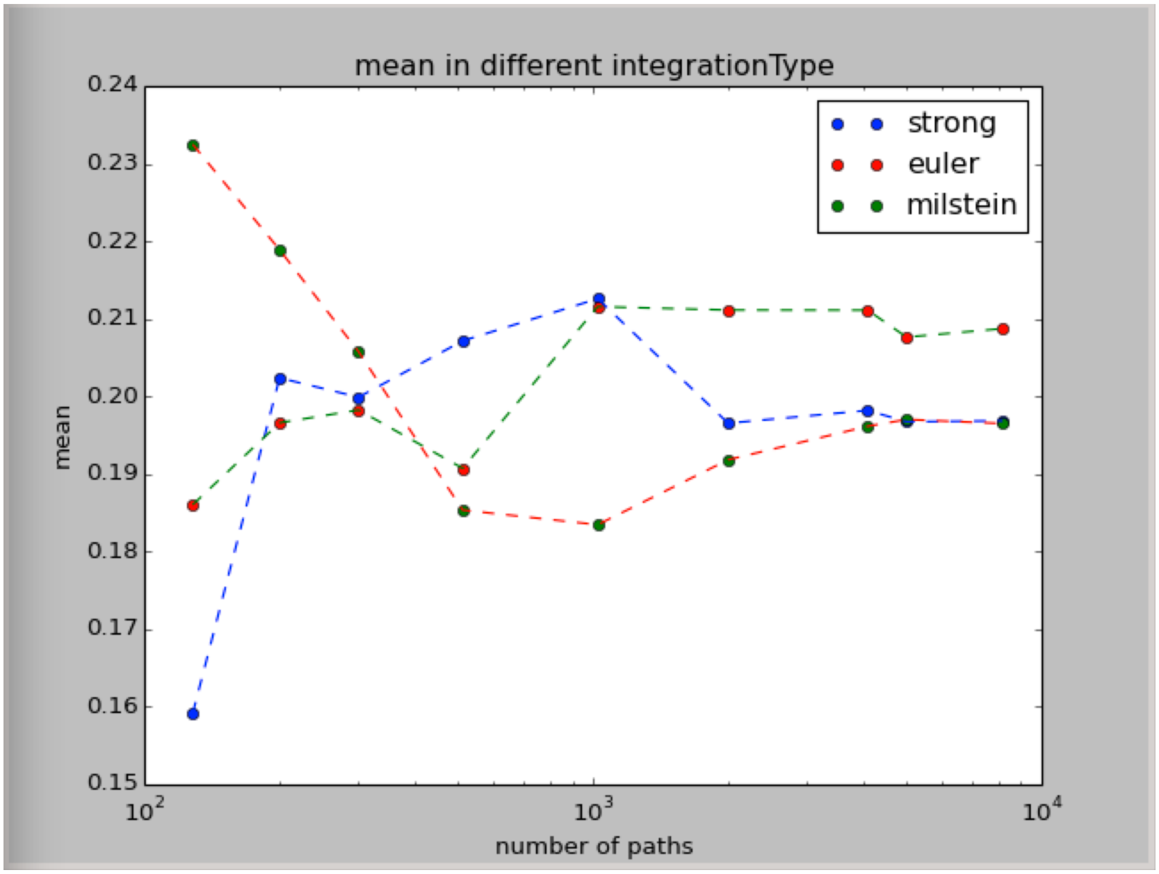
Pseudorandom std err=0.00288258114176;
Quasirandom std err=0.00288597857613.

It is obviously that Pseudorandom has smaller std err, so in this case the importance sampling beats quasirandom.

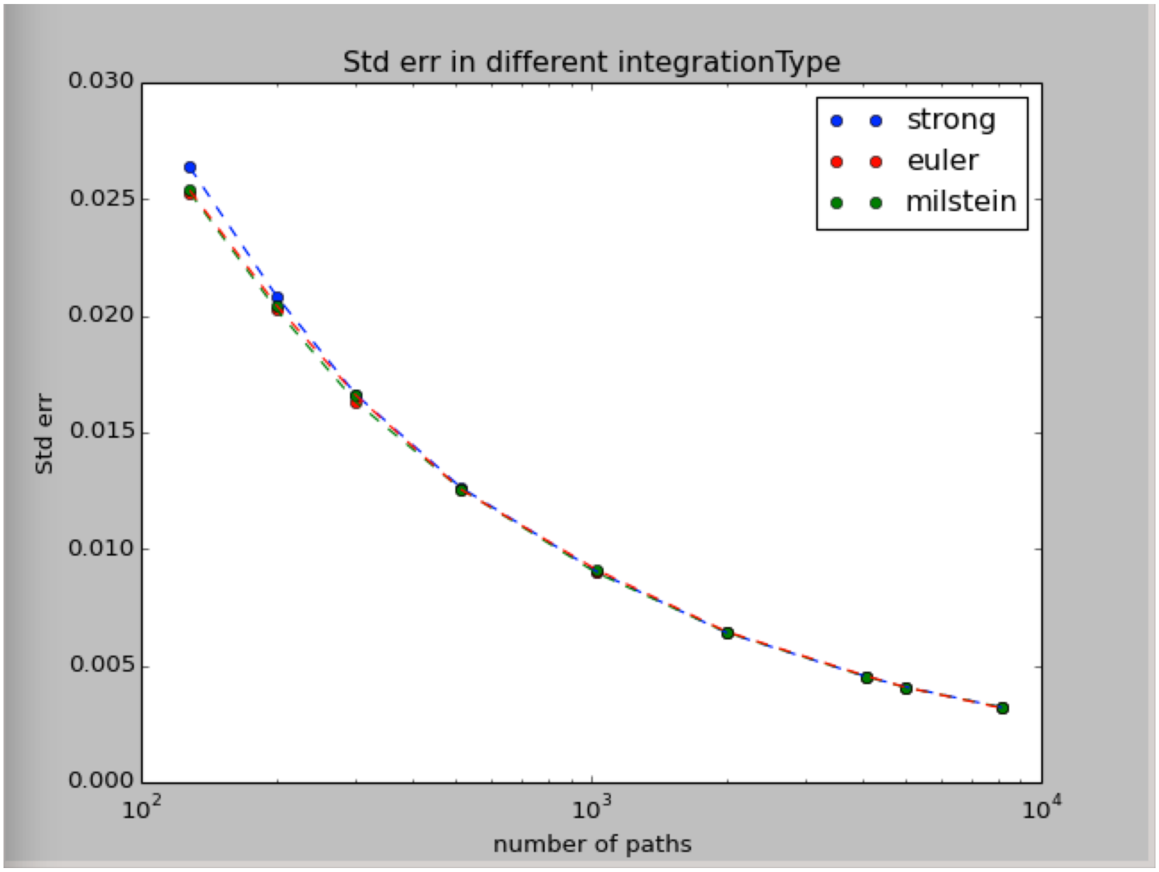
Valuations of the prototype option using strong, Milstein, and Euler integration to form the path.

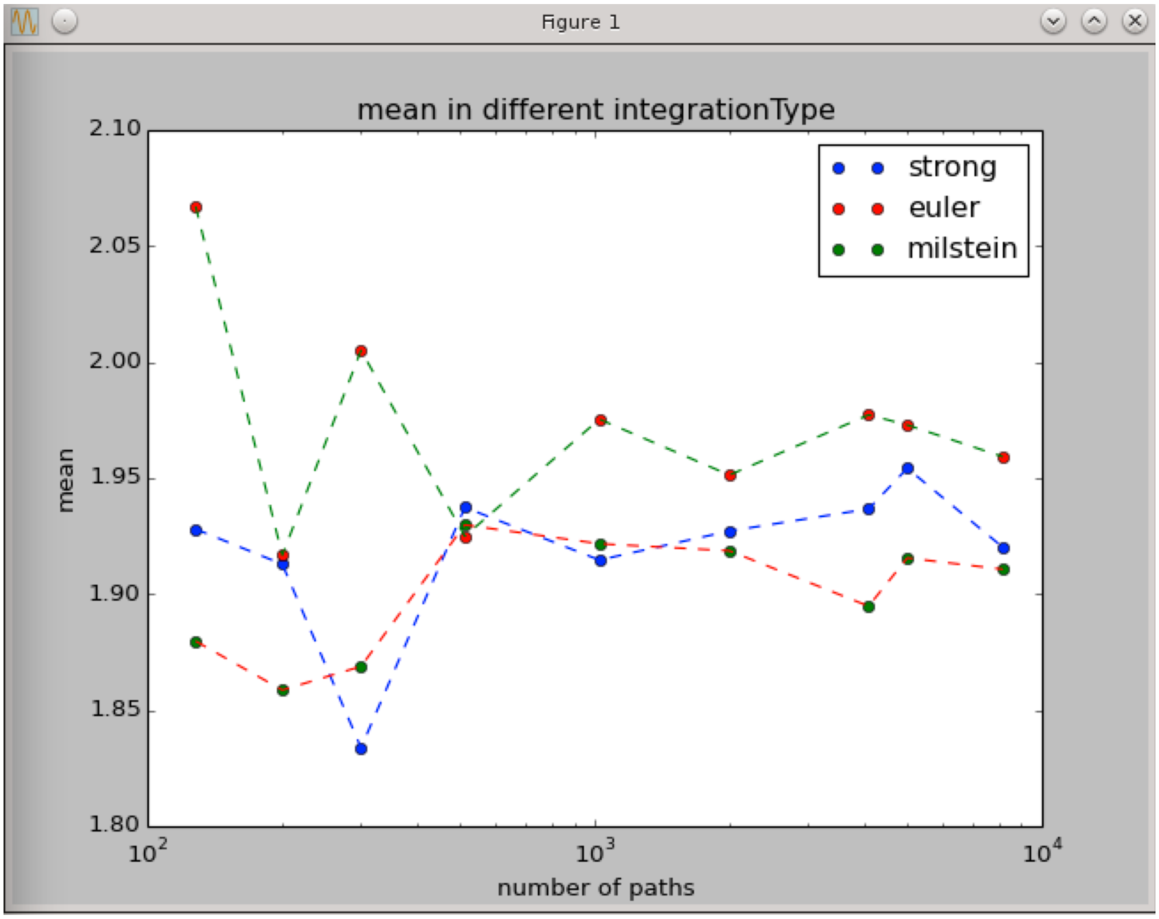


std err in different integrationType
strong
euler
milstein
0.030
0.025
0.020
0.015
0.010
0.005
0.000
std err
10^2
10^3
10^4
number of paths



Valuations of the “interesting” options using strong, Milstein, and Euler integration to form the path.


Std err in different integrationType
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Std err
strong
euler
milstein
10^2
10^3
10^4
number of paths



With M setting large enough, it seems that the order of these three integration type is: Milstein's mean> Strong's mean>Euler' mean.

6.Conclusion

Both pseudorandom and quasirandom samplings are good ways for Monte Carlo integration to find the BSpath and option value.