Exercise 4 (6.10)

Data

```
==== CRUDE ESTIMATOR RESULTS (0.0) ====================
beta =  1.0
meanY = 12.270398915786515,   Var(Y) = 246.2293125684911
meanX = 12.270398915786515,   Var(X) = 246.2293125684911
meanD = 0.0,   Var(D) = -3.410605131648481E-13
Covar(Y,X) = 246.22931256849128
Covar(Y,D) = -1.7053025658242404E-13

==== CV RESULTS (0.0) ====================
meanX = 12.105832683237722
Var(D) = -3.410605131648481E-13
ratio = -1.3851336772504644E-15



==== CRUDE ESTIMATOR RESULTS (75.0) ====================
beta =  1.0
meanY = 12.270398915786515,   Var(Y) = 246.2293125684911
meanX = 12.268266902320324,   Var(X) = 246.25836124721684
meanD = 0.002132013466191296,   Var(D) = 0.023268773503446027
Covar(Y,X) = 246.23220252110224
Covar(Y,D) = -0.0028899526111274554

==== CV RESULTS (75.0) ====================
meanX = 12.105832683237722
Var(D) = 0.023268773503446027
ratio = 9.448927291482577E-5



==== CRUDE ESTIMATOR RESULTS (80.0) ====================
beta =  0.9981809412635513
meanY = 12.270398915786515,   Var(Y) = 246.2293125684911
meanX = 12.243017025720734,   Var(X) = 246.48186099968055
meanD = 0.027381890065781178,   Var(D) = 0.41799251544807703
Covar(Y,X) = 246.1465905263618
Covar(Y,D) = 0.0827220421293191

==== CV RESULTS (80.0) ====================
meanX = 12.106082229614419
Var(D) = 0.41799251544807703
ratio = 0.0016958347918698113
```

```
==== CRUDE ESTIMATOR RESULTS (90.0) ====================
beta =  0.9609114394457439
meanY = 12.270398915786515,   Var(Y) = 246.2293125684911
meanX = 11.448613223222285,   Var(X) = 249.3080430191164
meanD = 0.8217856925642302,   Var(D) = 15.739764491961694
Covar(Y,X) = 239.8987955478229
Covar(Y,D) = 6.330517020668196

==== CV RESULTS (90.0) ====================
meanX = 12.080142920577472
Var(D) = 15.739764491961694
ratio = 0.06313380146646452



==== CRUDE ESTIMATOR RESULTS (95.0) ====================
beta =  0.8838385072189944
meanY = 12.270398915786515,   Var(Y) = 246.2293125684911
meanX = 9.73740547053013,   Var(X) = 245.7388981035333
meanD = 2.532993445256386,   Var(D) = 49.82491637645887
Covar(Y,X) = 221.07164714778276
Covar(Y,D) = 25.157665420708355

==== CV RESULTS (95.0) ====================
meanX = 11.830712642666452
Var(D) = 49.82491637645887
ratio = 0.202755513111591
```

Conclusion

Analysis of the results

The barrier value is given in the title of each section.

Note that CRNs were used betwen the simulations.

So, as expected, the VC reduction of the variance is very impressive. In the worst cases (e.g. at a barrier of 95) it still is able to reduce the variance immensely. The best values are obviously obtained when there is no difference between the two. I did some additionnal simulations with barriers set to 0 and 75. From this, we can observe that at 0, as expected, there is no difference between the two systems and this allows us to get $\beta = 1.0$ which in turn means that the estimator is then perfect (yields exactly the value expected by the Black-Scholes equation).

The value of β goes down slowly and in the worst case, it still is at 0.88 (for the barrier of 95).

So, as far as I know, this worked pretty well.

Other ways to improve the variance

The question also asks about other ways to reduce the variance of the simulation. According to the article referenced in the book (Monte Carlo methods for security pricing, Boyle, Broadie, and Glasserman, 1997), there are a few other ways to achieve this.

1- antithetic variates

The first one (apart from just a better estimator) was the antithetic variates. These can be used when the path of the GBM is generated, basically generating the opposite path. This assures that the sample mean is of 0 in the standard pricing scheme and also a reduced covariance without twice the computing cost (we get twice the sample size anyways which is pretty good already). The covariance induced between the pairs of variables generated is demonstrated to always be ≤ 0.

2- control variates (other ones)

Although I believe that the actual control variate is pretty much the best we could hope for, we could use another one that is also correlated and for which we know the expectation.

3- others

Well, the other ones are pretty complicated to understand (or even understand), so I'll just name them:

- Moment matching methods (quadratic resampling)
- Stratified and Latin hypercube sampling (this one is pretty nice)

These are all the ones suggested in the article from Boyle. There are probably many other ways to do it.