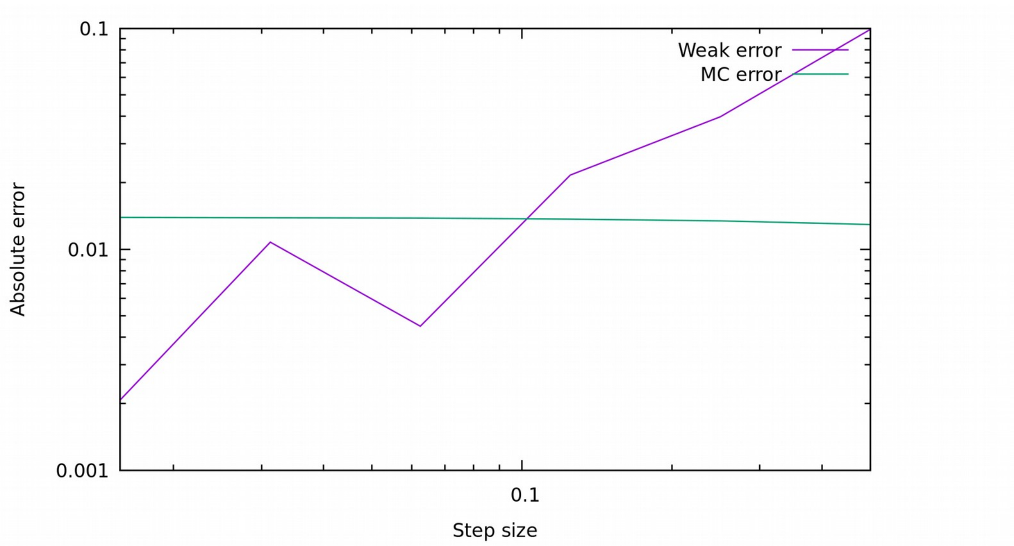
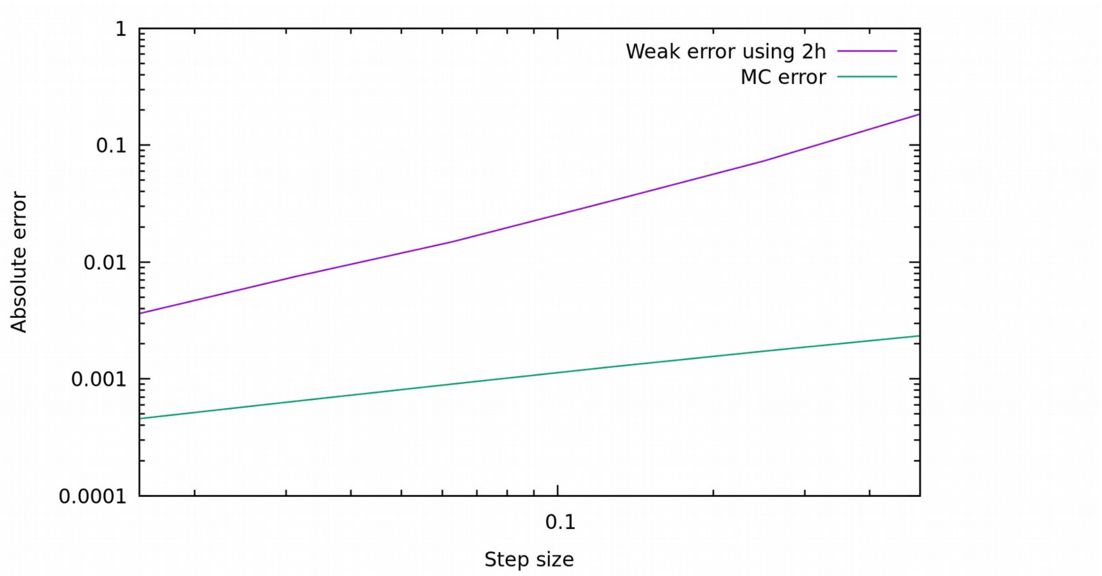
# Practical 2

2.a. Weak convergence of SDE
Analysis of the weak convergence of GBM SDE with N = 10000000 paths and 6 levels.



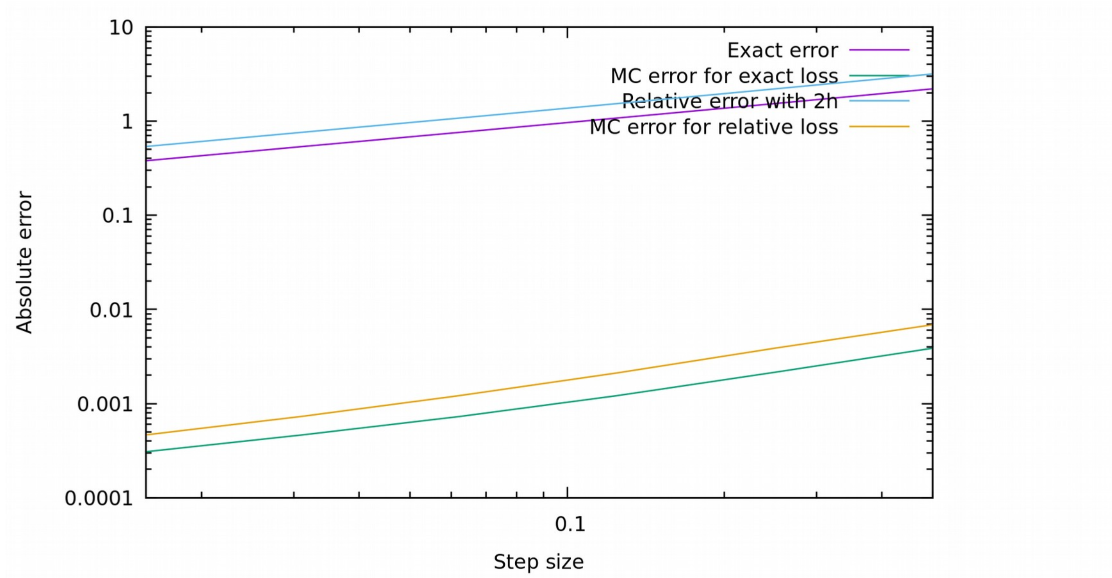

With a linear interpolation on the log-log plot above, we infer that the convergence of the weak error is of order O(h). The Monte-Carlo error stays roughly constant as expected.

2.b. Weak convergence of SDE with steps 2h
Analysis of the weak convergence of GBM SDE with N = 10000000 paths and 6 levels, with the 2h trick.



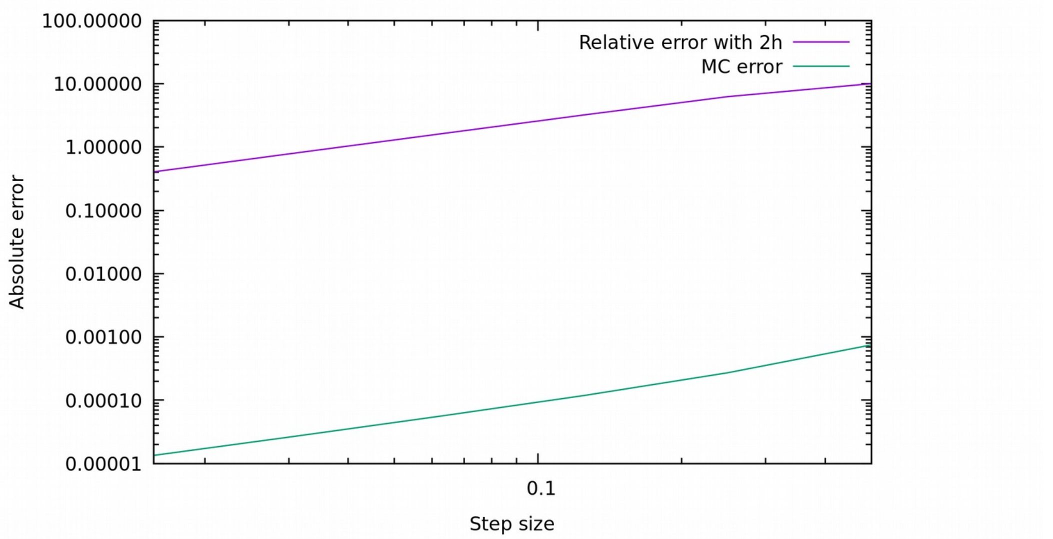

With a linear interpolation on the log-log plot above, we infer that the convergence of the weak error using the 2h trick is of order O(h). The Monte-Carlo error is now an order of magnitude lower, as the path with step size h and step size 2h are heavily correlated.

2.c. Strong convergence of SDE with steps 2h
Analysis of the strong convergence of GBM SDE with N = 10000000 paths and 6 levels, with the 2h trick.



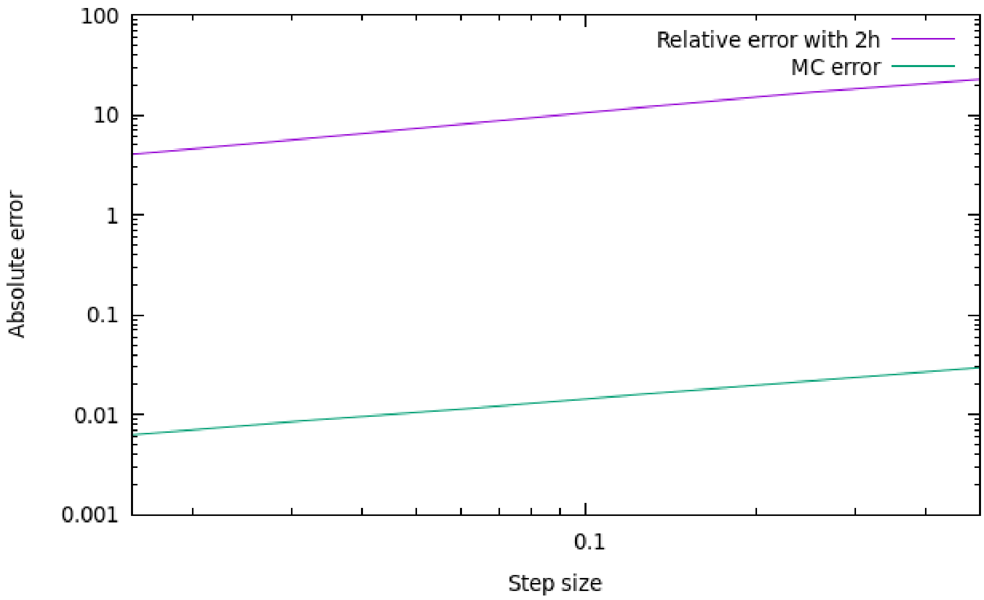

With a linear interpolation on the log-log plot above, we infer that the convergence of the strong error using the 2h trick and the exact error is of order O(h^0.5). The Monte-Carlo error are both several orders of magnitude lower.

3. Strong convergence of the Ornstein-Uhlenbeck process with steps 2h.
Analysis of the strong convergence of Ornstein-Uhlenbeck SDE with N = 10000000 paths and 6 levels, with the 2h trick.




With a linear interpolation on the log-log plot above, we infer that the convergence of the strong error using the 2h trick is of order O(h). My best guess for this faster convergence is that the Ornstein-Uhlenbeck process is mean-reverting, so that paths don't drift away like the geometric Brownian paths do.

4. Strong convergence of the Heston stochastic volatility model with steps 2h.
Analysis of the strong convergence of GBM SDE with N = 10000000 paths and 6 levels, with the 2h trick.




With a linear interpolation on the log-log plot above, we infer that the convergence of the strong error using the 2h trick is of order O(h^0.5). The strong error is still very large compared to the Monte-Carlo error, so decreasing the time step is needed here. The fact that the order is still O(h^0.5) by adding a “nested“ SDE is a bit surprising to me, as it adds an additional cumulated discretization error to the scheme.

5. Analysis of pricing financial options using a Milstein discretisation

```
 ---- option 1: European call ----

**********************************************************
*** MLMC file version 0.9     produced by             ***
*** C++ mlmc_test on Mon Nov 18 19:31:53 2019         ***
**********************************************************

**********************************************************
*** Convergence tests, kurtosis, telescoping sum check ***
*** using N =  20000 samples                           ***
**********************************************************

 l   ave(Pf-Pc)    ave(Pf)   var(Pf-Pc)    var(Pf)    kurtosis     check        cost
-------------------------------------------------------------------------------
 0   1.0064e+01  1.0064e+01  1.9757e+02  1.9757e+02  5.9784e+00  0.0000e+00  1.0000e+00
 1   1.8461e-01  1.0278e+01  1.5478e-01  2.0682e+02  5.8921e+01  4.7871e-02  2.0000e+00
 2   1.0313e-01  1.0384e+01  4.1604e-02  2.1064e+02  4.1238e+01  5.6546e-03  4.0000e+00
 3   5.4703e-02  1.0405e+01  1.1833e-02  2.1402e+02  1.8193e+01  5.5490e-02  8.0000e+00
 4   2.8138e-02  1.0478e+01  3.3422e-03  2.2091e+02  1.7814e+01  7.1988e-02  1.6000e+01
 5   1.3890e-02  1.0432e+01  8.1471e-04  2.1711e+02  1.1223e+01  9.5699e-02  3.2000e+01
 6   7.0644e-03  1.0572e+01  2.0848e-04  2.1891e+02  1.0084e+01  2.1211e-01  6.4000e+01
 7   3.5145e-03  1.0345e+01  5.4824e-05  2.1954e+02  9.9735e+00  3.6660e-01  1.2800e+02
 8   1.7623e-03  1.0516e+01  1.3085e-05  2.1534e+02  8.9373e+00  2.7110e-01  2.5600e+02

******************************************************
*** Linear regression estimates of MLMC parameters ***
******************************************************

 alpha = 0.967004  (exponent for MLMC weak convergence)
 beta  = 1.929346  (exponent for MLMC variance)
 gamma = 1.000000  (exponent for MLMC cost)

***************************
```

```
*** MLMC complexity tests ***
*****************************

  eps       value    mlmc_cost   std_cost  savings     N_l
---------------------------------------------------------
0.0050  1.0450e+01  1.353e+07  2.940e+09   217.24  11891386   232787    86464    32889    12275    4385    1558     563     233
0.0100  1.0447e+01  3.777e+06  3.747e+08    99.19   3368897    60380    21431     8155     3342    1200     385     159
0.0200  1.0433e+01  8.266e+05  4.670e+07    56.49    735020    13970     5419     1919      786     276      82
0.0500  1.0474e+01  1.403e+05  3.705e+06    26.42    125277     2391      955      365      125      46
0.1000  1.0401e+01  3.457e+04  2.283e+05     6.60     31687      670      223       81
```

The weak convergence is of order O(h), as in the Euler-Maruyama scheme. But we achieve a reasonable error with only 20000 samples here.

```
 ---- option 2: Asian call ----

**********************************************************
*** MLMC file version 0.9     produced by            ***
*** C++ mlmc_test on Mon Nov 18 19:32:04 2019        ***
**********************************************************

**********************************************************
*** Convergence tests, kurtosis, telescoping sum check ***
*** using N =  20000 samples                       ***
**********************************************************

 l   ave(Pf-Pc)    ave(Pf)   var(Pf-Pc)    var(Pf)    kurtosis     check        cost
-------------------------------------------------------------------------------
 0   5.6236e+00  5.6236e+00  6.0357e+01  6.0357e+01  5.6679e+00  0.0000e+00  1.0000e+00
 1   3.8958e-02  5.6751e+00  2.4225e-01  6.1737e+01  1.7728e+01  3.6483e-02  2.0000e+00
 2   4.2510e-02  5.7075e+00  3.7573e-02  6.2121e+01  1.2834e+01  2.9712e-02  4.0000e+00
 3   2.6405e-02  5.8037e+00  6.1997e-03  6.3959e+01  8.6445e+00  2.0598e-01  8.0000e+00
 4   1.4836e-02  5.7887e+00  1.2580e-03  6.4783e+01  8.2411e+00  8.7289e-02  1.6000e+01
 5   7.6109e-03  5.7747e+00  2.8261e-04  6.4769e+01  8.3631e+00  6.3308e-02  3.2000e+01
 6   3.9159e-03  5.8086e+00  6.3417e-05  6.3240e+01  7.0431e+00  8.8195e-02  6.4000e+01
 7   1.9410e-03  5.6982e+00  1.5320e-05  6.3330e+01  7.2586e+00  3.3278e-01  1.2800e+02
 8   9.8726e-04  5.8041e+00  3.7443e-06  6.3731e+01  7.0457e+00  3.1022e-01  2.5600e+02

******************************************************
*** Linear regression estimates of MLMC parameters ***
******************************************************

 alpha = 0.816718  (exponent for MLMC weak convergence)
 beta  = 2.263786  (exponent for MLMC variance)
 gamma = 1.000000  (exponent for MLMC cost)

*****************************
*** MLMC complexity tests ***
*****************************

  eps       value    mlmc_cost   std_cost  savings     N_l
---------------------------------------------------------
0.0050  5.7660e+00  4.811e+06  8.701e+08   180.87   3872483   207540    53889    15086     4477    1425     530     171      55
0.0100  5.7573e+00  1.173e+06  1.081e+08    92.16    963166    44548    12024     3745     1102     357     128      41
0.0200  5.7511e+00  2.902e+05  1.349e+07    46.49    238673    10968     3061      953      264     102      35
0.0500  5.7334e+00  4.517e+04  5.528e+05    12.24     37907     1677      486      152       47
0.1000  5.8321e+00  1.191e+04  6.822e+04     5.73      9905      431      200       43
```

Path-dependent option, the variance of the MLMC estimator is converging quite fast here. The order of convergence is better than the strong error in the Euler-Maruyama scheme (0.81 > 0.5), the MLMC method is quite powerful.

```
 ---- option 3: lookback call ----

**********************************************************
*** MLMC file version 0.9     produced by            ***
*** C++ mlmc_test on Mon Nov 18 19:32:10 2019        ***
**********************************************************

**********************************************************
```

```
*** Convergence tests, kurtosis, telescoping sum check ***
*** using N =  20000 samples                         ***
**********************************************************

 l   ave(Pf-Pc)    ave(Pf)   var(Pf-Pc)    var(Pf)    kurtosis     check       cost
-------------------------------------------------------------------------------------
 0   2.0492e+01  2.0492e+01  2.0466e+02  2.0466e+02  4.5965e+00  0.0000e+00  1.0000e+00
 1  -1.0042e-01  1.9989e+01  2.0793e+00  2.1332e+02  1.3111e+01  6.2498e-01  2.0000e+00
 2  -1.4702e-01  1.9416e+01  7.3927e-01  2.1272e+02  1.1482e+01  6.6953e-01  4.0000e+00
 3  -1.0367e-01  1.8766e+01  2.2793e-01  2.1307e+02  1.1575e+01  8.6789e-01  8.0000e+00
 4  -6.1888e-02  1.8503e+01  6.6431e-02  2.1850e+02  1.2493e+01  3.2030e-01  1.6000e+01
 5  -3.4587e-02  1.8087e+01  1.7425e-02  2.1400e+02  1.0537e+01  6.0852e-01  3.2000e+01
 6  -1.7557e-02  1.7997e+01  4.3729e-03  2.1480e+02  1.0159e+01  1.1648e-01  6.4000e+01
 7  -8.8618e-03  1.7600e+01  1.1472e-03  2.1635e+02  1.0021e+01  6.2186e-01  1.2800e+02
 8  -4.5789e-03  1.7621e+01  2.9928e-04  2.1070e+02  1.1564e+01  4.0609e-02  2.5600e+02
 9  -2.2201e-03  1.7422e+01  7.1190e-05  2.1361e+02  1.0043e+01  3.1774e-01  5.1200e+02
10  -1.1070e-03  1.7376e+01  1.8453e-05  2.1240e+02  1.0689e+01  7.2423e-02  1.0240e+03

******************************************************
*** Linear regression estimates of MLMC parameters ***
******************************************************

 alpha = 0.804660  (exponent for MLMC weak convergence)
 beta  = 1.890051  (exponent for MLMC variance)
 gamma = 1.000000  (exponent for MLMC cost)

***************************
*** MLMC complexity tests ***
***************************

  eps       value   mlmc_cost   std_cost  savings     N_l
---------------------------------------------------------
0.0050  1.9979e+01  2.727e+07  5.833e+09   213.90  17203052  1237100   520434   206011    77203    28345    10410     3760     1318
460
0.0100  2.0004e+01  6.686e+06  7.192e+08   107.56   4263627   304823   128305    50604    18870     7095     2536      921      335
0.0200  1.9985e+01  1.688e+06  3.646e+08   215.98   1070056    76980    33007    12250     4592     1725      615      221       85       31
0.0500  2.0025e+01  2.593e+05  7.332e+06    28.27    172706    12515     4851     1902      757      275       95
0.1000  2.0028e+01  5.570e+04  4.661e+05     8.37     39861     2800     1121      398      161
```

Same as the Asian option.

```
 ---- option 4: digital call ----

**********************************************************
*** MLMC file version 0.9     produced by              ***
*** C++ mlmc_test on Mon Nov 18 19:32:38 2019          ***
**********************************************************

**********************************************************
*** Convergence tests, kurtosis, telescoping sum check ***
*** using N = 200000 samples                           ***
**********************************************************

 l   ave(Pf-Pc)    ave(Pf)   var(Pf-Pc)    var(Pf)    kurtosis     check       cost
-------------------------------------------------------------------------------------
 0   5.6951e+01  5.6951e+01  1.0000e-10  1.0000e-10 -9.3731e+19  0.0000e+00  1.0000e+00
 1  -2.7025e+00  5.4314e+01  2.0321e-01  7.2260e+02  1.1225e+01  3.6092e-01  2.0000e+00
 2  -7.1986e-01  5.3488e+01  1.9119e+00  1.1918e+03  1.8826e+01  2.5182e-01  4.0000e+00
 3  -1.9782e-01  5.3482e+01  1.2835e+00  1.5065e+03  2.4435e+01  3.8405e-01  8.0000e+00
 4  -5.9823e-02  5.3140e+01  5.6668e-01  1.7230e+03  3.7001e+01  5.2037e-01  1.6000e+01
 5  -2.3277e-02  5.3420e+01  2.0433e-01  1.8722e+03  3.9073e+01  5.3113e-01  3.2000e+01
 6  -8.5493e-03  5.3184e+01  7.8465e-02  1.9776e+03  6.3606e+01  3.8521e-01  6.4000e+01
 7  -4.1095e-03  5.3198e+01  2.7400e-02  2.0529e+03  7.4843e+01  3.0158e-02  1.2800e+02
 8  -1.3468e-03  5.3197e+01  1.0663e-02  2.1032e+03  1.2577e+02  9.4702e-04  2.5600e+02

 WARNING: kurtosis on finest level = 125.765526
 indicates MLMC correction dominated by a few rare paths;
 for information on the connection to variance of sample variances,
 see http://mathworld.wolfram.com/SampleVarianceDistribution.html

******************************************************
*** Linear regression estimates of MLMC parameters ***
******************************************************

 alpha = 1.535899  (exponent for MLMC weak convergence)
```

```
 beta  = 0.880433  (exponent for MLMC variance)
 gamma = 1.000000  (exponent for MLMC cost)

****************************
*** MLMC complexity tests ***
****************************

  eps      value   mlmc_cost   std_cost  savings     N_l
-----------------------------------------------------------
0.0100  5.3236e+01  3.776e+06  3.504e+09  927.80      200   70780   153687   89550   41747   18347    8635    3872
0.0200  5.3236e+01  7.851e+05  4.219e+08  537.37      200   16281    35108   20233    9580    4540    2367
0.0500  5.3271e+01  5.684e+04  1.470e+07  258.68      200    1815     4116    2122    1223
0.1000  5.3280e+01  1.193e+04  3.676e+06  308.06      200     352      843     493     232
0.2000  5.3417e+01  2.200e+03  4.017e+05  182.60      200     200      200     100
```

Discontinuous payoff, the variance of the difference between the coarse and the fine level is not decreasing fast (0.88), but its average is shrinking fast, which is surprising to me. Maybe it's because of the discontinuity : decreasing the time step won't affect much the final probability of being in-the-money, but the variance can remain quite high if you finish nearly at-the-money.

```
 ---- option 5: barrier call ----

**********************************************************
*** MLMC file version 0.9     produced by              ***
*** C++ mlmc_test on Mon Nov 18 19:33:18 2019          ***
**********************************************************

**********************************************************
*** Convergence tests, kurtosis, telescoping sum check ***
*** using N = 200000 samples                           ***
**********************************************************

 l   ave(Pf-Pc)    ave(Pf)   var(Pf-Pc)    var(Pf)    kurtosis     check        cost
-------------------------------------------------------------------------------------
 0   9.4249e+00  9.4249e+00  1.8948e+02  1.8948e+02  6.7485e+00  0.0000e+00  1.0000e+00
 1   1.6052e-01  9.6448e+00  1.6798e-01  2.0048e+02  5.0294e+01  3.1269e-01  2.0000e+00
 2   1.4501e-01  9.7560e+00  1.5470e-01  2.0639e+02  9.5410e+01  1.7472e-01  4.0000e+00
 3   9.7639e-02  9.9464e+00  1.3865e-01  2.1488e+02  3.5931e+02  4.7038e-01  8.0000e+00
 4   4.5567e-02  9.8334e+00  5.0557e-02  2.1210e+02  2.7356e+03  8.0227e-01  1.6000e+01
 5   2.0725e-02  9.9590e+00  1.1591e-02  2.1500e+02  1.2428e+03  5.3251e-01  3.2000e+01
 6   9.9106e-03  9.9413e+00  3.6583e-03  2.1697e+02  3.0275e+03  1.3966e-01  6.4000e+01
 7   4.6982e-03  9.9590e+00  9.8927e-04  2.1801e+02  5.4389e+03  6.5588e-02  1.2800e+02
 8   2.2779e-03  9.8992e+00  2.6170e-04  2.1676e+02  1.9218e+03  3.1370e-01  2.5600e+02

 WARNING: kurtosis on finest level = 1921.844116
 indicates MLMC correction dominated by a few rare paths;
 for information on the connection to variance of sample variances,
 see http://mathworld.wolfram.com/SampleVarianceDistribution.html

******************************************************
*** Linear regression estimates of MLMC parameters ***
******************************************************

 alpha = 0.937498  (exponent for MLMC weak convergence)
 beta  = 1.423632  (exponent for MLMC variance)
 gamma = 1.000000  (exponent for MLMC cost)

****************************
*** MLMC complexity tests ***
****************************

  eps      value   mlmc_cost   std_cost  savings     N_l
-----------------------------------------------------------
0.0050  9.9568e+00  2.096e+07  5.919e+09  282.44  14450772   307386   213755   139517    66932    26937    11103     4173     1798
550
0.0100  9.9621e+00  4.710e+06  3.721e+08   79.00   3413829    72123    50950    32204    15986     6816     2095      644
0.0200  9.9550e+00  1.202e+06  9.302e+07   77.41    865512    20341    12790     9003     3853     1589      503      215
0.0500  9.9739e+00  1.878e+05  7.406e+06   39.44    135057     3248     2299      944      375      501      116
0.1000  9.8828e+00  3.668e+04  4.525e+05   12.34     30037      531      611      242       75
```

Path dependent option with discontinuous payoff, it is impressive to see a weak convergence of order O(h) hear, even though the kurtosis is very high.