D. Advanced Monte Carlo

This section will build upon the provided Monte Carlo code from the previous section, by adding methods to track the accuracy of the MC simulation. You are expected to submit code in addition to a document with the answers to the below questions. The document should contain a detailed, complete analysis and will be graded on how well it presents understanding of the accuracy and efficiency of Monte Carlo methods in the below context.

We wish to add functionality to the Monte Carlo pricer by providing estimates for the *standard deviation* (SD) and *standard error* (SE), defined by:

$$\mathrm{SD} = \sqrt{\frac{\sum C_{T,j}^2 - \frac{1}{M}\left(\sum C_{T,j}\right)^2}{M-1}} \times \exp\left(-rT\right).$$

$$SE = \frac{SD}{\sqrt{M}}$$

where

$C_{T,j}$ = call output price at $t = T$ for the jth simulation, $1 \leq j \leq M$,

M = number of simulations.

Implement this new functionality and test the software for a range of data for call and put options.

Answer the following questions:

a) Create generic functions to compute the standard deviation and standard error based on the above formulae. The inputs are a vector of size M (M = NSIM), the interest-free rate and expiry time T. Integrate this new code into *TestMC.cpp*. Make sure that the code compiles.

b) Run the MC program again with data from Batches 1 and 2. Experiment with different values of NT (time steps) and NSIM (simulations or draws). How do SD and SE react for these different run parameters, and is there any pattern in regards to the accuracy of the MC (when compared to the exact method)?

For part a ; the SD and SE code is inside the TestMC which can be accessed by directly running the project inside \Level 9\Exercise D\Projects\MC.

For part b; upon running different simulations for batch 1 and 2 and checking how SD and SE react according to Nsim and NT.

Batch #1 - call	NT	NSIM	Closed Solution	Value -(Call)	Error	SD	SE
	500	500,000	2.13337	2.1253	0.00807	4.51365	0.00638326
	500	900,000	2.13337	2.13058	0.00279	4.51349	0.00475764
	300	1,000,000	2.13337	2.1347	-0.00133	4.51766	0.00451766
	500	1,000,000	2.13337	2.13071	0.00266	4.51286	0.00451286
	500	3,000,000	2.13337	2.13232	0.00105	4.51043	0.0026041
	300	15,000,000	2.13337	2.13179	0.00158	4.51307	0.00116527
	500	15,000,000	2.13337	2.13335	0.00002	4.51475	0.0011657

	Closed Solution	Value - (Put)	Error	SD	SE
Batch #1 - Put	5.84628	5.85493	-0.00865	6.05373	0.00856126
	5.84628	5.84038	0.0059	6.04769	0.00637483
	5.84628	5.85369	-0.00741	6.05714	0.00605714
	5.84628	5.84125	0.00503	6.04743	0.00604743
	5.84628	5.84109	0.00519	6.04822	0.00349194
	5.84628	5.84504	0.00124	6.04849	0.00156171
	5.84628	5.84624	0.00004	6.0481	0.00156161

Batch #2 - Call	500	500,000	7.96557	7.9418	0.02377	13.1421	0.0185857
	500	700,000	7.96557	7.94876	0.01681	13.1404	0.0157058
	500	900,000	7.96557	7.96172	0.00385	13.1433	0.0138542
	300	1,000,000	7.96557	7.97235	-0.00678	13.1535	0.0131535
	500	1,000,000	7.96557	7.96142	0.00415	13.1421	0.0131421
	500	3,000,000	7.96557	7.96675	-0.00118	13.1372	0.0075848
	300	15,000,000	7.96557	7.96437	0.0012	13.1427	0.0033934
	500	15,000,000	7.96557	7.96672	-0.00115	13.1473	0.0033946

Batch #2 - Put	7.96557	7.97869	-0.01312	10.4208	0.0147373
	7.96557	7.97051	-0.00494	10.4154	0.0124488
	7.96557	7.95525	0.01032	10.4058	0.0109687
	7.96557	7.98455	-0.01898	10.4229	0.0104229
	7.96557	7.95663	0.00894	10.4052	0.0104052
	7.96557	7.95794	0.00763	10.4058	0.006008
	7.96557	7.9652	0.00037	10.4069	0.002687
	7.96557	7.9666	-0.00103	10.4055	0.002687

SD and SE appear to decrease as NSIM approaches infinity, with SE having a more direct correlation with NSIM approaching infinity, given how SE is derived. This makes sense – the more data we have, the more precise our estimate is. We can also observe that NSIM impacts SE more so than NT.