October 23, 2021
David Skovmand & Rolf Poulsen
Dept' of Math'

Computational Finance: Hand-In #2

Answers must be handed in via Absalon no later than 23:59 on Sunday November 14. Your answers must consist of a (zipped) folder with a report (pdf- or Word-format; remember to put your name on it) that can be read on a stand-alone basis, as well your code (Matlab, R, C++, Python, or other) in easy-to-run format (subfolders).

Simulation

S.1 Replicate figures 2-4 in Frandsen, Pedersen & Poulsen (2021). Supplementary information about figure 4: #simulations on the x-axis denotes the number paths (each with different S_0) used to estimate the $\Delta(\cdot,\cdot)$ function. #repetitions denotes the number of paths (with a fixed $S_0 = 1$) used to calculate the standard deviation of the hedging error). Although not stated in the paper, batching has been applied to the point estimates shown in the graph. That means that each point in the graphs is actually an average of the same experiment repeated 10 times with different random numbers. You should also do batching to minimize the MC error.

S.2 Draw the same graphs but use the Black-Scholes model instead (with the same σ) and comment on any potential differences between the quality of the approximation in the two models.

S.3 In Frandsen, Pedersen & Poulsen (2021), the Delta labels D_j are generated using the Pathwise method. But the LRM method (see MC slides p. 62) provides an alternative way to generate a random variable whose expectation is equal to the Delta of an option. Use this method instead of the Pathwise method (still using the Black-Scholes model), to generate the $D_j's$ in Frandsen, Pedersen & Poulsen (2021). Draw a graph similar to that of figure 4, where you compare the quality of the Pathwise and LRM Delta regularization. (You don't need to do it for all polynomial orders, just 2 and 8 is fine) Comment and explain the potential causes of the difference between the two methods.

S.4 Consider now a digital option. Explain why the pathwise method cannot be used and use the LRM method instead to draw graphs comparable to figures 2-4. Do so for the same Black-Scholes model as in question S.2. Why do think the errors are much higher than for the call option?

S.5 Do the same as in the previous question but apply the Mixed method estimator (see MC slides p. 69) with an appropriately chosen ϵ. Comment on the potential differences between the pure LRM and the Mixed method.

Finite Difference

The figure below shows Table 1 from the famous paper Longstaff & Schwartz (2001)

Table 1

S	σ	T	Finite difference American	Closed form European	Early exercise Value	Simulated American	(s.e.)	Closed form European	Early exercise value	Difference in early exercise value
36	.20	1	4.478	3.844	.634	4.472	(.010)	3.844	.628	.006
36	.20	2	4.840	3.763	1.077	4.821	(.012)	3.763	1.058	.019
36	.40	1	7.101	6.711	.390	7.091	(.020)	6.711	.380	.010
36	.40	2	8.508	7.700	.808	8.488	(.024)	7.700	.788	.020
38	.20	1	3.250	2.852	.398	3.244	(.009)	2.852	.392	.006
38	.20	2	3.745	2.991	.754	3.735	(.011)	2.991	.744	.010
38	.40	1	6.148	5.834	.314	6.139	(.019)	5.834	.305	.009
38	.40	2	7.670	6.979	.691	7.669	(.022)	6.979	.690	.001
40	.20	1	2.314	2.066	.248	2.313	(.009)	2.066	.247	.001
40	.20	2	2.885	2.356	.529	2.879	(.010)	2.356	.523	.006
40	.40	1	5.312	5.060	.252	5.308	(.018)	5.060	.248	.004
40	.40	2	6.920	6.326	.594	6.921	(.022)	6.326	.595	−.001
42	.20	1	1.617	1.465	.152	1.617	(.007)	1.465	.152	.000
42	.20	2	2.212	1.841	.371	2.206	(.010)	1.841	.365	.006
42	.40	1	4.582	4.379	.203	4.588	(.017)	4.379	.209	−.006
42	.40	2	6.248	5.736	.512	6.243	(.021)	5.736	.507	.005
44	.20	1	1.110	1.017	.093	1.118	(.007)	1.017	.101	−.008
44	.20	2	1.690	1.429	.261	1.675	(.009)	1.429	.246	.015
44	.40	1	3.948	3.783	.165	3.957	(.017)	3.783	.174	−.009
44	.40	2	5.647	5.202	.445	5.622	(.021)	5.202	.420	.025

Comparison of the finite difference and simulation values for the early exercise option in an American-style put option on a share of stock, where the option is exercisable 50 times per year. The early exercise value is the difference between the American and European put values. In this comparison, the strike price of the put is 40, the short-term interest rate is .06, and the underlying stock price S, the volatility of returns σ, and the number of years until the final expiration of the option T are as indicated. The European option values are based on the closed-form Black–Scholes formula. The simulation is based on 100,000 (50,000 plus 50,000 antithetic) paths for the stock-price process. The standard errors of the simulation estimates (s.e.) are given in parentheses.

FD.1 Replicate the numbers in column 5 in the table using a *suitable* implementation of a *suitable* finite difference method. You may find Jesper Andreasen's slide sets 1 and 3 as well as this old project useful.

FD.2 Suppose you want to simulate your way to European (not American) option prices of *similar accuracy* as you get when using a finite difference method from the first part. What would run-times look like? It seems reasonable to define accuracy for a finite difference method via Østerby's equations (10.9), (10.12), and (10.30) and to use the standard errors for simulated values.

FD.3 Replicate the American option prices in column 4 using Crank-Nicolson. What is the global convergence order (as analyzed in Østerby, chapter 10)? Does that surprise you?

FD.4 Explain how (stock price, strike)-homogeneity and time-homogeneity of call and put option prices in the Black-Scholes model can be used to price a lot of different options from a single finite difference grid. (You don't need to actually implement this.)

FD.5 If we go beyond basic Black-Scholes, the previous homogeneity tricks do not work. But something else does (for European options, not American):

1. Look up what Dupire's forward equation is (eg. here, equation (4)).
2. Adjust your finite difference code to solve Dupire's forward equation for European puts and calls.
3. Using the constant elasticity of variance model as an example, demonstrate that your code can now calculate many option prices from a single finite difference grid.