QF4102 Financial Modelling and Computation: Assignment 2 Report

G04: Dong Shaocong, He Xinyi

October 25, 2017

Abstract

This document is the report for QF4102 Financial Modelling and Computation, assignment 2. In this report, we will show all the numerical results we got, and comment on them with respect to the results and questions respectively. In general, in this assignment, we examined the Forward Shooting Grid method (FSGM) for American put arithmetic and look back options. We also looked at explicit numerical scheme and its convergence problem both for American and European vanilla call options.

1 A2.1(i) American fixed-strike arithmetic-average put option

1.1 Description of work done

This is the first question of assignment 2. We considered firstly a not newly issued American style fixed-strike arithmetic-average put option using Forward shooting grid method. We firstly wrote a Matlab function by using two state $FSGM$ algorithm with a slight modification that we firstly calculate a current running average and create the average grid based on this value. After writing out the function, we tried it out with different values of ρ and different values of number of time periods. We not only recorded the numerical value we got, we also recorded the time amount taken for each set of calculations.

1.2 Numerical results we get:

ρ, N	50	100	200	400
1	5.3765	5.4228	5.4515	5.4665
0.5	5.3435	5.411	5.445	5.462
0.2	5.3371	5.4045	5.4408	5.4597

Table 1: Option values with different values of ρ and N

ρ, N	50	100	200	400
1	0.088266	0.544189	4.532513	38.266119
0.5	0.172278	1.154307	9.133832	78.389066
0.2	0.361367	2.898972	24.377280	193.867936

Table 2: Calculation time spent (in seconds) with different values of ρ and N

1.3 Comment on your numerical results and computation times taken.

blah.

2 A2.1(ii) American fixed-strike lookback put option

2.1 Description of work done

This is the second question of assignment 2. We considered a not newly issued American style fixed-strike lookback put option using Forward shooting grid method. We again wrote a Matlab function by using two state $FSGM$ algorithm. Compared to the last question, we also changed the algorithm for running minimum calculation. Because of the running minimum will surely be one of the price states of the underlier, we can choose $\rho = 1$. After writing the Matlab function, we tried it out with different number of time periods in the lattice being 50 to 500 in increments of 50. The first set of value is done by taking running minimum equal to 0.97 and for the second round, we choose the running minimum from the previous time periods equal to 0.57.

2.2 Taking previous running minimum as 0.97

2.2.1 Numerical results

table	50	100	150	200	250
value	0.09621	0.10479	0.10262	0.098652	0.099186
time	0.019038	0.125343	0.414005	0.969385	1.895254

Table 3: Option value and time spent with different values of N from 50 to 250

table	300	350	400	450	500
value	0.10038	0.10081	0.094152	0.094508	0.096121
time	3.354355	5.378967	8.073453	11.306953	15.598436

Table 4: Option value and time spent with different values of N from 300 to 500

2.2.2 Comment on your numerical results and computation times taken.

comments.

2.3 Taking previous running minimum as 0.57

2.3.1 Numerical results

table	50	100	150	200	250
value	0.38	0.38	0.38	0.38	0.38
time	0.021754	0.125839	0.404128	0.969833	1.936496

Table 5: Option value and time spent with different values of N from 50 to 250

table	300	350	400	450	500
value	0.38	0.38	0.38	0.38	0.38
time	3.349201	5.361782	7.911092	11.201771	15.721890

Table 6: Option value and time spent with different values of N from 300 to 500

2.3.2 Comment on your numerical results and computation times taken.

comments.

3 Explicit difference scheme III

3.1 Description of work done

This is the second part of assignment 2. In this question, we examined the third explicit difference scheme (III) discussed in lecture. We firstly wrote a Matlab function using explicit scheme III for pricing European vanilla call option. We have identified the coefficients in the finite difference equations that are not positive. We also compared it against the exact value we should get. Next, we tried to derive a a lower bound for $N = \dfrac{T}{\Delta t}$ such that all coefficients in the finite difference equations are positive. rerunning the algorithm with this bound and a slightly lower value of N, we determined the cut-off value where the option estimate loses all its significant figures. In addition, we wrote another Matlab function for pricing American vanilla call options using explicit scheme III algorithm. Using this new function, we redid the previous experiments.

3.2 (i) Compare the estimate against the exact value of option.

At $S_0 = 9.8$, from the output, we get the exact value which is equal to 0.84688 but the option value estimate we get from this explicit scheme is $7.055274186430886 \times 10^{21}$. This estimation clearly loses all the significant figures and it's not acceptable as an answer for option value. Checking the conditions violation, we get the following:

Coeff a, Of 719 elements, 0 violated the positivity condition.
Coeff b, Of 719 elements, 653 violated the positivity condition.
Coeff c, Of 719 elements, 0 violated the positivity condition.

Clearly, the coefficient b has positivity condition violation thus making the scheme not convergent.

3.3 (ii) A lower bound for N such to have all positive coefficients.

As we can see from the output as well as the explicit scheme formulation, only the parameter b can be negative sometimes which may affect the convergence of this scheme, which may thus in turn affect the precision of the final estimate for the option value. We can obtain the lower bound for N such to have all positive coefficients by getting the lowest N to make b positive for every i.

$$1 - \sigma^2 \times i^2 \times \Delta t - (r - q) \times i \times \Delta t > 0 \tag{1}$$

Substituting the parameters given into equation 1 we can get:

$$0.0225 \times i^2 \times \Delta t - 0.009 \times i \times \Delta t < 1 \tag{2}$$

$$\Delta t = \frac{T}{N} \tag{3}$$

From equation 2 and 3 we can have:

$$N > 0.0225 \times i^2 \times 0.25 - 0.009 \times i \times 0.25 \tag{4}$$

As we have the biggest i given as follows:

$$i_{max} = \frac{4 \times X}{\Delta S} \tag{5}$$

Substituting equation 5 into equation 4, we can have $N_{min} = 2914.38 \approx 2915$.

3.4 (iii) Obtain another estimate to the option value using this N.

At $S_0 = 9.8$, the exact value of the option is equal to 0.84688 and the estimate of this option value we get from explicit scheme is 0.82529. which is very close. Checking the condition violations we have:

```
Coeff a, Of 719 elements, 0 violated the positivity condition.
Coeff b, Of 719 elements, 0 violated the positivity condition.
Coeff c, Of 719 elements, 0 violated the positivity condition.
```

We can see no coefficients is violating positivity conditions.

3.5 (iv) Determine the cut-off value where the option estimate loses all its significant figures.

This is the output from our function

```
——question iv—— Finding cut-off value Euro
Calculating...
At N = 205, option value is 0.92474, it loses all significant figures.
```

As we can see, the cut-off value is 205 for N, where we have estimation of the option value equal to 0.9274, which clearly loses all the significant figures.

3.6 (v) Repeat iii for American vanilla call option pricing function.

At $S_0 = 9.8$, we have the exact value for European call vanilla option is equal to 0.84688, but the estimation for American vanilla call option is equal to 0.87583, which is slightly bigger than the European vanilla call option.