

# QF4102 Financial Modelling and Computation: Assignment 2

## Report

G04: Dong Shaocong, He Xinyi

October 24, 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  $\rho$  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:**

Numerical results go here.

### **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 Numerical results for taking previous running minimum as 0.97**

Numerical results go here.

### **2.3 Comment on your numerical results and computation times taken.**

comments.

### **2.4 Numerical results for taking previous running minimum as 0.57**

Numerical results go here.

### **2.5 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 lower bound for  $N = \frac{T}{\Delta t}$  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.

Blah.

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

Mathematical derivations go here.

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

Numerical results and discussion.

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

Numerical results and discussion.

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

Numerical results and discussion.