CS 476/676 Assignment 4

Winter 2023

Instructor: Yuying Li	Office: DC3623	*yuying@uwaterloo.ca*
Lecture Times:	MW 11:30-12:50	MC 2035
Yuying Li:	OH: Thurs 3-4pm	
TA: Chendi Ni	DC 3594 (SciCom)	chendi.ni@uwaterloo.ca
Chendi Ni	OH: Mar 22, Wednesday, 2-3pm	DC 3594
Chendi Ni	OH: Mar 31, Friday, 2-3pm	DC 3594
Chendi Ni	OH: Apr 6, Thursday, 4-5pm	DC 3594

My Web Site: http://www.cs.uwaterloo.ca/~yuying

Due: 11am April 10, 2023

Note on Programming Questions: Most of the marks for programming questions are allocated for explanations of algorithms and explanation of results. If all you hand in is the listing of the "Raw Code" or "Raw Output" by itself, you will get poor marks. All coding should be done in well documented (commented) Matlab. By default, you should submit listings of all Matlab code used in your assignment. Please ensure your code is well commented

1. [(14 marks)] (Finite Difference, constant timesteps, European)

Using a finite difference method as discussed in lectures, develop a Matlab code for pricing European options under a generalized Black Scholes local volatility function model (under risk neutral probability)

$$\frac{dS_t}{S_t} = r dt + \sigma(S,t) dZ_t. \tag{1}$$

In the implementation, use constant timestep sizes and forward/backward/central differencing as appropriate to ensure a positive coefficient discretization. Your code should be able to use fully implicit, Crank-Nicolson, and CN-Rannacher timestepping. For efficiency, use the Matlab sparse matrix function **spdiags** to set up a sparse matrix of the correct size and use **lu** Matlab function to solve linear systems. Avoid unnecessary LU factorization computation, if it is possible.

Since we are interested in the solution near $S^0 = S(0)$, set up the underlying price grid as below (a matlab assigment):

$$\begin{aligned} S = [\quad & 0 : 0.1{*}S^0 : 0.4{*}S^0, \ldots \\ & 0.45{*}S^0 : 0.05{*}S^0 : 0.8{*}S^0, \ldots \\ & 0.82{*}S^0 : 0.02{*}S^0 : 0.9 * S^0, \ldots \\ & 0.91{*}S^0 : 0.01{*}S^0 : 1.1{*}S^0, \ldots \\ & 1.12{*}S^0 : 0.02{*}S^0 : 1.2{*}S^0, \ldots \\ & 1.25{*}S^0 : .05{*}S^0 : 1.6{*}S^0, \ldots \\ & 1.7{*}S^0 : 0.1{*}S^0 : 2{*}S^0, \ldots \\ & 2.2{*}S^0, 2.4{*}S^0, 2.8{*}S^0, \ldots \\ & 3.6{*}S^0, 5{*}S^0, 7.5{*}S^0, 10{*}S^0 \ ], \end{aligned} \tag{2}$$

You will also investigate convergence rate by analyzing the ratio below computationally

$$\frac{V(h) - V(h/2)}{V(h/2) - V(h/4)} \tag{3}$$

where

$$\begin{aligned} h &= C_1 \cdot \Delta S \\ h &= C_2 \cdot \Delta \tau \end{aligned}$$

for some constants C_1, C_2. Assume that the pricing error obeys the following theoretical result:

$$\text{Error} \quad = \quad O((\Delta\tau)^2, (\Delta S)^2), \quad \text{where } \Delta S = \max_i (S_{i+1} - S_i) \tag{4}$$

Then the solution on each grid (at a given point) has the form

$$\begin{aligned} V(h) &= V_{exact} + A \cdot h^2 \\ V(h/2) &= V_{exact} + A \cdot (h/2)^2 \\ V(h/4) &= V_{exact} + A \cdot (h/4)^2 \end{aligned} \tag{5}$$

where we have assumed that the mesh size and timestep are small enough that the coefficient A in equation (5) is approximately constant. Now, equation (5) implies that

$$\frac{V(h) - V(h/2)}{V(h/2) - V(h/4)} \simeq 4 \tag{6}$$

Carry out a computational convergence study by solving the pricing problem on a sequence of grids as follows. Begin with a timestep of $\Delta\tau = T/25$. Subsequently, each grid has twice as many intervals as the previous grid (new nodes inserted halfway between the coarse grid nodes) and the timestep size is halved.

(a) (10 marks) Assume that the local volatility function is

$$\sigma(S, t) = \frac{\alpha}{\sqrt{S}}, \quad \alpha > 0 \text{ is a constant.} \tag{7}$$

where $\alpha > 0$ is a constant. First, write matlab code to price European Butterfly Spreads, whose payoff is defined by strike prices $K_1 < K_2 < K_3$, where $K_2 = \frac{1}{2}(K_1 + K_3)$, as follows:

$$\text{payoff}(S_T) = \begin{cases} 0 & \text{if } S_T \leq K_1 \\ S_T - K_1 & \text{if } K_1 < S_T \leq K_2 \\ K_3 - S_T & \text{if } K_2 < S_T \leq K_3 \\ 0 & \text{if } S_T > K_3 \end{cases}$$

using the data given in the Table 1. Carry out the above tests using fully implicit, Crank Nicolson, and CN-Rannacher timestepping. Show the option value at $t = 0, S = S(0)$. Show a convergence table for each test, with a series of grids (with 4 refinements, and the table should be formatted like Table 20.1 on page 165 in the course notes). Are your observations consistent with the theory regarding the rate of convergence? Explain

(b) (4 marks) In one plot, graph of the option value at $t = \frac{T}{4}$ and $t = \frac{3T}{4}$ for the range $S = [0.5S(0), 1.5S(0)]$ respectively for your solution on the finest grid for CN-Rannacher timestepping. Similarly, make a plot for delta and gamma. What's your observations on the change of the option value, delta, and gamma with respect to the underlying price S and time t?

(c) (bonus, 3 points) How do you achieve convergence ratio of 4 for CN-Rannacher? Describe your change and show convergence table.

. Submit your matlab code, plots, tables and discussion

.

2. [(14 marks)] (Finite Difference, variable time step, American option)

Modify your European pricing code to implement the penalty method to price the American butterfly spread option with the option payoff specified in the previous question. Setting $K_3 = 10^7$, carry out a convergence study assuming the local volatility model assumed in Question 1 and other data in Table 1. Use CN-Rannacher with both constant and variable timesteppings described in the course notes. For the variable timestepping, use $dnorm = .1$ and an initial timestep of $\Delta\tau = T/25$. On each grid refinement, reduce the initial timestep by a factor of 4 and reduce $dnorm$ by a factor of 1/2. Be sure that your timestep selector stops at the pricing code at $t = T$ exactly.

Table 1: Data

α	2.5
r	.05
Time to expiry (T)	1 years
Initial asset price $S(0)$	\$95
Strike Price K_1	$0.9S(0)$
Strike Price K_3	$1.1S(0)$

Table 2: Implied Volatility Surface

	strike ($\%S_0$)				
T	90%	95%	100%	105%	110%
0.425	.155	.138	.125	.109	.103
0.695	.157	.144	.133	.118	.104
1	.159	.149	.137	.127	.113

(a) (8 marks) Show the convergence table. What standard option does butterfly spread become when $K_3 \to +\infty$?

(b) (4 marks) In one plot, graph the European and American butterfly spread option values at $t = \frac{T}{2}$ for the finest grid within the range $S = [0.5S(0), 1.5S(0)]$ respectively for CN-Rannacher variable timestepping. On the same plot, also plot the payoff function. How does values for the European butterfly spread option differ from that of American option? Can it be optimal to exercise the butterfly spread option at $\frac{T}{2}$? If yes, what is the region of the stock price in which it is optimal to exercise? If not, explain.

(c) (2 marks) Make a plot for delta and gamma at $t = \frac{T}{2}$ for $S \in [0.5S(0), 1.5S(0)]$ using variable timestepping.

Submit your matlab code, plots, tables, and discussion.

3. [(14 marks)] (Model Calibration)

A set of implied volatilities of European call options on an underlying on a given day are given in Table 2. You are to calibrate a local volatility function model (1) from implied volatilities in Table 2.

(a) (2 marks) Graph the implied volatility surface and corresponding option value surface against strike K and expiry T.

(b) (3 marks) Assume that $r = 0.03$, and the underlying price follows a local volatility model

$$\frac{dS_t}{S_t} = rdt + \sigma(S_t)dZ_t$$

and we do not know the local volatility function $\sigma(S_t)$. Assume that the local volatility function is represented by the following simple feed forward neural network, with input feature S and specification below:

$$\sigma(S) = \frac{1}{1 + e^{w_1 y_1 + w_2 y_2}}, \quad y_i = u_i + v_i \times S, \quad i = 1, 2 \tag{8}$$

Here $u_i, v_i, w_i, i = 1, 2$, are unknown weights. Modify your Matlab code in Question 1 so that it now computes efficiently option values for the above volatility function model.

(c) (3 marks) Assume that the market price $V_0^{\text{mkt}}(K_j, T_j), j = 1, 2, \ldots, m$, are given, where (K_j, T_j) denotes the strike and expiry of the j-th option. Assume that $V_0(K_j, T_j; x)$ denotes the initial option value of a

European call option with strike K_j and T_j under a local volatility model described by a set of parameters $x = (w_1, w_2, u_1, u_2, v_1, v_2)$. Assume $S_0 = 100$.

Determine the model which best fits the market option prices by solving the following *nonlinear least squares problem*

$$\min_{x \in \Re^n} \frac{1}{2} \sum_{j=1}^{m} \left(V_0(K_j, T_j; x) - V_0^{\mathrm{mkt}}(K_j, T_j) \right)^2 \tag{9}$$

Write a matlab function to return the residual vector F, $F_j = V_0(K_j, T_j; x) - V_0^{\mathrm{mkt}}(K_j, T_j)$, and its m-by-n Jacobian matrix J, if necessary, for any given x, as described in the lecture notes. Your Matlab function must return, in the second output argument, the Jacobian matrix J at x. Note that, by checking the value of **nargout**, you can avoid computing J when your Matlab function is called with only one output argument (in the case where the optimization algorithm only needs the value of F but not J).

```
function [F,J] = myfun(x)
F = ...                 % Objective function values at x
if nargout > 1          % Two output arguments
    J = ...             % Jacobian of the function evaluated at x
End
```

Compute J using finite difference approximation as described in class.

(d) (6 marks) Use Matlab function **lsqnonlin** with LevenbergMarquardt as the choice of the optimization method to estimate the unknown coefficients x for the LVF model (8), using the option prices calculated from the implied volatility in Table 2 by **blsprice** as the market prices $\{V_0^{\mathrm{mkt}}(K_j, T_j)\}$.

You can set the options for optimization as follows

options = optimset('Jacobian', 'on', 'Algorithm', 'levenberg-marquardt', 'Display', 'iter', 'MaxIter', 50);

Perform the volatility calibration computation. Assume the starting values for the optimization are $u_i = w_i = 0, v_i = -0.0001$, $i = 1, 2$. Report the estimated optimal parameter $(w_1^*, w_2^*, u_1^*, u_2^*, v_1^*, v_2^*)$ and the calibration error (objective function value). Plot the implied volatilities from the true model and and the implied volatilities from the computed $(w_1^*, w_2^*, u_1^*, u_2^*, v_1^*, v_2^*)$ for the given options. Comment on your observations.

4. [(12 marks)] (**Optimal Static Hedging**)

Let $\mu, , \sigma, r$ be given constants. Consider a stock with the price S_t follows

$$\frac{dS_t}{S_t} = \mu \; dt + \sigma \; dZ_t, \quad dZ_t : \text{ increment of a standard Brownian motion}$$

Consider a European butterfly spread in Question 3 which pays at the expiry T. Using Monte Carlo simulations, you are to formulate an optimization problem to determine the optimal static hedging portfolio Π of the underlying and bond. An optimal static hedging portfolio can be constructed today and hold it until the expiry. At $t = 0$, let $x = (x_1, x_2)$, where x_1, x_2 denote units of the underlying and and the bond to hold. Let $c_1 = S(0)$ denote the initial stock price, $c_2 = 1$ the initial bond price, and the column vector $c = (c_1, c_2)$. Using M simulations of the underlying price at $t_n = n \cdot \left(\frac{T}{N}\right), n = 0, 1, \ldots, N$, the optimal static hedging portfolio with a fixed initial cost of c^0 can be determined by solving the following optimization problem

$$\begin{array}{ll} \displaystyle\min_{x=(x_1,x_2)} & \displaystyle\frac{1}{M} \|Ax - b\|_2^2 \\ \text{subject to} & c^T x = V_0^B \end{array} \tag{10}$$

where V_0^B is the initial butterfly spread price, which you can assume that it is given. $\frac{1}{M}\|Ax - b\|_2^2$ approximates the expected quadratic error $\mathbf{E}((\Pi_T - \text{payoff})^2)$, and Π_T is the value of the portfolio of the underlying and call option at T.

(a) (3 marks) Provide explicit forms for A, b, which are in terms of prices of underlyng and bond at time $t = 0$ and T, as well as the butterfly payoff. Explain.

(b) (5 marks) Write a pseudo code to compute A, b, c.

(c) (4 marks) Let x^* be the optimal solution of (10) and ν^* be the Lagrangian multiplier associated with the constraint. Provide mathematical equations for computing x^* and ν^*.

5. [(10 marks)] (CVaR and Convexity)

Assume that a random loss L has a continuous distribution with the density $p(l) > 0$, for $l \in (-\infty, +\infty)$. Suppose that a confidence level $\beta, 0 < \beta < 1$, is given. Consider the following function

$$f(\alpha) = \mathbf{E}\Big(\max(L-\alpha, 0)\Big)$$

(a) (3 marks) Show that $f(\alpha)$ is continuously differentiable with

$$f'(\alpha) = -\left(\int_{\alpha}^{\infty} p(l)dl\right)$$

Hint: assuming $a(x), g(x,y)$ are continuously differentiable, Leibniz's rule states that

$$\frac{d}{dx}\left(\int_{a(x)}^{+\infty} g(x,y)dy\right) = -g(x, a(x)) \cdot \frac{d}{dx}a(x) + \int_{a(x)}^{+\infty} \frac{\partial}{\partial x} g(x,y)dy$$

(b) (3 marks) What is $f''(\alpha)$? Is $f(\alpha)$ convex? Explain.

(c) (4 marks) Let $0 < \beta < 1$ be given. Is the following optimization problem convex? Explain.

$$\min_{\alpha} \alpha + \frac{1}{1-\beta}\mathbf{E}\Big(\max(L-\alpha, 0)\Big) \tag{11}$$

Assume that α^* is the solution to (11). Show that α^* is the VaR with confidence β.

6. [(Graduate Student Question)] (10 marks).

Markowtiz's mean variance portfolio optimization is typically used to allocate stocks. When the asset returns are not normal, the standard deviation is not an appropriate risk measure. Assume that n risky assets returns are $\mathbf{r}_1, \cdots, \mathbf{r}_n$. Consider the following CVaR risk constrained allocation problem:

$$\begin{array}{ll} \max\limits_{x_1,\cdots,x_n} & \mathbf{E}\left(\sum\limits_{i=1}^{n} x_i \mathbf{r}_i\right) \\ \text{subject to} & \sum\limits_{i=1}^{n} x_i = 1, \quad x \geq 0 \\ & \mathrm{CVaR}_{\beta}\left(-\sum\limits_{i=1}^{n} x_i \mathbf{r}_i\right) \leq \rho \end{array} \tag{12}$$

where x_i is the portfolio weight of the asset i, and $\mathbf{r}_i$ is the (random) rate of return of asset i.

Assume that we have M independent samples of the asset returns, the above CVaR minimization can be approximated by the simulation CVaR problem below :

$$\begin{array}{ll} & \max_{(x,y,\alpha)} \ \sum_{i=1}^{n} x_i \bar{r}_i \\ & \sum_{i}^{n} x_i = 1, \ \ x_i \geq 0 \\ \text{subject to} & \alpha + \frac{1}{M(1-\beta)} \sum_{j=1}^{M} y_j \leq \rho \\ & y_j \geq -\sum_{i=1}^{n} r_i^j x_i - \alpha, \quad j = 1, \cdots, M \\ & y_j \geq 0, \ j = 1, \cdots, M \end{array} \tag{13}$$

Table 3: Data CVaR Optimization

α^A	0.85
α^B	0.9
r	.03
Initial asset price $S^A(0)$	\$20
Initial asset price $S^B(0)$	\$25
Strike Price K_1^A	$0.9S^A(0)$
Strike Price K_3^A	$1.1S^A(0)$
Strike Price K_1^B	$0.9S^B(0)$
Strike Price K_3^B	$1.1S^B(0)$

where r_i^j denote the jth sample of the return r_i, $\bar{r}_i$ is the sample average of $\mathbf{r}_i$, and ρ is an upper bound on the CVaR risk.

Assume that two stocks A and B, whose price S^A and S^B follow a LVF model (1) with the local volatility function

$$\sigma(S,t) = \frac{\alpha}{\sqrt{S}}, \quad \alpha > 0 \text{ is a constant.} \tag{14}$$

where the α value for stock A and B are given in Table 3.

European butterfly spread has the payoff

$$\text{payoff}(S_T) = \begin{cases} 0 & \text{if } S_T \le K_1 \\ S_T - K_1 & \text{if } K_1 < S_T \le K_2 \\ K_3 - S_T & \text{if } K_2 < S_T \le K_3 \\ 0 & \text{if } S_T > K_3 \end{cases} \tag{15}$$

where strike prices $K_1 < K_2 < K_3$, where $K_2 = \frac{1}{2}(K_1 + K_3)$.

(a) Assume that a trading desk currently has a portfolio consisting of 2 stocks A, B, and the following options on stock A and stock B respectively as specified below:

- Stock A: call with strikes $S^A(0), 1.1S^A(0), 1.2S^A(0)$ and expiry $T = 0.5$ and a European butterfly spread with payoff European butterfly spread with payoff (15), K_1^A, K_3^A in Table 3, and expiry $T = 2/3$.
- Stock B: put with strikes $0.8S^B(0), 0.9S^B(0), S^B(0)$ and $T = 1$ year and European butterfly spread option with payoff (15) with K_1^B, K_3^B in Table 3, and expiry $T = 1$.

Modify your code in question 1 and a Matlab code to efficiently generate 10000 1-month return scenarios for stock A, stock B, and the specified options. Plot the histogram of 1-month return of equally weighted portfolio using these instruments.

(b) For $\beta = 80\%, 85\%, 90\%, 95\%$ respectively, compute optimal portfolios using Matlab **linprog** based on (13) with ρ setting to CVaR_β of the constant proportion allocation. Tabulate and compare the mean return and CVaR values of the equal allocation and optimal portfolio (13). Comment on your observations on performance on the equal allocation and the optimal portfolio.