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

Using a finite difference method as discussed in class, develop a Matlab code for pricing European options under a generalized Black Scholes local volatility function model

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

Use forward/backward/central finite difference as appropriate to ensure a positive coefficient discretization. Use constant timestep sizes. 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. When appropriate, avoid unnecessary LU factorization computation. Begin with a timestep of \( \Delta \tau = T/25 \), and the grid below

\[
S = [0:0.1*K:0.4*K, ... \\
0.45*K:0.05*K:0.8*K, ... \\
0.82*K:0.02*K:0.9*K, ... \\
0.91*K:0.01*K:1.1*K, ... \\
1.12*K:0.02*K:1.2*K, ... \\
1.25*K:0.05*K:1.6*K, ... \\
1.7*K:0.1*K:2*K, ... \\
2.2*K, 2.4*K, 2.8*K, ... \\
3.6*K, 5*K, 7.5*K, 10*K];
\]

where \( K \) is the strike, and we are interested in the solution near \( S = K \).

Assume that the pricing error obeys the following theoretical result:

\[
\text{Error} = O((\Delta \tau)^2,(\Delta S)^2); \quad \Delta S = \max_i (S_{i+1} - S_i)
\]  
(2)

Let

\[
h = C_1 \cdot \Delta S
\]

\[
h = C_2 \cdot \Delta \tau
\]
We can label each computation according to the $h$ value. Then the solution on each grid (at a given point) has the form

\[
\begin{align*}
V(h) &= V_{\text{exact}} + A \cdot h^2 \\
V(h/2) &= V_{\text{exact}} + A \cdot (h/2)^2 \\
V(h/4) &= V_{\text{exact}} + A \cdot (h/4)^2
\end{align*}
\]

(3)

where we have assumed that the mesh size and timestep are small enough that the coefficient $A$ in equation (3) is approximately constant. Now, equation (3) implies that

\[
\frac{V(h) - V(h/2)}{V(h/2) - V(h/4)} \simeq 4
\]

(4)

To carry out a computational convergence study, you should solve the pricing problem on a sequence of grids. 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) Assume the following local volatility function

\[
\sigma(S, t) = \frac{\alpha}{\sqrt{St}}, \quad \alpha > 0 \text{ is a constant.}
\]

First, write a matlab code to price an European straddle with the payoff

\[
\max(K - S, S - K)
\]

using the data given in the Table 1. Carry out above tests using fully implicit, Crank Nicolson, and CN-Rannacher timestepping. Show the option value at $t = 0$, $S = 100$. Show a convergence table for each test, with a series of grids (the table should be formatted like Table 20.1 on page 165 in the course notes). Check the theory by examining the rate of convergence of your pricing code.

(b) Show plots of the option value, delta, gamma for the range $S = [50, 150]$, for your solution on the finest grid for CN-Rannacher timestepping.

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

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

Modify your code to use the penalty method for an American straddle option with payoff(S)=$\max(K - S, S - K)$. Carry out a convergence study with a constant volatility $\sigma = 0.15$ (which is different from 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.

- Show the convergence table.
- Show plots of the price and delta respectively, for the finest grid using CN-Rannacher timestepping. Explain what you see.

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

3. (14 marks) (Model Calibration)

A set of implied volatilities for S&P 500 index European call options on a given day is given in Table 2. You are to calibrate a local volatility function model (1) from implied volatilities in Table 2. Assume that $S(0) = 100$ and $r = 0.03$.

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

Table 1: Data for Put Example

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>1.5</td>
</tr>
<tr>
<td>r</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to expiry (T)</td>
<td>1 years</td>
</tr>
<tr>
<td>Strike Price</td>
<td>$100</td>
</tr>
<tr>
<td>Initial asset price $S^0$</td>
<td>$100</td>
</tr>
</tbody>
</table>

Table 2: Implied Volatility Surface

<table>
<thead>
<tr>
<th></th>
<th>strike (%$S_0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>90%</td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>0.425</td>
<td>.155</td>
</tr>
<tr>
<td>0.695</td>
<td>.157</td>
</tr>
<tr>
<td>0.94</td>
<td>.159</td>
</tr>
</tbody>
</table>

(b) Now assume that the underlying price follows a local volatility model
\[
\frac{dS_t}{S_t} = \mu dt + \sigma(S_t, t)dZ_t
\]
where the local volatility function is represented by the following quadratic formula:
\[
\sigma(S, t) = \max(x_1 + x_2 \cdot S + x_3 \cdot t + x_4 \cdot S^2 + x_5 \cdot t^2 + x_6 \cdot S \cdot t, 0)
\] (5)
and \(x = (x_1, x_2, x_3, x_4, x_5, x_6)\) is the vector of unknown weights.

Modify your Matlab code in Question 3 so that it now computes efficiently option values for the above volatility function model.

(c) Assume that the market price \(V_{mkt}^0(K_j, T_j), j = 1, 2, \ldots, m\), are given, where \((K_j, T_j)\) denotes the strike and expiry of the \(j\)-th option. Assume that \(V_0^0(K_j, T_j; x)\) denotes the price of an option with strike \(K_j\) and \(T_j\) under the local volatility model (5).

Determine the model which best fits the market option prices by solving the following nonlinear least squares problem
\[
\min_{x \in \mathbb{R}^n} \frac{1}{2} \sum_{j=1}^{m} (V_0^0(K_j, T_j; x) - V_{mkt}^0(K_j, T_j))^2
\] (6)

Write a matlab function which returns the residual vector \(F\), \(F_j = V_0^0(K_j, T_j; x) - V_{mkt}^0(K_j, T_j)\) and its \(m\)-by-\(n\) Jacobian matrix \(J\), if necessary, for any given \(x\), as described in the lecture notes. Note that, by checking the value of \text{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 \(J\)).

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

Compute \(J\) using finite difference approximation as described in class.

(d) Use Matlab function \text{lsqnonlin} with LevenbergMarquardt as the choice of the optimization method to estimate the unknown coefficients \(x\) for the LVF model (5), using the set option prices \(\{V_{mkt}^0(K_j, T_j)\}\) in Table 2 as the market prices. 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 using the starting point \( x = [0.12, 0.0, 0.0, 0.0, 0.0, 0.0] \). Report the estimated parameter set \( x^* \) and the calibration error \( \|F(x^*)\|^2 \). Graph the estimated local volatility function \( \sigma(S, t) \). Plot the surface of the market implied volatilities from Table 2 and the corresponding surface of the implied volatility from the calibrated model based on \( x^* \). Comparing the two plots, when does the calibrated model yield the largest error in implied volatilities of given options?

4. (10 marks) (CVaR and Convexity)

Assume that the random loss \( L \), with the support in \((-\infty, +\infty)\), has a continuous distribution with the density \( p(l) > 0 \). Assume that the confidence level \( 0 < \beta < 1 \). Consider the following function \( F(\alpha) = \alpha + \frac{1}{1-\beta} \mathbb{E}([L-\alpha]^+) \) where \( 0 < \beta < 1 \) and

\[
[L-\alpha]^+ = \begin{cases} 
L-\alpha & \text{if } L > \alpha \\
0 & \text{otherwise}
\end{cases}
\]

Assuming \( a(x), g(x,y) \) are continuously differentiable, Leibniz’s rule states that

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

(a) Show that \( F(\alpha) \) is continuously differentiable with

\[
F'(\alpha) = \frac{1}{1-\beta} \left( \int_{-\infty}^{\alpha} p(l)dl - \beta \right)
\]

(b) Prove that \( F(\alpha) \) is strictly convex by establishing that

\[
F(\bar{\alpha}) > F(\alpha) + F'(\alpha)(\bar{\alpha} - \alpha), \quad \forall \alpha, \bar{\alpha}.
\]

Hint: what is \( F''(\alpha) \)?

(c) Let \( \alpha^* \) be the minimizer of \( F(\alpha) \). Show that \( \alpha^* \) is the VaR with confidence \( \beta \).

5. (10 marks) (Discrete Hedging and Optimality Conditions)

Consider an \( N \)-period binomial model for an underlying price movement in the interval \([0, T]\) and an European option with the payoff function, \( \text{payoff}(S_T) \), at \( T \). Let the interest rate be a constant \( r > 0 \). At time \( t = 0 \), construct the hedging portfolio \( \Pi_0 = \{\delta_0S_0, \eta_0\beta_0\} \) of the underlying \( S \) and bond \( \beta \) (assuming \( \beta_0 = 1 \)) to minimize the expected quadratic difference between \( \text{payoff}(S_T) \) and the replicating portfolio value \( \Pi_T \) at the expiry \( T \). In addition, at \( t = 0 \), the portfolio satisfies \( \Pi_0 = 0 \). Show that the optimal portfolio \( x^* = (\delta^*, \eta^*) \) solves the constrained optimization below

\[
\min_{x=(\delta, \eta)} \|Ax - b\|^2_2
\]

subject to

\[
c^Tx = d
\]

where \( \| \cdot \|^2_2 \) denotes the Euclidean norm, \( A \) is a matrix, and \( b \) is a vector. Provide the explicit forms for \( A, b, c, d \). Show that \( A \) is of full rank. Describe the KKT conditions for the constrained problem (7), derive expressions for the primal solution \( x^* \), the dual solution \( \nu^* \).
6. (10 marks) (Graduate Student Question)

Markowitz’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 \( r_1, \ldots, r_n \). Consider the following CVaR risk constrained allocation problem:

\[
\max_{x_1, \ldots, x_n} \quad \mathbb{E}\left( \sum_{i=1}^{n} r_i x_i \right) \\
\text{subject to} \quad \sum_{i=1}^{n} x_i = 1, \quad x \geq 0 \\
\text{CVaR}_\beta \left( -\sum_{i=1}^{n} r_i x_i \right) \leq \rho
\]

where \( x_i \) is the portfolio weight of the asset \( i \), and \( r_i \) 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 :

\[
\max_{(x,y,\alpha)} \quad \sum_{i=1}^{n} \mathbb{E}(r_i) x_i \\
\text{subject to} \quad \sum_{i=1}^{n} x_i = 1, \quad x_i \geq 0 \\
\quad \quad \alpha + \frac{1}{M(1-\beta)} \sum_{j=1}^{M} y_j \leq \rho \\
\quad \quad y_j \geq -\sum_{i=1}^{n} r_j^i x_i - \alpha, \quad j = 1, \ldots, M \\
\quad \quad y_j \geq 0, \quad j = 1, \ldots, M
\]

where \( r_j^i \) denote the jth sample of the return \( r_i \), \( \mathbb{E}(r_i) \) can be computed as the average from \( M \) samples, and \( \rho \) is an upper bound on the CVaR risk.

(a) Assume that a trading desk currently has a portfolio consisting of 2 stocks A, B, 6 options on stock A, 6 options on stock B as specified below:

- Stock A: OTM put with strikes 95\%\( S_0 \), 85\%\( S_0 \), 75\%\( S_0 \) and expiries of 5 months and 10 months respectively.
- Stock B: OTM call with strikes 105\%\( S_0 \), 110\%\( S_0 \), 115\%\( S_0 \) and expiries of 5 months and 10 months respectively.

Each stock price follows a constant volatility Black-Scholes model with \( \sigma_A = 15\% \) and \( \sigma_B = 25\% \), and the the correlation between returns of A and B is 70\%. In addition, assume that the expected returns are 6\% and 10\% for A and B respectively. The prices today for stock A and B are $100 and $115 respectively and the risk free rate is 2.5\%.

Write 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) Use matlab linprog and data generated above to plot an efficient frontier of investing in above stocks and options, with horizontal axis representing CVaR value with 90\% confidence and the vertical axis representing expected return. The efficient frontier should be generated by computing optimal portfolio with the upper bound \( \rho = \text{linspace}(-0.9, 0.9, 50) \). Describe how the optimal holdings change as \( \rho \) increases.

(c) What happens when a much smaller value is specified for the upper bound \( \rho \)? Explain why your observation is or is not reasonable.