Assignment guidelines:

- You should not hand in a first draft. Rewrite your solutions carefully and neatly. Use complete sentences. Make sure your arguments are clear.
- Unless otherwise specified MACF students should use an object-oriented programming language (*C++* or *Java*) for implementations/programming problems. Graduate and actuarial students may use *R* or *Matlab* if they wish but they are be required to implement everything from scratch without using any built-in mathematical functions.
- Your solutions should include the source code for all necessary functions, classes, etc. Your code should also be commented appropriately to indicate what your are doing in each step. Functions should be sufficiently general so they can be reused in other programming applications. You may be asked to demonstrate your program in class so that I can make sure it works. Please attach print-outs of you code with your written assignments.

Problems:

1. **Programming a Black-Scholes Implied Volatility Solver:** The purpose of this project is to build a flexible and and adaptable implied volatility solver in an object oriented frame-work using *C++*. Graduate and actuarial students may use *R* or *Matlab* if they wish but they shall be required to implement everything from scratch without using any built-in functions.

   The basic elements required shall be

   - A function that computes the values of the normal density $\phi(x)$ and cumulative distribution function $\Phi(x)$ (cdf). For the cdf you may use the approximation

   $$\Phi(x) \approx 1 - \phi(x)(b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5)$$

   where $t = \frac{1}{1+b_0 x}$, $b_0 = 0.2316419$, $b_1 = 0.319381530$, $b_2 = -0.356563782$, $b_3 = 1.781477937$, $b_4 = -1.821255978$, and $b_5 = 1.330274429$. All students should explore the accuracy of the normal cdf approximation given for a wide range of x values. Compare the accuracy of this approximation to the to another implementation (e.g. the pnorm function in R). There are intervals where where some adjustments are necessary but these are easy to make if you use simple properties of the normal cdf.
   - A function that computes Black-Scholes prices of European Call and Put Options. Having your function also return the option Delta would be useful.
   - A nonlinear equation solver (either bisection, Newton-Rhapson, or both) for $f(x) = y$.
   - An implied volatility calculator which accepts as inputs the observed option price, the market price of the stock, the strike price, the time to maturity, the interest rate, and an initial guess of the volatility (for Newton-Rhapson) or another method of initializing the algorithm (for bisection you need two values (a,b) that satisfy the conditions $f(a) < y$ and $f(b) > y$)).

   Write a program that uses the data from July 29, 2002 to calculate the implied volatilities for call and put option ask prices which expire on September 26, 2002. Assume $S_0 = 4.75$, and $r = 0.0492$ per annum (use a time scale in units of years with 365 days per year).

K	Call price-ask	Put price-ask
4.00	-	0.02
4.25	-	0.04
4.50	0.33	0.09
4.75	0.16	0.20
5.00	0.06	0.38
5.25	0.02	0.59
5.50	0.01	-
5.75	0.01	-

You will need to calculate the number of years to expiry.

What do you observe with this data (is there a notable 'volatility smile)? Compare your results to those you obtained in Assignment 1 for the same data.

2. **Fitting an Implied Volatility Surface** Data on out-of the money option with Apple (AAPL) stock as the underlying for February 17, 2012 is given in a spreadsheet posted on Moodle. The closing price of Apple stock on February 17, 2012 was \$502.12. The data includes the Last Price, Change in Price, Bid Price, Ask Price, Volume, Open Interest, and Strike Price. There is data for options expiring on March 17, 2012, April 21, 2012, ..., January 18, 2014. Data is also provided on the Greeks for each option on separate sheets (this is for information only).

   (a) Transfer the relevant data to a format that can be read by your programming environment for both Calls and Puts (Strike, Time to Maturity, Last Price, and Volume) for March, April, and May expiries. You may use Excel to do this if it is convenient.

   (b) Create a program (in C++ or R/Matlab for graduate students) that takes this data and filters out options for which the daily volume was less than 100 and for which the Last price was less than \$0.10.

   (c) Use the implied volatility solver you created (you might want to adjust it to use Newton-Rhapson rather than bisection) to calculate the implied volatility of each of these options. Assume $r = 0.0175$ and there are 365 days in a year. You should end up with data that looks something like Figure 1 for each expiry.

   (d) Create a program that takes the data and applies the one-dimensional kernel smoother

$$\hat{m}(x) = \frac{\sum_{i=1}^{n} K_h(x - x_i) y_i}{\sum_{i=1}^{n} K_h(x - x_i)}$$

   where

$$K_h(x) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x}{h}\right)^2},$$

   $h > 0$ is the smoothing parameter, (x_i, y_i) are the observed moneyness (S/K) and observed implied volatility, and x is an unobserved moneyness level. You do not need to do cross-validation to find the optimal smoothing parameter $h > 0$ but you should experiment to find a value that does not lead to over-smoothing (h too large) or over-fitting (h too small). Examples are illustrated in Figure 2. Use your function to create volatility smiles for moneyness in the interval $[0.5, 1.5]$ using a fine partition of the interval. Export this data to a program (Excel, R, Matlab) to create plots like the ones I have given. (You don't need to create plots in C++) **You should only use implied volatilities corresponding to out-of-the money options.**

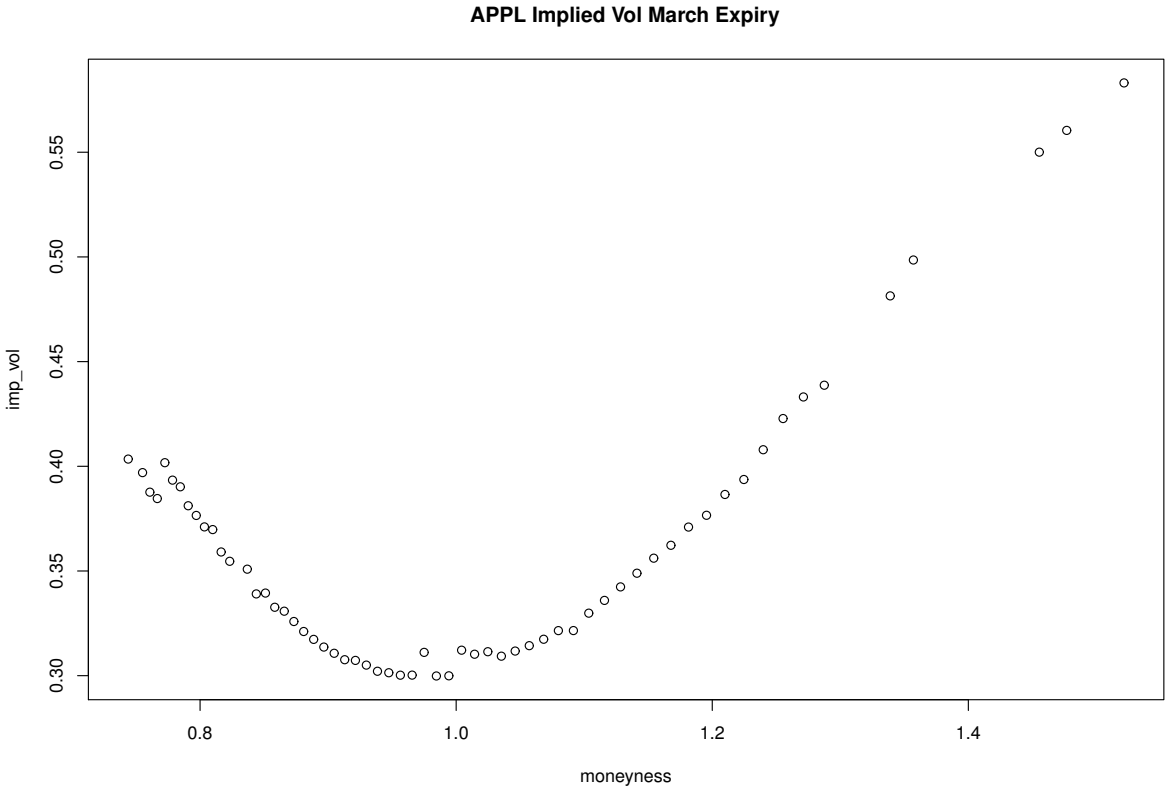

Figure 1: Implied Volatilities for AAPL March Expiry options

(e) Now create a program similar to the above to implement the two-dimensional Nadaraya-Watson estimator for the implied volatility surface

$$\hat{\sigma}(m,T) = \frac{\sum_{i=1}^{n} \sigma_{imp}(m_i,T_i) g(m-m_i, T-T_i)}{\sum_{i=1}^{n} g(m-m_i, T-T_i)}$$

where

$$g(x,y) = \frac{1}{2\pi} e^{-x^2/2h_1} e^{-y^2/2h_2},$$

$h_1 > 0$ is the smoothing parameter for moneyness, $h_2 > 0$ is the smoothing parameter for time to maturity, (m_i, T_i) are the moneyness ($m = S/K$) and time to maturity associated with the observed implied volatility $\sigma_{imp}(m_i, T_i)$, and use the data you generated in part (c) for the March, April, and May expiry options. Experiment with the h_i parameters to get a volatility surface on the moneyness time to maturity grid $(m,T) \in [m_{min}, m_{max}] \times [T_{min}, T_{max}] = [0.5, 1.5] \times [7/365, 105/365]$. Use a fine partition of the grid to generate the surface points and export the data to a program where you can graph the surface easily. **You should only use implied volatilities corresponding to out-of-the money options.**

(f) Create a new surface on the grid where $[m_{min}, m_{max}] \times [T_{min}, T_{max}]$ where the maximum and minimum values are correspond to the range of the data you are using. Comment on the effect of extending the surface past the last data points. Do you think this is wise?

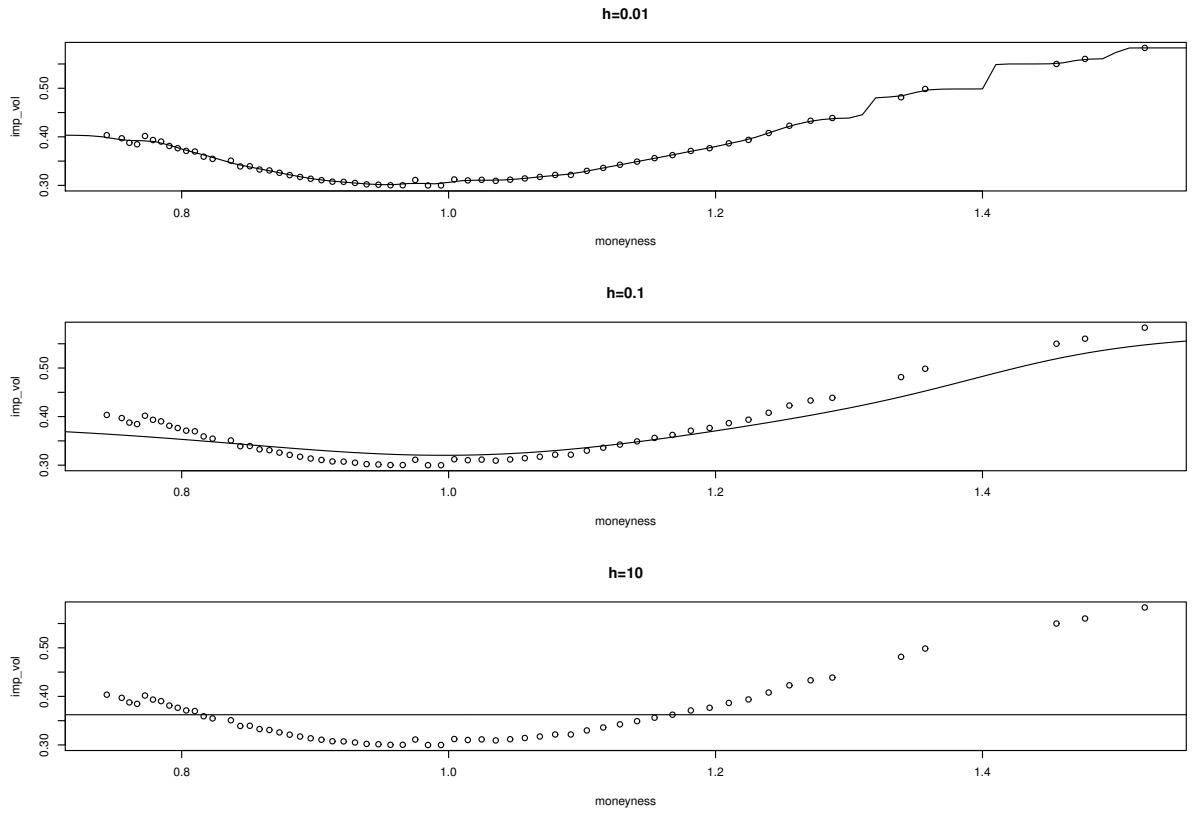

Figure 2: Fitted Smiles for AAPL March Expiry options

3. **No arbitrage restrictions on the volatility surface.** There are several properties of an implied volatility surface that must be satisfied to preclude arbitrage. Three important properties are

   - $\frac{\partial c}{\partial K} \leq 0$, $\frac{\partial p}{\partial K} \geq 0$ (No call or put spread arbitrage)
   - $\frac{\partial^2 c}{\partial K^2} \geq 0$, $\frac{\partial^2 p}{\partial K^2} \geq 0$ (No butterfly spread arbitrage)
   - $\frac{\partial c}{\partial T} \geq 0$, $\frac{\partial p}{\partial T} \geq 0$ (No calendar spread arbitrage)

   where c and p denote, respectively, the call and put values induced by the implied volatility surface $\hat{\sigma}(m, T)$.

   (a) Outline a method for testing these inequalities on the grid points of your implied volatility surface. Note that the prices of calls and puts implied by your implied volatility model can be given by the Black-Scholes formula evaluated at the corresponding point on the surface (eg $c = C_{BS}(S, T, K, r, \hat{\sigma}(m, T))$ so when considering the above derivative inequalities you should think about the chain rule for partial derivatives (combine this with formulae for the Black-Scholes model Greeks which you can find below).

   (b) Implement your method and apply it the surface from Question 2.

   (c) If you identify any arbitrage opportunities in your implied volatility surface try to refine your surface (by changing the smoothing parameter, dropping some outliers, etc) to eliminate them.

4. **Pricing a vanilla option based on the volatility surface.** Consider a fixed maturity of $T = 60/365$.

   (a) Use your implied volatility surface generated from the Apple data to price a European call option with strike price $K = 572.50$

   (b) To delta hedge a European call option we should use the hedge ratio

   $$\Delta = \frac{\partial C_{BS}}{\partial S}(S, T, K, r, \hat{\sigma}(m, T))$$

   Give the hedge ratio for the option from part (a).

Selected Black-Scholes Model Greeks for European Call Option ($t = 0$)

$$\Delta_{BS} = \frac{\partial C_{BS}}{\partial S} = \Phi(d_1)$$
$$\Gamma_{BS} = \frac{\partial^2 C_{BS}}{\partial S^2} = \frac{\phi(d_1)}{S\sigma\sqrt{T}}$$
$$vega_{BS} = \frac{\partial C_{BS}}{\partial \sigma} = S\sqrt{T}\phi(d_1)$$
$$\rho_{BS} = TKe^{-rT}\Phi(d_2)$$