1. Given a smooth function $y : \mathbf{R} \to \mathbf{R}$ let y_m denote the value $y(\Delta \cdot m)$ where $\Delta > 0$ and m is an integer. The quantity

$$y_{m+1} - y_m$$

is called the *forward difference* and the associated *forward difference operator*, δ^+, is defined by

$$\delta^+ y_m = y_{m+1} - y_m.$$

(a) By expanding $y(\Delta(m+1))$ as a Taylor series about the point (Δm) verify the Taylor series expansion

$$\delta^+ y_m = \Delta y'(\Delta m) + \frac{1}{2}(\Delta)^2 y''(\Delta m) + \cdots$$

What are the assumptions?

(b) Use Taylor's theorem to show that

$$\delta^+ y_m = \Delta y'(z_m)$$

for some point $z_m \in (\Delta m, \Delta(m+1))$.

2. Similar to Question 1 define the *backward difference*

$$y_m - y_{m-1}$$

and the associated *backward difference operator*, δ^-, by

$$\delta^- y_m = y_m - y_{m-1}.$$

(a) Verify the Taylor series expansion

$$\delta^- y_m = \Delta y'(\Delta m) - \frac{1}{2}\Delta^2 y''(\Delta m) + \cdots$$

(b) Show that

$$\delta^- y_m = \Delta y'(z_m)$$

for some point $z_m \in (\Delta(m-1), \Delta m)$.

3. Define the *second order central difference*

$$y_{m+1} - 2y_m + y_{m-1}$$

and the *second order central difference operator*, δ^2, by

$$\delta^2 y_m = y_{m+1} - 2y_m + y_{m-1}.$$

Define the *half central difference*

$$y_{m+\frac{1}{2}} - y_{m-\frac{1}{2}},$$

the associated *half central difference operator*, δ, by

$$\delta y_m = y_{m+\frac{1}{2}} - y_{m-\frac{1}{2}},$$

and $y_{m\pm\frac{1}{2}} = y((m \pm \Delta)$.

(a) Give a Taylor series for $\delta^2 y_m$.

(b) Similar to part (b) of Questions 1 and 2 show that the error of the second order central difference is proportional to Δ^2.

(c) Show that $\delta^+\delta^- = \delta^2$.

4. Recall we showed in class that the FTCS algorithm for the canonical heat equation may be written in the form

$$\vec{U}^{n+1} = F\vec{U}^n + \vec{p}^n$$

for $0 \le n \le N-1$, a suitable $(J-1)\times(J-1)$ matrix F, and a suitable $(J-1)\times 1$ vector $\vec{p^n}$ (in terms of the boundary conditions in the x coordinate). The initial conditions $\vec{U}^0$ are in terms of the initial condition in the time coordinate $u(x,0)$.

   (a) Verify that the BTCS algorithm may be written in the form

$$B\vec{U}^{n+1} = \vec{U}^n + \vec{q}^n$$

   for $0 \le n \le N-1$, a suitable $(J-1)\times(J-1)$ matrix B, and a suitable $(J-1)\times 1$ vector $\vec{q^n}$ (in terms of the boundary conditions in the x coordinate). Specify the initial condition $\vec{U}^0$ in terms of the initial condition in the time coordinate $u(x,0)$.

Note: With implicit methods a tri-diagonal system of equations must be solved in order to get $\{U_j^{n+1}\}$ from $\{U_j^n\}$. Using Gaussian elimination or matrix inversion is typically very inefficient. The Thomas algorithm (see `http://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm`) is a version of Gaussian elimination: suppose that

$$c_i v_{i-1} + a_i v_i + b_i v_{i+1} = f_i, \quad 1 \le i \le J,$$

and that v_0 and v_{J+1} are known. Then we perform the following:

$f_1 \leftarrow f_1 - a_1 v_0$
$f_J \leftarrow f_J - a_J v_{J+1}$
for $j = 1$ to $J-1$
  $a_{j+1} \leftarrow a_{j+1} - b_j c_{j+1}/a_j$
  $f_{j+1} \leftarrow f_{j+1} - f_j c_{j+1}/a_j$
end
$v_J \leftarrow f_J/a_J$
for $j = J-1$ to 1
  $v_j \leftarrow (f_j - b_j v_{j+1})/a_j$
end

Thomas algorithm

5. Consider a one-year maturity, at-the-money, European put option with $S_0 = 100$, $\sigma = 0.20$, and $r = 0.03$. The goal of this question is to implement the FTCS algorithm to find the price of the option at time zero. First transforming the Black-Scholes PDE and boundary conditions to the canonical heat equation $u_t = u_{xx}$ as indicated in class, write a program to implement the FTCS algorithm in this case, finally ensure that your program transforms the solution $u(x, t)$ back into a solution of the Black-Scholes PDE. We use a maximum value for the stock price at least three times the strike (i.e., $S_{\text{max}} \geq 3K$) to create the grid.

   (a) Investigate the performance of the algorithm relative to the ("true") Black-Scholes price for $J = 10$, 30, 50, 70, 90, and 110 by reporting the maximum errors in each case (use an appropriate N to ensure stability).

   (b) Produce surface plots of the approximate value of the European put as a function of stock-price and time to maturity.

   (c) **[MAST 729 only]** compare the delta of the option calculated using the finite difference

   $$\frac{V(S+h,t) - V(S,t)}{h}$$

   at each point on your mesh and plot the result as a surface. Compare with the theoretical delta of the option. Do this for $J = 10, 30, 50$ and comment on the results.

6. The delta-normal approximation to VaR for the change in the value of a financial instrument $V(S)$ depending on risk factor S is approximates the change in value ΔV related to the change ΔS in S by the approximation

   $$\Delta V \approx V'(S_0)\Delta S.$$

   Under the assumption that ΔS is $N(0, \sigma^2)$ we have that $\Delta V \approx N(0, \delta_0^2\sigma^2)$.

   Consider a European call option with parameters S_0, K, r, σ, T (in years). Assuming a geometric Brownian motion model for the stock price process S_t use the delta-normal valuation to compute the 95% VaR over a horizon of 5 days for

   (a) a short position and

   (b) a long position.

   (c) Write a program to simulate the value of your portfolio (in both the long and the short case) N times over the 5 day period. Let $S_0 = 100$, $K = 100$, $r = 0.02$, $\sigma = 0.5$, $T = 1$. Run the program with $N = 10^2, 10^3, 10^4, 10^5, 10^6$ and use the results to calculate the number of times the 95% VaR calculated using the results of (a) and (b) is exceeded. Report the results in the long and short case for each N.