This is a closed-book test: no books, no notes, no calculators, no phones, no tablets, no computers (of any kind) allowed.

Do NOT turn this page over until you are TOLD to start.

Duration of the exam: 3 hours.

The exam consists of 13 pages, including this one. Make sure you have all 13 pages.

The exam consists of 4 questions. Answer all 4 questions in the exam booklets provided. The mark for each question is listed at the start of the question.

Please fill-in ALL the information requested on the front cover of EACH exam booklet that you use.

The exam was written with the intention that you would have ample time to complete it. You will be rewarded for concise well-thought-out answers, rather than long rambling ones. We seek quality rather than quantity.

Moreover, an answer that contains relevant and correct information as well as irrelevant or incorrect information will be awarded fewer marks than one that contains the same relevant and correct information only.

Write legibly. Unreadable answers are worthless.
1. [10 marks: 5 marks for each part]

When you multiply \( M + 1 \) floating-point numbers, \( x_0, x_1, \ldots, x_M \), together in IEEE floating-point arithmetic, we know that the computed value, \( \text{fl}(x_0 \cdot x_1 \cdot x_2 \cdots x_M) \), satisfies

\[
\text{fl}(x_0 \cdot x_1 \cdot x_2 \cdots x_M) = x_0 \cdot x_1 \cdot x_2 \cdots x_M (1 + \delta_1)(1 + \delta_2) \cdots (1 + \delta_M) \tag{1}
\]

where \( |\delta_i| \leq \frac{1}{2} \varepsilon_{\text{mach}} \), for \( i = 1, 2, \ldots, M \), and \( \varepsilon_{\text{mach}} \), also called \textit{machine epsilon}, is the distance from 1 to the next larger machine number in the floating-point system used for the calculation of the product in (1). Assuming none of the \( x_i = 0 \), the relative error associated with the calculation (1) is

\[
\frac{\text{fl}(x_0 \cdot x_1 \cdot x_2 \cdots x_M) - x_0 \cdot x_1 \cdot x_2 \cdots x_M}{x_0 \cdot x_1 \cdot x_2 \cdots x_M} = (1 + \delta_1)(1 + \delta_2) \cdots (1 + \delta_M) - 1 \approx \delta_1 + \delta_2 + \cdots + \delta_M \tag{2}
\]

where the \( \approx \) in (2) is quite good if \( M \varepsilon_{\text{mach}} \ll 1 \).

If we were to use the worse case bound for (2), we would get

\[
\left\| \frac{\text{fl}(x_0 \cdot x_1 \cdot x_2 \cdots x_M) - x_0 \cdot x_1 \cdot x_2 \cdots x_M}{x_0 \cdot x_1 \cdot x_2 \cdots x_M} \right\| = |\delta_1 + \delta_2 + \cdots + \delta_M| \\
\leq |\delta_1| + |\delta_2| + \cdots + |\delta_M| \\
\leq M \varepsilon_{\text{mach}} \tag{3}
\]

However, in practice, the error is usually much smaller than \( M \varepsilon_{\text{mach}} \). To illustrate this, I multiplied \( M = 1,000 \) positive single-precision IEEE floating-point numbers together and calculated the relative error

\[
\frac{\text{fl}(x_0 \cdot x_1 \cdot x_2 \cdots x_M) - x_0 \cdot x_1 \cdot x_2 \cdots x_M}{x_0 \cdot x_1 \cdot x_2 \cdots x_M} \tag{4}
\]

Note that \( \varepsilon_{\text{mach}} = 2^{-23} \) for IEEE single-precision floating-point numbers. Therefore, \( M \varepsilon_{\text{mach}} \approx 1.2 \cdot 10^{-4} \ll 1 \) in this case.

I repeated this calculation \( N = 10,000 \) times, each time using a different set of randomly chosen numbers, \( x_0, x_1, \ldots, x_M \). I saved the \( N = 10,000 \) relative errors, one for each calculation of (4). I then divided each of these relative errors by

\[
\sqrt{M \varepsilon_{\text{mach}}} \tag{5}
\]

and plotted a histogram of these “normalized” relative errors in Figure 1. That is, the values plotted in the histogram shown in Figure 1 are

\[
\frac{\left( \text{fl}(x_0 \cdot x_1 \cdot x_2 \cdots x_M) - (x_0 \cdot x_1 \cdot x_2 \cdots x_M) \right)}{\sqrt{M \varepsilon_{\text{mach}}}}
\]
Notice that the histogram looks very much like a normal distribution with a mean that appears to be about 0 and a standard deviation that appears to be about

\[ c \sqrt{M} \frac{\epsilon_{\text{mach}}}{2} \]  

for some constant \( c \).

(a) Explain why the \( N = 10,000 \) normalized relative errors should have a distribution that is approximately normal.

(b) What is your best estimate for the value of \( c \) in formula (6)? Justify your answer.

In answering this question, you can assume that the rounding errors \( \delta_i, i = 1, 2, \ldots, M \), in (1) are independent and each \( \delta_i \) is uniformly distributed in \( [-\frac{\epsilon_{\text{mach}}}{2}, \frac{\epsilon_{\text{mach}}}{2}] \). That is, \( \delta_i \sim \text{Unif}[-\frac{\epsilon_{\text{mach}}}{2}, \frac{\epsilon_{\text{mach}}}{2}] \), for each \( i = 1, 2, \ldots, M \).

(The assumption above is a little too strong, but use it for this question. It is reasonable to assume that the \( \delta_i \) are independent and identically distributed (i.e., i.i.d.), but they don’t really seem to be uniformly distributed in \( [-\frac{\epsilon_{\text{mach}}}{2}, \frac{\epsilon_{\text{mach}}}{2}] \).)

![Figure 1: Histogram of the “normalized” relative errors](image-url)
2. [10 marks: 5 marks for each part]

Suppose you need to generate a random variable \( X \) having the probability density function (pdf)

\[
f(x) = xe^{-x}, \quad x \geq 0.
\]

You could try using the inverse CDF method, but that is not so easy in this case, since the CDF associated with the pdf \( f(x) \) is

\[
F(x) = \int_0^x f(t) \, dt = 1 - e^{-x} - xe^{-x}
\]

and it is not too easy to compute \( X = F^{-1}(U) \) (or equivalently to solve \( F(X) = U \)) for \( X \) given a \( U \sim \text{Unif}[0,1] \).

So, let’s consider the acceptance-rejection method. For this method, we first need to find a function \( t(x) \) such that

\[
f(x) \leq t(x) \quad \text{for all} \quad x \geq 0
\]

and we want

\[
c = \int_0^\infty t(x) \, dx
\]

to be reasonably small, since \( 1/c \) is the probability of acceptance in the acceptance-rejection method. Note that

\[
c = \int_0^\infty t(x) \, dx \geq \int_0^\infty f(x) \, dx = 1
\]

So, \( c \geq 1 \).

In addition, we need to be able to generate easily random numbers \( Y \) having the probability density function

\[
g(x) = \frac{1}{c} t(x)
\]

As a first guess for a suitable function \( t(x) \) consider the function

\[
t_1(x) = \alpha e^{-\beta x}, \quad x \geq 0,
\]

where \( \alpha \) and \( \beta \) are free parameters to be chosen.

(a) Describe how you should choose \( \alpha \) and \( \beta \) so that

\[
f(x) \leq t_1(x) \quad \text{for all} \quad x \geq 0
\]

and

\[
c_1 = \int_0^\infty t_1(x) \, dx
\]

is as small as possible.

Give mathematical expressions (in as simple a form as possible) for \( \alpha, \beta \) and \( c_1 \). (You probably won’t be able to give actual numbers for all of \( \alpha, \beta \) and \( c_1 \). To do so, you would need a calculator and I know you don’t have one for this exam.)
The function $t_1(x)$ that I found is shown in Figure 2 below. For my $t_1(x)$, $c_1 \approx 1.4715$ and $1/c_1 \approx 0.6796$. Since the probability of acceptance in the acceptance-rejection method is $1/c_1$, this implies that an acceptance-rejection method built on this $t_1(x)$ and the corresponding pdf

$$g_1(x) = \frac{1}{c_1}t_1(x)$$

would accept about 68% of the $Y$’s that it generated from $g_1(x)$. This is not bad, but we can do much better.

Figure 2: The dashed curve is $f(x) = xe^{-x}$ and the solid curve is $t_1(x) = \alpha e^{-\beta x}$.

Note that my $t_1(x)$ significantly overestimates $f(x)$ for $x \in [0, 2]$. Yours probably does too. So, let’s construct a new $t_2(x)$ by modifying $t_1(x)$ to better fit $f(x)$ for $x \in [0, 2]$. However, when making this modification, we must ensure that it is easy to generate random variables from the pdf $g_2(x)$ associated with $t_2(x)$.

To begin, note that

$$f'(x) = e^{-x} - xe^{-x} = (1 - x)e^{-x}$$

So, $f(x)$ has a unique maximum at $x = 1$ and $f(1) = e^{-1}$. So, around $x = 1$, it is reasonable to approximate $f(x)$ by $t_2(x) = e^{-1}$. So, for $x \geq 1$, let’s approximate $f(x)$ by

$$t_2(x) = \min(e^{-1}, t_1(x))$$
Note that 
\[ f(x) \leq t_2(x) \]
since 
\[ f(x) \leq e^{-1} \quad \text{and} \quad f(x) \leq t_1(x) \]
Also note that 
\[ e^{-1} = t_1(x) = \alpha e^{-\beta x} \text{ at } x_2 = \frac{1 + \log(\alpha)}{\beta} \]

So, for \( x \geq 1 \),
\[ t_2(x) = \min(e^{-1}, t_1(x)) = \begin{cases} e^{-1} & \text{for } x \in [1, x_2] \\ t_1(x) & \text{for } x > x_2 \end{cases} \] (7)

For \( x \) a little less than 1, it seems reasonable to approximate \( f(x) \) by \( e^{-1} \) also, but \( e^{-1} \) is not a very good approximation to \( f(x) \) for \( x \) near 0. However, note that 
\[ f''(x) = -2e^{-x} + xe^{-x} = (x - 2)e^{-x} \]
So, \( f''(x) < 0 \) for \( x \in [0, 1] \). That is, \( f(x) \) is concave for \( x \in [0, 1] \). Hence, any line that is tangent to \( f(x) \) lies above \( f(x) \). That is, for any \( \bar{x} \in [0, 1] \), if 
\[ l(x) = f(\bar{x}) + f'(\bar{x})(x - \bar{x}) \]
then 
\[ f(x) \leq l(x) \quad \text{for all } x \in [0, 1] \]

Hence, for \( x \) near 0, we can approximate \( f(x) \) by 
\[ l_2(x) = f(0) + f'(0)(x - 0) = x \]
and we are guaranteed that 
\[ f(x) \leq l_2(x) = x \quad \text{for all } x \in [0, 1] \]

Hence, for \( x \in [0, 1] \), we can approximate \( f(x) \) by 
\[ t_2(x) = \min(l_2(x), e^{-1}) = \min(x, e^{-1}) \]
and we are guaranteed that 
\[ f(x) \leq t_2(x) \quad \text{for all } x \in [0, 1] \]

Now note that \( x = l_2(x) = e^{-1} \) at \( x_1 = e^{-1} \). So, for \( x \in [0, 1] \),
\[ t_2(x) = \min(l_2(x), e^{-1}) = \begin{cases} x & \text{for } x \in [0, x_1] \\ e^{-1} & \text{for } x \in [x_1, 1] \end{cases} \] (8)
Putting (7) and (8) together, we get

\[ t_2(x) = \begin{cases} 
  x & \text{for } x \in [0, x_1] \\
  e^{-1} & \text{for } x \in [x_1, x_2] \\
  t_1(x) & \text{for } x > x_2
\end{cases} \]  

(9)

and we are guaranteed that

\[ f(x) \leq t_2(x) \quad \text{for all } x \geq 0 \]

My \( t_2(x) \) is shown in Figure 3. For my \( t_2(x) \), \( c_2 \approx 1.1781 \) and \( 1/c_2 \approx 0.8488 \). Since the probability of acceptance in the acceptance-rejection method is \( 1/c_2 \), this implies that an acceptance-rejection method built on this \( t_2(x) \) and the corresponding pdf

\[ g_2(x) = \frac{1}{c_2} t_2(x) \]

would accept about 85% of the \( Y \)'s that it generated from \( g_2(x) \). This is quite good.

Figure 3: The dashed curve is \( f(x) = xe^{-x} \) and the solid curve is \( t_2(x) \) defined in (9).

(b) Describe how to generate a random number \( Y \) having the pdf \( g_2(x) \).

Give mathematical expressions (in as simple a form as possible) for the values that you need in your method for generating \( Y \). (You probably won’t be able to give actual numbers for all of the values you need in these expressions. To do so, you would need a calculator and I know you don’t have one for this exam.)
You can actually get a more efficient version of the acceptance-rejection method described above by choosing $\alpha$ and $\beta$ to minimize
\[
\int_{2}^{\infty} \alpha e^{-\beta x} \, dx
\]
subject to
\[
f(x) \leq \alpha e^{-\beta x} \quad \text{for all } x \geq 2
\]
This also changes the value of $x_2$. With this modification, $c_2 \approx 1.0914$ and $1/c_2 \approx 0.9162$. An acceptance-rejection method built on this $t_2(x)$ and the corresponding pdf
\[
g_2(x) = \frac{1}{c_2} t_2(x)
\]
would accept about 92% of the $Y$’s that it generated from $g_2(x)$. This is very good.

(You don’t have to answer anything about what I’ve written above on this page; I included it only to give you an idea of how effective this approach can be.)
3. [15 marks: 5 marks for each part]

Assume that a stock price, $S_t$, follows the simple SDE

$$dS_t = rS_t dt + \sigma S_t dW_t$$

where the risk-free interest rate, $r$, and the volatility, $\sigma$, are constants. Also, $W_t$ is a Wiener process (also called a standard Brownian motion).

Suppose we wish to price a European put option based on $S_t$ that has the payoff

$$h(X) = \begin{cases} (K_1 - S_T)^+ & \text{if } S_{T/2} < H \\ (K_2 - S_T)^+ & \text{otherwise} \end{cases}$$

where $S_{T/2}$ is $S_t$ evaluated at $t = T/2$, $S_T$ is $S_t$ evaluated at $t = T$, $X = (S_{T/2}, S_T)$ and

$$(K - S_T)^+ = \max(K - S_T, 0)$$

(a) Give a simple Monte Carlo method to evaluate this European put option with payoff function $h(X)$.

(b) The method in part (a) needs two independent normal random numbers (one to compute $S_{T/2}$ and another to compute $S_T$) for each Monte Carlo replication that it performs. (Each replication evaluates $h(X)$ once.)

Give a Monte Carlo method with conditioning that requires only one normal random number for each Monte Carlo replication that it performs.

(c) You wrote two Monte Carlo methods to compute the price of this European put option with payoff function $h(X)$, one for part (a) and one for part (b). Which of these two Monte Carlo methods is more efficient?

Justify your answer.

In writing your Monte Carlo methods for parts (a) and (b), you can use any of the standard MatLab functions, including rand, randn, blsprice, etc.
4. [15 marks: 5 marks for each part]

The Black-Scholes PDE for a European put option is

\[
\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0
\]

(10)

for \( t \in [0, T] \) and \( S \in [0, \infty) \). The terminal and boundary conditions associated with the Black-Scholes PDE (10) for a European put option are

\[
V(T, S) = \max(K - S, 0)
\]

\[
V(t, 0) = Ke^{-r(T-t)}
\]

\[
\lim_{S \to \infty} V(t, S) = 0
\]

(11)

In (10) and (11), \( t \) is time, \( S \) is the stock price, \( \sigma \) is the volatility, \( r \) is the risk-free interest rate, \( K \) is the strike price, \( T \) is the expiry (maturity) time of the option and \( V(t, S) \) is the price of the European put option at any time \( t \in [0, T] \) and any stock price \( S \in [0, \infty) \).

This problem is simpler if we make the change of variables \( \tau = T - t \) and \( x = \log(S/K) \). Then we let \( v(\tau, x) = V(T - \tau, Ke^{x}) \) and derive the following PDE for \( v(\tau, x) \)

\[
\frac{\partial v}{\partial \tau} = \frac{1}{2} \sigma^2 \frac{\partial^2 v}{\partial x^2} + \left( r - \frac{1}{2} \sigma^2 \right) \frac{\partial v}{\partial x} - rv
\]

(12)

where \( \tau \in [0, T] \) and \( x \in (-\infty, \infty) \). The PDE (12) has the initial and boundary conditions

\[
v(0, x) = K \max(1 - e^x, 0)
\]

\[
\lim_{x \to -\infty} v(\tau, x) = Ke^{-r\tau}
\]

\[
\lim_{x \to \infty} v(\tau, x) = 0
\]

(13)

Before we can solve the PDE (12) numerically, we need to truncate the \( x \)-domain. That is, we need to choose an \( x_{\min} \) and an \( x_{\max} \) and truncate the \( x \)-domain from \((-\infty, \infty)\) to \([x_{\min}, x_{\max}]\). We’ll solve (12) for \( t \in [0, T] \) and \( x \in [x_{\min}, x_{\max}] \) with the initial and boundary conditions

\[
v(0, x) = K \max(1 - e^x, 0)
\]

\[
v(\tau, x_{\min}) = Ke^{-r\tau}
\]

\[
v(\tau, x_{\max}) = 0
\]

(14)
(a) Suppose you want to price the European put option at \( t = 0 \) and \( S = K \). How would you choose \( x_{\text{min}} \) and \( x_{\text{max}} \) so that your solution for \( V(0, K) = v(T, 0) \) using the PDE (12) and initial and boundary conditions (14) is a good approximation to the solution \( v(T, 0) \) using the PDE (12) and initial and boundary conditions (13)? Justify your answer.

Hint: it may be helpful to recall that the Black-Scholes PDE is based on the simple SDE model

\[
dS_t = rS_t dt + \sigma S_t dW_t
\]

for the stock price, \( S_t \), where \( W_t \) is a Wiener process (also called a standard Brownian motion).

Having chosen \( x_{\text{min}} \) and \( x_{\text{max}} \) in part (a) above, we can now choose integers \( M \) and \( N \) and an associated grid

\[
\tau_i = i\Delta \tau, \text{ for } i = 0, 1, \ldots, M, \text{ and } \Delta \tau = T/M,
\]

\[
x_j = x_{\text{min}} + j\Delta x, \text{ for } j = 0, 1, \ldots, N + 1, \text{ and } \Delta x = \frac{x_{\text{max}} - x_{\text{min}}}{N + 1}
\]

Now consider the explicit method

\[
\frac{v_{i+1,j} - v_{i,j}}{\Delta \tau} = \frac{1}{2} \sigma^2 \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{(\Delta x)^2} + \left( r - \frac{1}{2} \sigma^2 \right) \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta x} - rv_{i,j}
\]

for \( i = 0, 1, \ldots, M - 1 \) and \( j = 1, 2, \ldots, N \) with the initial and boundary conditions

\[
v_{0,j} = K \max(1 - e^{x_j}, 0), \text{ for } j = 0, 1, \ldots, N + 1
\]

\[
v_{i,0} = Ke^{-r\tau_i}, \text{ for } i = 1, 2, \ldots, M
\]

\[
v_{i,N+1} = 0 \text{ for } i = 1, 2, \ldots, M
\]

Note that \( v_{i,j} \) is meant to be an approximation to \( v(\tau_i, x_j) \), where \( v(\tau, x) \) is the solution to the PDE (12) with the initial and boundary conditions (14).

(b) Show that the explicit method (16) with the initial and boundary conditions (17) is a consistent approximation to the PDE (12) with the initial and boundary conditions (14) of order 1 in \( \tau \) and order 2 in \( x \). That is, the truncation error is \( O(\Delta \tau) + O((\Delta x)^2) \).

Hence, if we can show that the explicit method (16) with the initial and boundary conditions (17) is stable, we’ll be able to conclude that the explicit method converges with order 1 in \( \tau \) and order 2 in \( x \). That is,

\[
v_{i,j} = v(\tau_i, x_j) + O(\Delta \tau) + O((\Delta x)^2)
\]

for \( i = 0, 1, \ldots, M \) and \( j = 0, 1, \ldots, N + 1 \). So, in the remainder of this question, we’ll try to show that the explicit method (16) with the initial and boundary conditions (17) is stable.
To that end, rewrite (16) as

\[ v_{i+1,j} = \alpha v_{i,j-1} + \beta v_{i,j} + \gamma v_{i,j+1} \]  

(18)

for \( i = 0, 1, \ldots, M - 1 \) and \( j = 1, 2, \ldots, N \), where

\[
\alpha = \frac{1}{2} \sigma^2 \frac{\Delta \tau}{(\Delta x)^2} - \left( r - \frac{1}{2} \sigma^2 \right) \frac{\Delta \tau}{2 \Delta x}
\]

\[
\beta = 1 - r \Delta \tau - \sigma^2 \frac{\Delta \tau}{(\Delta x)^2}
\]

\[
\gamma = \frac{1}{2} \sigma^2 \frac{\Delta \tau}{(\Delta x)^2} + \left( r - \frac{1}{2} \sigma^2 \right) \frac{\Delta \tau}{2 \Delta x}
\]

Note that \( \alpha, \beta \) and \( \gamma \) depend on \( \Delta \tau \) (equivalently \( M \)) and \( \Delta x \) (equivalently \( N \)) as well as the constants \( \sigma \) and \( r \), although I haven’t indicated this explicitly through the notation here.

Suppose we perturb (18) to

\[ \hat{v}_{i+1,j} = \alpha \hat{v}_{i,j-1} + \beta \hat{v}_{i,j} + \gamma \hat{v}_{i,j+1} + \Delta \tau \epsilon_{i,j} \]  

(19)

for \( i = 0, 1, \ldots, M - 1 \) and \( j = 1, 2, \ldots, N \). The explicit method (16) with the initial and boundary conditions (17) is stable if we can show that there exists a constant \( L \) such that, for any \( M, N \) and \( \epsilon > 0 \), if

\[
|\hat{v}_{0,j} - v_{0,j}| \leq \epsilon \text{ for } j = 0, 1, \ldots, N + 1
\]

\[
|\hat{v}_{i,0} - v_{i,0}| \leq \epsilon \text{ for } i = 1, 2, \ldots, M
\]

\[
|\hat{v}_{i,N+1} - v_{i,N+1}| \leq \epsilon \text{ for } i = 1, 2, \ldots, M
\]

and

\[
|\epsilon_{i,j}| \leq \epsilon \text{ for } i = 0, 1, \ldots, M - 1 \text{ and } j = 1, 2, \ldots, N
\]

then

\[
|\hat{v}_{i,j} - v_{i,j}| \leq L\epsilon
\]

(20)

for \( i = 0, 1, \ldots, M \) and \( j = 0, 1, 2, \ldots, N + 1 \).

(c) Show that, if you restrict your choices of \( M \) and \( N \) (equivalently \( \Delta \tau \) and \( \Delta x \)) to those for which \( \alpha \geq 0, \beta \geq 0 \) and \( \gamma \geq 0 \), then the explicit method (16) with the initial and boundary conditions (17) is stable.

Hint: Let \( E_i = \max_{0 \leq j \leq N+1} |\hat{v}_{i,j} - v_{i,j}| \) and show that

\[
E_{i+1} \leq (\alpha + \beta + \gamma) E_i + \Delta \tau \epsilon
\]

(21)
Then show that (21) implies that

\[ E_i \leq \epsilon + i\Delta \tau \epsilon \] (22)

for \( i = 0, 1, \ldots, M \). Hence,

\[ |\hat{v}_{i,j} - v_{i,j}| \leq E_i \leq (1 + T)\epsilon \]

for \( i = 0, 1, \ldots, M \) and \( j = 0, 1, \ldots, N + 1 \). This shows that the stability bound (20) holds with \( L = 1 + T \).

Note that the condition \( \beta \geq 0 \) that we required for the stability of the explicit method (16) with the initial and boundary conditions (17) is very similar to the requirement that

\[ \frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2} \]

that we found in class for the stability of the explicit method applied to the heat equation.

The conditions \( \alpha \geq 0 \) and \( \gamma \geq 0 \) are equivalent to

\[ \frac{1}{2} \sigma^2 \frac{\Delta \tau}{(\Delta x)^2} \geq \left| \left( r - \frac{1}{2} \sigma^2 \right) \frac{\Delta \tau}{2\Delta x} \right| \]

which in turn is equivalent to

\[ \sigma^2 \geq \Delta x \left| r - \frac{1}{2} \sigma^2 \right| \] (23)

Provided \( \sigma \neq 0 \), inequality (23) will be satisfied if \( \Delta x \) is small enough. Note that, if \( \sigma = 0 \), the nature of the differential equation (10) changes significantly from a parabolic equation to a hyperbolic equation. Note, also from a financial point of view, \( \sigma = 0 \) implies that the SDE (15), which governs the behavior of stock prices in the Black-Scholes model, changes to

\[ dS_t = rS_t dt \] (24)

That is, stock prices are completely deterministic, which is unrealistic. So, it is reasonable to assume in this context that \( \sigma \neq 0 \).

So, the conditions we found here for the stability of the explicit method (16) with the initial and boundary conditions (17) are very similar to the condition we found in class for the stability of the explicit method applied to the heat equation.

Have a Happy Holiday