

Duration of the exam: 3 hours.

Answer ALL questions.

This is a **closed-book exam**: no books, no notes, no calculators allowed.

1. [10 marks: 5 marks for each part]

In 250 BC, the Greek mathematician Archimedes estimated the number π as follows. He drew a circle with diameter 1, and hence circumference π . Inside the circle he inscribed a square. The perimeter of the square is smaller than the circumference of the circle, and so it is a lower bound for π . Archimedes then considered an inscribed octagon, 16-gon, 32-gon, etc., each time doubling the number of sides of the inscribed polygon, thereby producing ever better estimates of π . Using 96-sided inscribed and circumscribed polygons, he was able to show that $223/71 < \pi < 22/7$.

There is a recursive formula for these estimates. Let p_n be the perimeter of an inscribed polygon with 2^n sides. Then it is possible to show that $p_2 = 2\sqrt{2}$ and

$$p_{n+1} = 2^n \sqrt{2(1 - \sqrt{1 - (p_n/2^n)^2})} \quad \text{for } n \geq 2$$

Moreover, $p_n \rightarrow \pi$ as $n \rightarrow \infty$.

You don't have to prove the results above. Just take them as facts.

I used the formula above to compute p_n and $p_n - \pi$ for $n = 2, 3, \dots, 33$ in MatLab using IEEE double-precision arithmetic. The results that I obtained are listed below.

n	p_n	$p_n - \pi$
2	2.828427124746190	-3.1317e-01
3	3.061467458920719	-8.0125e-02
4	3.121445152258053	-2.0148e-02
5	3.136548490545941	-5.0442e-03
6	3.140331156954739	-1.2615e-03
7	3.141277250932757	-3.1540e-04
8	3.141513801144145	-7.8852e-05
9	3.141572940367883	-1.9713e-05
10	3.141587725279961	-4.9283e-06
11	3.141591421504635	-1.2321e-06
12	3.141592345611077	-3.0798e-07
13	3.141592576545004	-7.7045e-08
14	3.141592633463248	-2.0127e-08
15	3.141592654807589	1.2178e-09
16	3.141592645321215	-8.2686e-09
17	3.141592607375720	-4.6214e-08

n	p_n	$p_n - \pi$
18	3.141592910939673	2.5735e-07
19	3.141594125195191	1.4716e-06
20	3.141596553704820	3.9001e-06
21	3.141596553704820	3.9001e-06
22	3.141674265021758	8.1611e-05
23	3.141829681889202	2.3703e-04
24	3.142451272494134	8.5862e-04
25	3.142451272494134	8.5862e-04
26	3.162277660168380	2.0685e-02
27	3.162277660168380	2.0685e-02
28	3.464101615137754	3.2251e-01
29	4.000000000000000	8.5841e-01
30	0	-3.1416e+00
31	0	-3.1416e+00
32	0	-3.1416e+00
33	0	-3.1416e+00

- (a) Explain why the computed p_n is such a poor approximation to π for $n \geq 30$.
- (b) Explain why the error $p_n - \pi$ reaches its smallest magnitude for n about 15.

Note: although rounding error should play a role in your answers to both parts (a) and (b) above, there should be more to your explanation than just saying that there is rounding error in the computation, since there is rounding error in almost all floating-point computations, but most of them are accurate.

2. [5 marks]

Suppose that you wish to generate a series of random numbers X_i , $i = 1, 2, 3, \dots$, from the *Rayleigh distribution*, which has the probability density function

$$f(x) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}$$

for $x \in [0, \infty)$. Assume you have a random number generator, `rand`, that computes random numbers U_i , $i = 1, 2, 3, \dots$, from the uniform distribution on $[0,1]$ (i.e., $U_i \sim \text{Unif}[0,1]$).

Describe how you would compute the X_i .

3. [15 marks: 5 marks for each part]

Suppose you wish to price an Asian option with the payoff function

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

where

$$A(T) = \frac{1}{T} \int_0^T S(t) dt \tag{1}$$

is the average price of a stock over the time period $[0, T]$ and K is the exercise price. Assume that the stock price, $S(t)$, evolves in the *risk-neutral world* according to the SDE

$$dS = rSdt + \sigma SdW \tag{2}$$

where $W(t)$ is a standard Brownian motion, $r = 0.04$ is the risk-free interest rate, $\sigma = 0.30$ is the volatility, $T = 0.25$ is the time to expiry, $K = 90$ is the exercise price, and $S(0) = 100$ is the initial stock price.

Let M be a positive integer, $\Delta t = T/M$ and $t_m = m\Delta t$, for $m = 0, 1, 2, \dots, M$. You can compute the solution of the SDE (2) at the points t_m , for $m = 1, 2, \dots, M$, by

$$S(t_m) = S(t_{m-1}) \exp \left(\left(r - \frac{\sigma^2}{2} \right) \Delta t + \sigma \sqrt{\Delta t} Z_m \right) \tag{3}$$

where $Z_m \sim N(0, 1)$ is a standard normal random variable. Then, using these $S(t_m)$, $m = 0, 1, \dots, M$, you can approximate the integral in (1) by the *trapezoidal rule*:

$$\int_0^T S(t) dt \approx \frac{\Delta t}{2}(S(t_0) + S(t_M)) + \Delta t \sum_{i=1}^{M-1} S(t_m). \quad (4)$$

- (a) Assuming that you have a random number generator, `randn`, that computes standard normal random variables, write a computer program that uses the Monte Carlo method to compute an approximation to the price of this Asian option.

Your program should also determinate a 95% confidence interval for your approximation to the price of the option. In determining the confidence interval, ignore the bias introduced by the trapezoidal rule for now.

Your program will probably have an outer loop that generates N Monte Carlo samples and an inner loop that computes $S(t_m)$ for $m = 1, 2, \dots, M$ by (3) and approximates the integral in (1) by the trapezoidal rule (4). Leave N and M as free parameters for now.

(Your program does not have to be syntactically correct, but it should be clear enough so that I can understand what you want to compute.)

- (b) Suppose that the trapezoidal rule (4) has convergence order $p = 1/2$. That is,

$$\frac{\Delta t}{2}(S(t_0) + S(t_M)) + \Delta t \sum_{i=1}^{M-1} S(t_m) = \int_0^T S(t) dt + \mathcal{O}((\Delta t)^p)$$

for $p = 1/2$ and the equation above does not hold for $p > 1/2$. How would you relate the M and N that govern your inner and outer iterations in part (a) for an efficient Monte Carlo simulation?

Justify your answer.

- (c) If you require a tight confidence interval on your computed approximation to the price of the Asian option, your program will probably run for a very long time. Describe a *variance reduction technique* that you could use to reduce the running time. Explain why you believe your variance reduction technique will be effective in this context.

4. [10 marks: 5 marks for each part]

Consider the *transport equation*

$$u_t(t, x) + a u_x(t, x) = 0 \quad (5)$$

where $a > 0$ is a constant. An effective numerical method for solving this hyperbolic PDE is the *Lax-Wendroff* formula

$$U_n^{m+1} = U_n^m - \frac{\rho a}{2}(U_{n+1}^m - U_{n-1}^m) + \frac{\rho^2 a^2}{2}(U_{n+1}^m - 2U_n^m + U_{n-1}^m) \quad (6)$$

where $\Delta t > 0$, $\Delta x > 0$, $\rho = \Delta t / \Delta x > 0$, $U_n^m \approx u(m\Delta t, n\Delta x)$ and $u(m\Delta t, n\Delta x)$ satisfies (5) at $t = m\Delta t$ and $x = n\Delta x$.

(a) Show that the Lax-Wendroff formula (6) is second-order consistent in both time and space.

Hint: you may find it helpful to show that the solution to the PDE (5) also satisfies $u_{tt}(t, x) - a^2 u_{xx}(t, x) = 0$ (assuming u is sufficiently smooth) and use this in your determination of the order of consistency.

(b) The Lax-Wendroff formula (6) is stable if and only if $r = \rho a \leq 1$.

(Note $r > 0$, since $a > 0$ and $\rho > 0$, as noted above.)

We didn't learn enough about stability analysis in this course to show that the Lax-Wendroff formula is stable if $r = \rho a \leq 1$, but we did learn enough to show that it is not stable if $r = \rho a > 1$.

Show that the Lax-Wendroff formula (6) is not stable if $r = \rho a > 1$.

Hint: consider the initial condition $U_n^0 = (-1)^n \epsilon$ for all n associated with the x -domain of (6) and for some small constant ϵ .