UNIVERSITY OF ESWATINI

FACULTY OF SCIENCE

DEPARTMENT OF PHYSICS

MAIN EXAMINATION: 2020

TITLE OF THE PAPER: COMPUTATIONAL PHYSICS-II

COURSE NUMBER: PHY482/P482

TIME ALLOWED:
SECTION A: 1 HOUR
SECTION B: 2 HOURS

INSTRUCTIONS:
THE ARE TWO SECTIONS IN THIS PAPER:

- **SECTION A** IS A WRITTEN PART. ANSWER THIS SECTION ON THE ANSWER BOOK. IT CARRIES A TOTAL OF **40** MARKS.
- **SECTION B** IS A PRACTICAL PART FOR WHICH YOU WILL WORK ON A PC AND SUBMIT THE PRINTED OUTPUT. IT CARRIES A TOTAL OF **60** MARKS.
- You may proceed to do Section B only after you have submitted your answer book for Section A

Answer **all** the questions from **section A** and **all** the questions from **section B**. Marks for different sections of each question are shown in the right hand margin.

THE PAPER HAS 5 PAGES, INCLUDING THIS PAGE.

DO NOT OPEN THIS PAGE UNTIL PERMISSION HAS BEEN GIVEN BY THE INVIGILATOR

Section A

Question 1

(a) **(3 marks)** Consider the basic algorithm for a random number generator given as

$$x_{n+1} = (ax_n + c) \mod m$$

where x_{n+1} and x_n are respectively the $(n+1)^{\text{th}}$ and n^{th} numbers in the sequence. Take $a = 5$, $c = 2$, $m = 8$ and $x_0 = 1$ and generate the first few random numbers. Does the sequence $\{x_n\}$ look like a random sequence. Give two characteristics of a random sequence with a uniform distribution.

(b) **(5 marks)** Consider the outcome from throwing a fair *coin* 100 times, given as

$$THHHTTTTHT \ldots T.$$

with $H \equiv$ Head and $T \equiv$ Tail. Write a program that simulates the same random process throwing a fair coin using a random number generator.

(c) **(2 marks)** Give two applications of Monte Carlo methods in computational physics.

Question 2

a) **(5 marks)** Write a program that calculates the value of ln(2) using the series representation

$$L = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \ldots = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} \tag{1}$$

and terminate the calculation when the next term to be added is smaller than a predetermined fraction of the sum, i.e., 10^{-5}.

b) **(5 marks)** The diffusion equation for a density field $c(x,t)$ is given by a partial differential equation

$$\frac{\partial c(x,t)}{\partial t} = D\frac{\partial^c(x,t)}{\partial^2 x} \tag{2}$$

where D is a diffusion constant. Given the initial and boundary condition, this equation can be solved numerically using the finite difference schemes. Show that alternatively, the Eq. (2) can be transformed into a first-order *ordinary differential equation* by the application of a Fourier transform.

Question 3

a) **(4 marks)** The rectangle and trapezoid methods of integration give identical results for the integral of any function $f(x)$ in the interval $[a, b]$, if $f(a) = f(b)$. In other words, the two methods have the same error with respect to the correct value, even though they have different rates of convergence. How do you reconcile these two results?

b) **(4 marks)** How would you handle numerically these improper integrals?

$$\text{(i)} \quad \int_0^1 dx \, \frac{e^x}{\sqrt{x}} \qquad \text{(ii)} \quad \int_0^\pi \frac{\sin x}{x} \tag{3}$$

c) **(2 marks)** Name one condition under which the Monte Carlo integration method is most suitable compared to the standard numerical methods such as the Trapezoidal and the Simpson rule.

Question 4

a) ODEs: -You want to write a Euler algorithm to solve Newton's second law for the oscillation exhibited by a body mass m that is hanging from a spring of constant k in a vertical gravitational field with gravitational constant g. If y is the displacement of the spring from its equilibrium position, the differential equation for the motion of the mass is

$$\frac{d^2y}{dt^2} = -g - ky$$

(i) **(3 marks)** Convert this equation into two first -order differential equations written in terms of dimensionless quantities. These are equations that you will need to implement with your algorithm.

(ii) **(2 marks)** How many initial conditions will you need to solve this problem? Which ones?

b) **(5 marks)** The general one-dimensional Poisson equation reads as

$$-u''(x) = f(x), \quad x \in (0, 1), \quad u(0) = u(1) = 0$$

Show that this equation in a discretized approximations can be given as

$$-\frac{u_{i+1} + v_{i-1} - 2u_i}{h^2} = f_i \quad \text{for } i = 0, \ldots, N$$

where $f_i = f(x_i)$, $u_i = u(x_i)$, with the grid points $x_i = ih$ and the grid spacing $h = 1/N$

Section B

The answers to this question must include the computer code and output, in addition to any writing that might be needed.

Question 5

Consider the following nonlinear logistic map

$$x_{i+1} = a - x_i^2$$

where a is a constant.

(a) **(20 marks)** Write a program that calculates the trajectory of the system whose dynamics are governed by the logistic map given that the initial point $x_0 = 0.5$. Compute the trajectories for three cases $a = 0.5, 1.476$ and 2.0, let say 100 iterations.

(b) **(10 marks)** Plot graphs of x_i versus i for the different values of a. Does changing the value of a change the dynamics of the system. Describe the nature of the states observed.

Question 6

The dynamic behavior of a ferromagnet after a sudden change of temperature is often given by the following continuous equation

$$\frac{\partial m(x)}{\partial t} = \frac{\partial^2 m(x)}{\partial x^2} + rm(x) - m^3(x)$$

where $m(x,t)$ in the magnetization, and $r \propto (T/Tc - 1)$ where T is temperature and T_c is the critical point where the ferromagnet loses its magnetic properties. The corresponding discretized version of the dynamic equation is given as

$$m_i^{n+1} = m_i^n + \frac{\Delta t}{(\Delta x)^2} \cdot (m_{i+1}^n + m_{i-1}^n - 2m_i^n) + \Delta t[rm_i^n - (m_i^n)^3] \tag{4}$$

where $m_i^n = m(i \cdot \Delta x, n \cdot \Delta t)$. The code $ferroM.f95$ implements the above difference method to calculate the dynamics of a system at $r = -0.5$ starting a paramagnetic state (random initial conditions) and applying periodic boundary conditions.

(i) **(10 marks)** Run the code with a different seed of the random number generator. Plot the configuration of the magnetization $m(x, t = 0)$, i.e., m_i^0 for $i = 1$ to 100.

(ii) **(20 marks)** Modify your code in order to compute the average magnetization of the system at each given time step: $\bar{m}(t_n) = \frac{1}{100}\sum_{i=1}^{100} m(x_i, t_n)$. Plot $\bar{m}$ vs t_n, for n =1 to 5000 for two cases: one with $r < 0$ and the other $r > 0$. Choose $|r| < 1$ in both cases. Discuss your results, which value of r leads to a paramagnetic or ferromagnetic state.