XIAMEN UNIVERSITY MALAYSIA

Practical Test

Course Code:	BSC128
Course Name:	Numerical Methods
Question Paper Setter:	Liu Meifeng
Academic Session:	2024/04 Question Paper: A ■ B ☐
Total No. of Pages:	3 Time Allocated: 2 hours
Additional Materials:	Algorithm 2.3, Algorithm 3.1, Algorithm 3.3
Apparatus Allowed:	Computer, Laptop

INSTRUCTIONS TO CANDIDATES

1. This paper consists of Three questions. Please answer ALL questions.
2. Read the above information carefully to ensure you have the correct and complete question paper.
3. Please follow the requirement of each section and write down all the corresponding answers on the answer book provided.
4. Please note that presentation of solutions (clarity, coherence, conciseness, and completeness) is important. Show your work and organize your solution. Answer without proper justification may receive less, or even zero credit.
5. Communication between candidates in any means is forbidden. Answers must be entirely individual candidate's independent effort. If you are found sharing your solutions with other candidates, or suspected of doing so, you would be penalized accordingly.

DO NOT TURN OVER THIS PAGE UNTIL INSTRUCTED TO DO SO.

(Student ID: _______________   Full Name: ___________________________________)

Question 1 (30 marks)

The Maclaurin series for the arctangent function, $\tan^{-1} x$, will be convergent for $-1 < x \leq 1$ and is given by

$$\arctan x = \lim_{n\to\infty} P_n(x) = \lim_{n\to\infty} \sum_{i=1}^{n} (-1)^{i+1} \frac{x^{2i-1}}{2i-1}.$$

(a) Generate a table for the value of $P_n(\pi/4)$ with different numbers of collecting terms, $n = 1,2,3,\ldots 10$, and compare the results with its exact solution by calculating the absolute error and relative errors for different n. Please print your results in formatted output.

(b) Use the fact that $\tan \pi/4 = 1$ to determine the number of n terms for the Maclaurin series that need to be summed to ensure that $|4P_n(1) - \pi| < 10^{-3}$. You can try your code in (a) with increasing values of n until the required accuracy is reached.

(c) The C++ programming language requires the value of π to be within 10^{-10}. How many terms of the series would we need to sum in order to obtain this degree of accuracy? Refer to the following algorithm for reference, in which the bound for alternating series is used.

INPUT value x, tolerance TOL, maximum number of iterations M.
OUTPUT degree N of the polynomial or a message of failure.
Step 1 Set $N = 1$;
$y = x - 1$;
$SUM = 0$;
$POWER = y$;
$TERM = y$;
$SIGN = -1$. (*Used to implement alternation of signs.*)

Step 2 While $N \leq M$ do Steps 3–5.

Step 3 Set $SIGN = -SIGN$; (*Alternate the signs.*)
$SUM = SUM + SIGN \cdot TERM$; (*Accumulate the terms.*)
$POWER = POWER \cdot y$;
$TERM = POWER/(N+1)$. (*Calculate the next term.*)

Step 4 If $|TERM| < TOL$ then (*Test for accuracy.*)
OUTPUT (N);
STOP. (*The procedure was successful.*)

Step 5 Set $N = N + 1$. (*Prepare for the next iteration.*)

Question 2 (30 marks)

The growth of a population can often be modelled over short periods of time by assuming that the population grows continuously with time at a rate proportional to the number present at that time. Suppose that $N(t)$ denotes the number in the population at time t and λ denotes the constant birth rate of the population. Then the population satisfies the following differential equation

$$\frac{dN(t)}{dt} = \lambda N(t),$$

whose solution is $N(t) = N_0 e^{\lambda t}$, where N_0 denotes the initial population. This exponential model is valid only when the population is isolated, with no immigration. If immigration is permitted at a constant rate v, then the differential equation becomes

$$\frac{dN(t)}{dt} = \lambda N(t) + v,$$

whose solution is

$$N(t) = N_0 e^{\lambda t} + \frac{v}{\lambda}(e^{\lambda t} - 1).$$

Suppose a certain population contains $N(0) = 1{,}000{,}000$ individuals initially, that 435,000 individuals immigrate into the community in the first year, and that $N(1) = 1{,}564{,}000$ individuals are present at the end of one year.

(a) Write a MatLab/Octave code setting the corresponding values for the parameter N_0, v and t, construct an equation based on the given conditions at the end of the first year.

(b) To determine the birth rate of this population, we need to solve the value of λ based on the equation obtained in (a). Please define a function $F(\lambda)$ such that $F(\lambda) = 0$ based on (a).

(c) Use Newton's method (Algorithm 2.3) with appropriate initial guess to find an approximation for the birth rate of population to within 10^{-4} accuracy.

(d) Use the approximation obtained in (c) to predict the population at the end of the second year, assuming that the immigration rate during this year remains the same.

Question 3 (40 marks)

Given the function $f(x) = e^{0.1x^2}$.

(a) Construct a table listing the data of function values and function derivatives as follows.

x	$f(x)$	$f'(x)$
1		
1.5		
2		
3		

(b) Apply Neville's method (Algorithm 3.1) to the given data by constructing a recursive table to approximate $f(1.25)$. Evaluate the absolute error.

(c) Use the Hermite Interpolating Method (Algorithm 3.3) by constructing a recursive table to approximate $f(1.25)$ by using $H_3(1.25)$ where H_3 uses the nodes $x_0 = 1$ and $x_1 = 1.5$. Evaluate the absolute error.

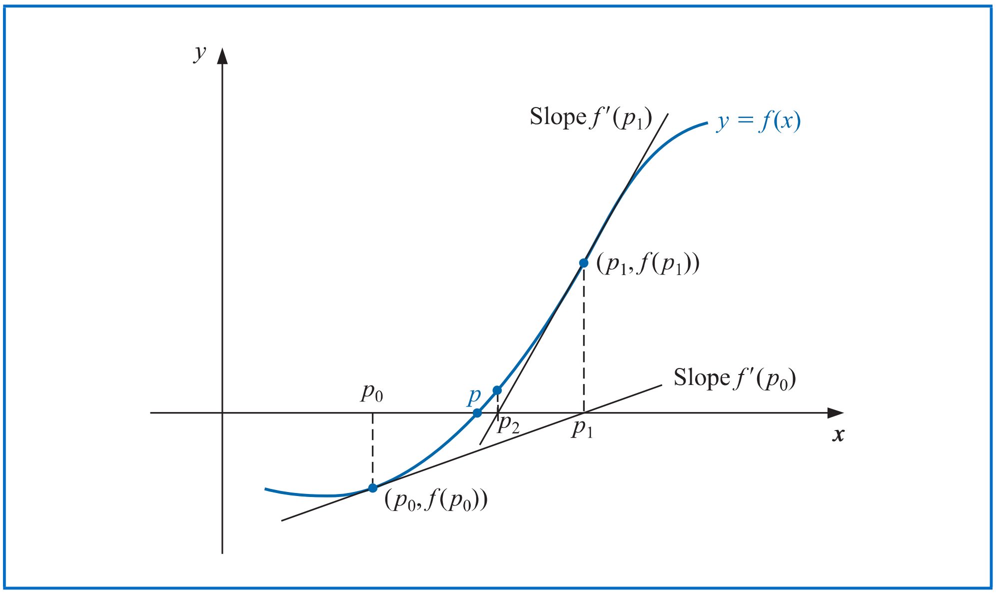
(d) Use the Hermite Interpolating Method (Algorithm 3.3) by constructing a recursive table to approximate $f(1.25)$ by using $H_5(1.25)$ where H_5 uses the nodes $x_0 = 1$, $x_1 = 2$, and $x_2 = 3$. Evaluate the absolute error.

Instructions:

- Save your script in Octave Online as **PracticalQ#_ID.m**, download it and print it as **PracticalQ#_ID.pdf**.
- Take a screenshot on your code and results, save the screenshot as **PracticalQ#_ID.png**.
- Share your script as Octave Bucket, copy the link and paste it on the comment field.
- Remember to attach the **PDF** file, the **M** file and the **PNG** file of your code in the Moodle system in order to complete your submission.

- END OF PAPER -

Figure 2.8



ALGORITHM 2.3

Newton's

To find a solution to $f(x) = 0$ given an initial approximation p_0:

INPUT initial approximation p_0; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$.

Step 2 While $i \leq N_0$ do Steps 3–6.

Step 3 Set $p = p_0 - f(p_0)/f'(p_0)$. (*Compute* p_i.)

Step 4 If $|p - p_0| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$.

Step 6 Set $p_0 = p$. (*Update* p_0.)

Step 7 OUTPUT ('The method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure was unsuccessful.*)
STOP. ■

The stopping-technique inequalities given with the Bisection method are applicable to Newton's method. That is, select a tolerance $\varepsilon > 0$, and construct $p_1, \ldots p_N$ until

$$|p_N - p_{N-1}| < \varepsilon, \tag{2.8}$$

$$\frac{|p_N - p_{N-1}|}{|p_N|} < \varepsilon, \quad p_N \neq 0, \tag{2.9}$$

or

$$|f(p_N)| < \varepsilon. \tag{2.10}$$

In the preceding example, we have $f(2.1) = \ln 2.1 = 0.7419$ to four decimal places, so the absolute error is

$$|f(2.1) - P_2(2.1)| = |0.7419 - 0.7420| = 10^{-4}.$$

However, $f'(x) = 1/x$, $f''(x) = -1/x^2$, and $f'''(x) = 2/x^3$, so the Lagrange error formula (3.3) in Theorem 3.3 gives the error bound

$$\begin{aligned} |f(2.1) - P_2(2.1)| &= \left| \frac{f'''(\xi(2.1))}{3!}(x - x_0)(x - x_1)(x - x_2) \right| \\ &= \left| \frac{1}{3\,(\xi(2.1))^3}(0.1)(-0.1)(-0.2) \right| \le \frac{0.002}{3(2)^3} = 8.\overline{3} \times 10^{-5}. \end{aligned}$$

Notice that the actual error, 10^{-4}, exceeds the error bound, $8.\overline{3} \times 10^{-5}$. This apparent contradiction is a consequence of finite-digit computations. We used four-digit rounding arithmetic, and the Lagrange error formula (3.3) assumes infinite-digit arithmetic. This caused our actual errors to exceed the theoretical error estimate.

- Remember: You cannot expect more accuracy than the arithmetic provides.

Algorithm 3.1 constructs the entries in Neville's method by rows.

ALGORITHM 3.1

Neville's Iterated Interpolation

To evaluate the interpolating polynomial P on the $n+1$ distinct numbers $x_0, \ldots, x_n$ at the number x for the function f:

INPUT numbers $x, x_0, x_1, \ldots, x_n$; values $f(x_0), f(x_1), \ldots, f(x_n)$ as the first column $Q_{0,0}, Q_{1,0}, \ldots, Q_{n,0}$ of Q.

OUTPUT the table Q with $P(x) = Q_{n,n}$.

Step 1 For $i = 1, 2, \ldots, n$
for $j = 1, 2, \ldots, i$

$$\text{set } Q_{i,j} = \frac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{x_i - x_{i-j}}.$$

Step 2 OUTPUT (Q);
STOP. ■

EXERCISE SET 3.2

1. Use Neville's method to obtain the approximations for Lagrange interpolating polynomials of degrees one, two, and three to approximate each of the following:
   a. $f(8.4)$ if $f(8.1) = 16.94410$, $f(8.3) = 17.56492$, $f(8.6) = 18.50515$, $f(8.7) = 18.82091$
   b. $f\left(-\frac{1}{3}\right)$ if $f(-0.75) = -0.07181250$, $f(-0.5) = -0.02475000$, $f(-0.25) = 0.33493750$, $f(0) = 1.10100000$
   c. $f(0.25)$ if $f(0.1) = 0.62049958$, $f(0.2) = -0.28398668$, $f(0.3) = 0.00660095$, $f(0.4) = 0.24842440$
   d. $f(0.9)$ if $f(0.6) = -0.17694460$, $f(0.7) = 0.01375227$, $f(0.8) = 0.22363362$, $f(1.0) = 0.65809197$

ALGORITHM
3.3

Hermite Interpolation

To obtain the coefficients of the Hermite interpolating polynomial $H(x)$ on the $(n+1)$ distinct numbers $x_0, \ldots, x_n$ for the function f:

INPUT numbers $x_0, x_1, \ldots, x_n$; values $f(x_0), \ldots, f(x_n)$ and $f'(x_0), \ldots, f'(x_n)$.

OUTPUT the numbers $Q_{0,0}, Q_{1,1}, \ldots, Q_{2n+1,2n+1}$ where

$$\begin{aligned} H(x) = Q_{0,0} &+ Q_{1,1}(x-x_0) + Q_{2,2}(x-x_0)^2 + Q_{3,3}(x-x_0)^2(x-x_1) \\ &+ Q_{4,4}(x-x_0)^2(x-x_1)^2 + \cdots \\ &+ Q_{2n+1,2n+1}(x-x_0)^2(x-x_1)^2 \cdots (x-x_{n-1})^2(x-x_n). \end{aligned}$$

Step 1 For $i = 0, 1, \ldots, n$ do Steps 2 and 3.

Step 2 Set $z_{2i} = x_i$;
$z_{2i+1} = x_i$;
$Q_{2i,0} = f(x_i)$;
$Q_{2i+1,0} = f(x_i)$;
$Q_{2i+1,1} = f'(x_i)$.

Step 3 If $i \neq 0$ then set

$$Q_{2i,1} = \frac{Q_{2i,0} - Q_{2i-1,0}}{z_{2i} - z_{2i-1}}.$$

Step 4 For $i = 2, 3, \ldots, 2n+1$
for $j = 2, 3, \ldots, i$ set $Q_{i,j} = \dfrac{Q_{i,j-1} - Q_{i-1,j-1}}{z_i - z_{i-j}}$.

Step 5 OUTPUT $(Q_{0,0}, Q_{1,1}, \ldots, Q_{2n+1,2n+1})$;
STOP. ■

EXERCISE SET 3.4

1. Use Theorem 3.9 or Algorithm 3.3 to construct an approximating polynomial for the following data.

a.

x	$f(x)$	$f'(x)$
8.3	17.56492	3.116256
8.6	18.50515	3.151762

b.

x	$f(x)$	$f'(x)$
0.8	0.22363362	2.1691753
1.0	0.65809197	2.0466965

c.

x	$f(x)$	$f'(x)$
−0.5	−0.0247500	0.7510000
−0.25	0.3349375	2.1890000
0	1.1010000	4.0020000

d.

x	$f(x)$	$f'(x)$
0.1	−0.62049958	3.58502082
0.2	−0.28398668	3.14033271
0.3	0.00660095	2.66668043
0.4	0.24842440	2.16529366

2. Use Theorem 3.9 or Algorithm 3.3 to construct an approximating polynomial for the following data.

a.

x	$f(x)$	$f'(x)$
0	1.00000	2.00000
0.5	2.71828	5.43656

b.

x	$f(x)$	$f'(x)$
−0.25	1.33203	0.437500
0.25	0.800781	−0.625000

c.

x	$f(x)$	$f'(x)$
0.1	−0.29004996	−2.8019975
0.2	−0.56079734	−2.6159201
0.3	−0.81401972	−2.9734038

d.

x	$f(x)$	$f'(x)$
−1	0.86199480	0.15536240
−0.5	0.95802009	0.23269654
0	1.0986123	0.33333333
0.5	1.2943767	0.45186776