National University of Computer & Emerging Sciences, Karachi
FAST, School of Computing,
Spring, 2023
Final Exam
26th May, 2023, 08:30 am – 11:30am

Course Code: CS 2008	**Course Name: Numerical Computing**
Instructor Names: Ms. Amber Shaikh, Mr. Usama Antuley, Mr Shahid Ashraf and Mr. Moheez Rahim	
Student Roll No:	**Section :**

- Solve all questions and return the question paper.
- Read each question completely before answering it.
- This paper consists of 7 pages and 10 Questions.
- All the required formulas are on page no 7.

Time: 180 minutes. **Total Marks:**100

Qno1 – CLO 3 Estimated Time 30 Minutes 15 Marks

Select the correct option, In all given codes assume all the required libraries have been called.

I)-If **A**= $\begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{7} & \mathbf{8} & \mathbf{9} \end{bmatrix}$ what will be value of submatrix?

submatrix = np.delete (np.delete(A, 1, axis=0), 2, axis=1)

a) $\begin{bmatrix} 1 & 3 \\ 7 & 9 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 2 \\ 7 & 8 \end{bmatrix}$ c) $\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$ d) $\begin{bmatrix} 1 & 3 \\ 4 & 6 \end{bmatrix}$ e) $\begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$

(II-VI) will be from below code

```
def  Method(A, b, X): #where AX=b is a system of linear equations
   k = 0
   while k <= 3:
    k += 1
    X0=X.copy()
    for i in range(len(b)):
      Sum = 0  # reset Sum to zero before the inner loop
      for j in range(len(b)):
         if j != i:
            Sum += A[i,j] * X0[j]
      X[i] = (1 / A[i,i]) * (b[i] - Sum)
   return(X)
```

II) What will be value of Sum for **i=0**, **j=1** and **k=2** while solving AX=b

where A= $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, b = \begin{bmatrix} 14 \\ 5 \\ -8 \end{bmatrix}$ and $X^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$.

a) 65 b)0 c)2 d)112 e) 100

III) If $X^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ is the initial vector, using above code what will be $X^{(1)}$.

a) $\begin{bmatrix} 14 \\ 1 \\ -0.8889 \end{bmatrix}$ b) $\begin{bmatrix} 14 \\ 1 \\ 0.8889 \end{bmatrix}$ c) $\begin{bmatrix} 1 \\ 14 \\ 0.8889 \end{bmatrix}$ d) $\begin{bmatrix} 14 \\ 15 \\ 0.8889 \end{bmatrix}$ e) $\begin{bmatrix} 14 \\ 17 \\ 0.8889 \end{bmatrix}$

IV) Which iterative method above code is referring to?

a) LU Decomposition c) Gauss Siedel d) Cholesky decomposition d) Jacobi Iterative
e) Newton Raphson

V) How many changes does this code required to convert it to the Jacobi or Gauss Siedel iterative method to solve AX=b?

a) 0 b) 1 c) 2 d) 3 e) 4

VI) How many times does this code will update X?

a) 2 b) 1 c) 4 d) 5 e) 3

(VII-XII) will be from below code

```
def LU_Decompose(A):
  n = len(A)
  L = np.zeros((n, n))
  U = np.zeros((n, n))
  for i in range(n):
    for k in range(i, n):
      s1 = sum((L[i][j] * U[j][k]) for j in range(i))
      U[i][k] = A[i][k] – s1
    for k in range(i, n):
      if i == k:
        L[i][i] = 1
      else:
        s2 = sum((L[k][j] * U[j][i]) for j in range(i))
        L[k][i] = (A[k][i] – s2) / U[i][i]
  return L, U
```

VII) Which LU decomposition method above code is referring to?

a) Dolittle b) Crout c) Cholesky d) LDL^t e) Secant

VIII- Perform LU decomposition method on the A=$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ by the algorithm in above code to select L from the given options?

a) L=$\begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ 7 & 8 & 1 \end{bmatrix}$ b) L=$\begin{bmatrix} 1 & 0 & 0 \\ 5 & 1 & 0 \\ 7 & 21 & 1 \end{bmatrix}$ c) L=$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 8 & 1 \end{bmatrix}$ d) L=$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{bmatrix}$ e) L=$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & -2 & 1 \end{bmatrix}$

IX- Select conditions on which the above code will not work for desired LU decomposition of a matrix?

I) If matrix is singular
II) If matrix is diagonally dominant
III) If matrix is non singular but need row exchanges while converting it to echelon form.

a) I and II   b) I and III   c) I, II and III   d) none   e) only I   f) Only III

X-- For i= 1and k=1 state the value of s1 produce by above code.

a) 2   b)0   c) 8   d)6   e) 9

XI- What will be the value of U[1,1],U[1,2] and U[2,3] by using above code for A=$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

a)1, -1,6   b) 1,-3,-6   c) 1, -2, 6   d) -1,3,6   e) 1, 2, -6

XII- For i=2 what will be the values of k and j.

a) k=0, 1 and j= 2   b)k=2 and j=0,1   c) k=2,3 and j=0,1,2   d) k= 1,2 and j=0,1,2   e) k=0,1,2 and j=1,2

(XIII-XV) will be from below code

```
def poly(x, y):
   n = len(x)
   p = np.poly1d(0.0)
   for i in range(n):
      L = np.poly1d(y[i])
      for j in range(n):
         if j != i:
            L *= np.poly1d([1.0, -x[j]]) / (x[i] - x[j])
      p += L
   return p
x = [0, 1, 2]
y = [12, 14, 15]
p = poly(x, y)
```

XIII- What will be the coefficient of x^3 by using above code.

a)3   b)9   c)2   d)5   e)0

XIV- Predict value of y at x=1.5 by using polynomial from above code.

a) 15.625   b)14.625   c) 17.625   d) 18.625

XV- Which algorithm above code is referring to?

a) Lagrange Polynomial   b) Newton divided difference   c) Newton forward difference   d) Newton backward difference

Qno2– CLO 3 Estimated Time 10 Minutes 3+2 Marks

Write code of python using given algorithm also give an example of problem that can be solved by the given code.

1. Initialize the initial value of y and the initial time t.
2. Set the number of steps n = (t_final - t_initial) / h.
3. Loop from i = 1 to i = n:
   a. Calculate
      k1 = f(t, y)
      k2 = f(t+h, y+k1*h)
   b) Calculate the new value of y using the formula:
      y(i) = y(i-1) + (k1/2 + k2/2)*h
   c) Update the value of time t(i) = t(i-1) + h
4. Return the values of time and y at each step.

Note: In the above algorithm y'= f(t, y) represents the differential equation, and h is the step size. The values of k1 and k2 are the slopes at different points in the interval, which are used to estimate the value of y at the next step.

Qno3– CLO 2 Estimated time 10 minutes 5+5 Marks

a) Find the truncation error of $e^{0.5}$ when the approximation of $e^{0.5}$ is the summation of the first four terms of the McLaurin's (special case of Taylor's series when x_0 =0) series of $e^{0.5}$.

b) Evaluate $f(t) = t^3 - 6.1t^2 + 3.2t + 1.5 \; at \; t = 4.71$ using three-digit arithematic (rounding and chopping). Assume the true value $f(4.71) = -14.263899$ calculate relative error for both chopping and rounding.

Qno4 – CLO 2 Estimated time 20 minutes 5+2+3 Marks

a) With the help of the below mentioned nodes, find the second Lagrange interpolating poynomial for $f(t) = \frac{1}{t}$; $t_0 = 2, \quad t_1 = 2.75 \; and \quad t_2 = 4$

b) A computer scientist wants to estimate the total time it would take to process a large dataset using different algorithms. The time taken by each algorithm is given by the following function: $\boldsymbol{T(n) = 2n^4 + 5n + 10}$, where *n* is the size of the dataset.

i) Using simple Trapezoidal Rule and simple Simpson's Rule estimate the total processing time for the dataset if the size of the dataset ranges from 0 to 2.
ii) Also calculate bound error for simple Trapezoidal and Simpson's rule and compare with actual error

Qno5 – CLO 2 Estimated time 20 minutes 10 Marks

A computer scientist is studying the performance of a computer algorithm and wants to estimate the execution time (in seconds) based on the input size (in number of elements). The scientist collects data from three different experiments, where the input size and execution time are recorded. The dataset is as follows:

Input Size (x)	2	3	4
Expected time (y)	7	10	13

In this dataset, the scientist has measured the execution time for different input sizes. The input size (x) represents the number of elements in the dataset, while the execution time (y) represents the time taken by the algorithm to process the dataset.

The scientist needs assistance to apply gradient descent method on the above data set to find the optimal parameters for the function. For this you are supposed to help him by considering the following:

$h_\theta = \theta_0 x_0 + \theta_1 x_{1,}\ where\ let\ \boldsymbol{x_0} = \mathbf{1}$

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{i=m}(h_{\theta_0\theta_1}(x_{1j)} - y_j)^2 \quad let\ \boldsymbol{\theta_0} = \boldsymbol{\theta_1} = \mathbf{0}(initially)$$

$\theta_j = \theta_j - \propto \frac{\partial y}{\partial \theta_j} \quad let\ \propto = \mathbf{0.05}$ **where,** m = no of data points

Note: Just perform three iterations.

Qno 6 – CLO 2 Estimated Time 15 Minutes 4+6 Marks

For the following linear system,

$$\begin{aligned} -3x_1 + x_2 + 12x_3 &= 5 \\ 6x_1 - x_2 - x_3 &= 31 \\ 6x_1 + 9x_2 + x_3 &= 4 \end{aligned}$$

(i) Compute true/actual solution of above system(Use Calculator).

(ii) Without any arrangement, solve the above system numerically by using Gauss-Seidal method with $X^{(0)} = (0,0,0)^t$ and perform three iterations. Compare the results with part(i) and state whether the solution of above system is convergent or not for Gauss-Seidal Method.

Note: If the answer of above part ((ii)) is divergent then work on the below part ((iii)) otherwise leave it.

(iii) Solve the above system again by by Gauss-Seidal method with $X^{(0)} = (0,0,0)^t$ and stops when $\|X^{(k+1)} - X^{(k)}\|_\infty < 0.01$ after making a suitable arrangement that can lead to a convergent solution.

Qno7 – CLO 2 Estimated Time 15 Minutes 4+6 Marks

For the following linear system,

$$\begin{aligned} x_1 + x_2 + x_3 &= 2 \\ x_1 + x_2 &= 3 \\ x_1 + x_3 &= 4 \end{aligned}$$

a) Write Co-efficient matrix A for above system

b) Given that absolute dominant eigen value of A is $\lambda_{Absolute\ dominant}^{(Actual\ value)} = \sqrt{2} + 1$, find the approximate absolute dominant eigen value $\lambda_{Absolute\ dominant}^{(Approximated\ value)}$ by using Power method with $X^{(0)} = (-1,0,1)^t$ and stops when $\left|\boldsymbol{\lambda}_{\boldsymbol{Absolute\ dominant}}^{(\boldsymbol{Actual\ value})} - \boldsymbol{\lambda}_{\boldsymbol{Absolute\ dominant}}^{(\boldsymbol{Approximated\ value})}\right| < \mathbf{0.05}$

Qno8 – CLO 2 Estimated Time 10 Minutes 4+1+1+2+2 Marks

For the following linear system,
$$\begin{aligned} x_1 + 2x_2 - x_3 &= 2 \\ 2x_1 + 4x_2 - x_3 &= 3 \\ x_1 + 2x_3 &= 2 \end{aligned}$$

(i) Write the co-efficient matrix A for above system and find the upper triangular matrix U corresponding to A by applying row elementary operations on A.

(ii) With the help of part (i), write the permutation matrix P

(iii) If there is no row interchange performed in part(i) then $P =$ ____ matrix.

(iv) With the help of part(i) construct lower triangular matrix L ($l_{ii} = 1, for\ all\ i$) such that $PA = LU$

(v) Verify $PA = LU$

Qno9 – CLO 2 Estimated Time 25 Minutes 2+3+1+3+1 Marks

Consider, $A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 3 & 2 \\ 1 & 2 & 4 \end{bmatrix}, B = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 4 & 2 \\ 0 & 2 & 2 \end{bmatrix}\ and\ C = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 2 \\ 0 & 2 & -4 \end{bmatrix}$

(i) Identify from matrices A, B and C which one is positive definite?

(ii) Decompose the matrix (answer of part(i)) into LDL^T where $l_{ii} = 1\ and\ d_{ii} > 0\ for\ all\ i$

(iii) Verify part(ii) decomposition

(iv) Decompose the matrix (answer of part(i)) into $Cholesky\ decomposition\ (LL^T)\ wher\ l_{ii} \neq 0, for\ all\ i)$

(v) Verify part(iv) decomposition

Qno10 – CLO 2 Estimated Time 25 Minutes 2+3+1+3+1 Marks

A particle is moving with velocity $v(t) = \frac{ds}{dt}(1+t) = s^2$, $s(0.5) = -2.46630$, where s is displacement at any time t. Find the displacement $\boldsymbol{s(0.54)}$ with $h = 0.04$ by using

(i) Heun's Method (ii) Midpoint Method and (iii) RK-4 Method (Classical approach).

Given that $s(t) = -\frac{1}{\ln(t+1)}$ is the actual solution, compute actual error in each part (i),(ii) and (iii) also and comments on the result.

THE END

Formula Box	
Euler's Method: $y_{i+1} = y_i + hf(t_i, y_i)$	**Heun's Method: (Special Case of RK-2 methods)** $$y_{i+1} = y_i + \frac{h}{2}(f(t_i, y_i) + f(t_{i+1}, y_i + hf(t_i, y_i))$$
Midpoint Method: (Special case of RK-2 methods) $$y_{i+1} = y_i + h\left(f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}f(t_i, y_i)\right)\right)$$	**RK-4 Classical approach method (special case of RK-4 Methods)** $y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ Where, $k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h)$ $k_3 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h)$ $k_4 = f(t_i + h, y_i + k_3 h)$
Simple Trapezoidal Rule $$\int_{x0}^{x1} f(x)dx = \frac{h}{2}[f(x0) + f(x1)] - \frac{h^3}{12}f''(\xi)$$	**Simple Simpson's Rule** $$\int_{x0}^{x2} f(x)dx = \frac{h}{2}[f(x0) + 4f(x1) + f(x2)] - \frac{h^5}{90}f^{(4)}(\xi)$$
Taylor Polynomial $$P_n(x) = f(x_0) + (x - x_0)f'(x_0) + (x - x_0)^2\frac{f''(x_0)}{2!} + \cdots + (x - x_0)^n\frac{f^{(n)}(x_0)}{n!}$$	**Lagrange Interpolating polynomial** $$P_n(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + \cdots + f(x_n)L_n(x)$$ $$L_0(x) = \frac{(x - x_1)(x - x_2)\dots(x - x_n)}{(x_0 - x_1)(x_0 - x_2)\dots(x_0 - x_n)}$$ $$L_1(x) = \frac{(x - x_0)(x - x_2)\dots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\dots(x_1 - x_n)}$$ $$L_2(x) = \frac{(x - x_0)(x - x_1)(x - x_3)\dots(x - x_n)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)\dots(x_2 - x_n)}$$