Department of Mathematics and Computing

IIT(ISM) Dhanbad

Course Name: **Numerical Methods-Practical**

Course Code: MCC506

Lab Location: Computer Lab II, M&C

General Guidelines/Instructions

Do's:

1. Connect to the server with IP:172.16.64.12
2. Login with your provided username & password only.
3. Update lab file with every lab class and get it checked by the instructors/Teaching assistants (JRF)
4. Use the computer properly to keep it in good working order.

Don't:

1. Don't open the internet browser.
2. Don't change the computer settings.
3. Don't use your flash-drives/hard-drives on laboratory's computer.

Apparatus/Software Required (Common to all):

Computerwith LINUX Operating system andC & C++

SL. NO.	LIST OF LAB PRACTICAL	Expected no. of lab hours
1	Numerical Solution of tri-diagonal system using Thomas algorithm	3
2	Solution of simultaneous non-linear equations using Newton Raphson method	3
3	Solution of simultaneous non-linear equations using Newton Raphson Method(Two variables)	3
4	Numerical solution of central difference interpolation by using Lagrange interpolation method	3
5	Numerical solution of integrals using Trapezoidal method	3
6	Numerical evaluation of integrals using Simpson's method	3
7	Numerical solution of characteristic value problem by Power method.	3
8	Numerical Solution of initial value problem using Euler's method	3
9	Numerical Solution of initial value problem using RungaKutta method	3
10	Numerical Solution for heat equation by suitable method	3
11	Numerical Solution of wave equation by suitable method	3
12	Numerical Solution of Laplace equation in two variables by suitable method	3
13	Practical Exam	3
	Total	**39**

Scheme: Out of total 13 weeks, there will be 12 Lab classes based on above specified topics/problems. Last week will be reserved for practical exam.

Lab 1

Aim: Numerical Solution of tri-diagonal system by using Thomas algorithm

Brief Theory: A *tridiagonal matrix* has nonzero elements only on the main diagonal, the diagonal upon the main diagonal, and the diagonal below the main diagonal. We use Thomas algorithm to get the solution of tridiagonal matrix.

Thomas' algorithm, also called Tridiagonal Matrix Algorithm (TDMA) is essentially the result of applying Gaussian elimination to the tridiagonal system of equations. It gives the values of variables in reverse order upon back substitutions.

Numerical Procedure:

1. Start of the program
2. To solve AX=D, enter the size of array
3. Enter the values of diagonal (b_i), sub diagonal (a_i) and super diagonal (c_i) elements in matrix A
4. Enter the values (d_i) of matrix D
5. Set $\alpha_1 = b_1$
6. Apply the formula $\alpha_i = b_i - \frac{a_i c_{i-1}}{\alpha_{i-1}}, i = 2,3,......,n$
7. Set $\beta_1 = \frac{d_1}{b_1}$
8. Apply the formula $\beta_i = \frac{d_i - a_i \beta_{i-1}}{\alpha_i}, i = 2,3,....,n$
9. Set $x_n = \beta_n$
10. Apply the formula $x_i = \beta_i - \frac{c_i x_{i+1}}{\alpha_i},\ i = n-1, n-2,....$
11. Stop

Examples:

i. Consider the tridiagonal linear system

x1 + 2x2 + 3x3 = -5

-x1 + x3 = -3

3x1 + x2 + 3x3 = -3

ii. Consider the tridiagonal linear system

3x1 - x2 = 2

-x1 + 3x2 - x3 = 1

...

-xn-2 + 3xn-1 - xn = 1

-xn-1 + 3xn = 2

iii. Consider the tridiagonal linear system

$3x_1 - x_2 = -5$

$-x_1 + 3x_2 - x_3 = 7$

$-x_2 + 3x_3 = 7$

Lab 2 –

Aim: Solution of simultaneous non-linear equations by using Newton Raphson method

Brief Theory: The **Newton-Raphson method** (also known as**Newton's method**) is a way to quickly find a good approximation for the root of a real-valued function f(x)=0. It uses the idea that a continuous and differentiable function can be approximated by a straight line tangent to it.

Numerical Procedure:

1. Start
2. Read x, e, n, d
   *x is the initial guess
   e is the absolute error i.e the desired degree of accuracy
   n is for operating loop
   d is for checking slope*
3. Do for i =1 to n in step of 2
4. f = f(x)
5. f1 = f'(x)
6. If ([f1] < d), then display too small slope and goto 11.
   [ ] is used as modulus sign
7. x1 = x – f/f1
8. If ([(x1 – x)/x1] < e), the display the root as x1 and goto 11.
   [ ] is used as modulus sign
9. x = x1 and end loop
10. Display method does not converge due to oscillation.
11. Stop

Examples:

i. Approximate the real root to two four decimal places of

$$x^3 + 5x - 3 = 0$$

ii. Approximate the only solution to the equation

x=cos(x)

iii. Find the real root of the equation

3x=cos(x)+1.

Lab 3 –

Aim: Solution of simultaneous non-linear equations using Newton Raphson Method (Two variables).

Brief Theory: **TheNewton-Raphson second order method for two variable's** function can be obtain as follows

The solution is obtained with an efficient linear solving method like the partial pivoting gauss algorithm.

The second order Taylor's development gives the exact approximation of the function.

Numerical Procedure:

1. Consider a system of two equations in two variables: f(x, y) = 0 g(x, y) = 0.
2. Suppose we have an approximation for a solution (x0, y0) and we would like to compute Δx and Δy so x1 = x0 + Δx and y1 = y0 + Δy satisfy the system: f(x1, y1) = f(x0 + Δx, y0 + Δy) = 0
3. g(x1, y1) = g(x0 + Δx, y0 + Δy) = 0. compute Δx and Δy.
4. Taylor series in two variables f(x0 + Δx, y0 + Δy) = f(x0, y0) + ∂f ∂x (x0, y0)Δx + ∂f ∂y (x0, y0)Δy + · · ·
5. g(x0 + Δx, y0 + Δy) = g(x0, y0) + ∂g ∂x (x0, y0)Δx + ∂g ∂y (x0, y0)Δy + · · ·
6. where ∂f ∂x (x0, y0) and ∂f ∂y (x0, y0) are the partial derivatives of f with respect to x and y evaluated at (x0, y0);
7. ∂g ∂x (x0, y0) and ∂g ∂y (x0, y0) are the partial derivatives of g with respect to x and y evaluated at (x0, y0);
8. and the · · · represent the higher order terms in the series, in (Δx) 2 , (Δx)(Δy), and (Δy) 2 . Because Δx and Δy are already small numbers, the higher order terms are even smaller.
9. Then solve system of equation by using matrix form of the system of the equations.

Examples:

i. Solve the system of non linear equation:

$$x^2 + y = 11, y^2 + x = 7$$

ii. Solve the equation: $x = 2(y+1), y^2 = 3xy - 7$ correct to 3 decimals.

iii. Find a root of the equations $xy = x + 9, y^2 = x^2 + 7$

Lab 4 –

Aim: Numerical solution of central difference interpolation by using Lagrange interpolation method

Brief Theory: **Interpolation** is a **method** of finding new data points within the range of a discrete set of known data points. In other words **interpolation** is the **technique** to estimate the value of a mathematical**function**, for any intermediate value of the independent variable.

Numerical Procedure:

1. . Read n, X

2. Read xi, yi, where i = 1,2, 3, ….., n
3. sum = 0
4. for i = 1 (1) n, do till (11)
5. prod = 1
6. for j = 1 (1) n, do till (9)
7. if j ≠ i,
8. prod = prod * (X-x[j])/(x[i]-x[j])
9. Else, Next j
10. sum = sum + (prod *y[i])
11. Next i
12. Print X, sum
13. stop

Examples:

i. Given the following data table, construct the Lagrange interpolation

polynomial f(x), to fit the data and find f(1.25)

i	0	1	2	3
x_i	0	1	2	3
$y_i = f(x_i)$	1	2.25	3.75	4.25

ii. Given the following data table, construct the Lagrange interpolation polynomial f(x), to fit the data and find f(1998)

i	0	1	2	3	4	5
x_i	1980	1985	1990	1995	2000	2005
$y_i = f(x_i)$	440	510	525	571	500	600

iii. Find the value of y when x=10, if the following values of x and y are given:

x_i	5	6	9	11
$y_i = f(x_i)$	12	13	14	16

Lab 5 –

Aim: Numerical solution of integrals using Trapezoidal method

Brief Theory:

Numerical Procedure:

1. Start
2. Define and Declare function
3. Input initial boundary value, final boundary value and length of interval
4. Calculate number of strips, n = (final boundary value –final boundary value)/length of interval
5. Perform following operation in loop
    x[i]=x0+i*h
    y[i]=f(x[i])
    print y[i]
    Initialize se=0, s0=0
6. Do the following using loop
    If i %2 = 0
    So=s0+y[i]
    Otherwise
    Se=se+y[i]
    ans= h/3*(y[0]+y[n]+4*so+2*se)
7. print the ans
8. stop

Examples: i. Use Trapezoidal rule with 6 subintervals to approximate $\int_0^2 dx/(16+x^2)$.

ii. Use multiple segment Trapezoidal rule to find the area under the curve $f(x)=300x/(1+e^x)$ from x=0 to x=10.

iii. Use Trapezoidal rule with n=4, approximate the value of the integral $\int_0^4 \sqrt{x}dx$.

Lab 6 –

Aim: Numerical evaluation of integrals using Simpson's method

Brief Theory:C program for Simpson 1/3 rule for easy and accurate calculation of numerical integration of any function which is defined in program. In the source code , a function *f(x)* has been defined. The calculation using **Simpson 1/3 rule in C** is based on the fact that the small portion between any two points is a parabola. The program follows the following steps for calculation of the integral.

Numerical Procedure:

1. Define f(x)

2. Enter the values of lower and upper limit of x, i.e. x0 and also enter number of intervals, N(N should be even number)
3. h = ((xn – x0)/N)
4. sum = 0
5. do
6. {
7. sum = sum + (h/3).[f(x0) +4f(x0 +h) +f(x0 + 2h)]
8. x0 = x0 + 2h
9. } while (x0<xn)
10. print sum
11. stop

Examples:

i. Approximate $\int_{2}^{3} dx/(x+1)$ using Simpson's Rule with n=4.

ii. Compute the integral $\int_{0}^{1} e^{x^2} dx$ by Simpson's rule.

iii. Calculate the value of $\int_{0}^{p/2} \sin x dx$ using 11 ordinates.

Lab 7 –

Aim: Numerical solution of characteristic value problem by Power method.

Brief Theory: The **Power method** is an iterative **technique** used to determine the dominant eigenvalue of a matrix—that is, the eigenvalue with the largest magnitude. It also produces Eigen vector for corresponding Eigen value.

Numerical Procedure:

1. Start
2. Define matrix X
3. Calculate Y = AX
4. Find the largest element in magnitude of matrix Y and assign it to K.
5. Calculate fresh value X = (1/K) * Y
6. If [Kn – K(n-1)] > delta, goto step 3.
7. Stop

Examples:

i.Use power method to approximate a dominant eigenvalue and the corresponding eigenvector of $\begin{pmatrix} 4 & -5 \\ 2 & 3 \end{pmatrix}$ correct to 3-significant figures, after 10 iterations.

ii. Using Power Method, find the dominant eigenvalue and the corresponding eigenvector of the following matrix $\begin{pmatrix} 1 & -5 \\ -3 & -1 \end{pmatrix}$.

iii. Determine the largest eigen value and the corresponding eigen vector of the matrix

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

Lab 8 –

<u>Aim</u>: Numerical Solution of initial value problem by using Euler's method

<u>Brief Theory</u>: The **Euler method** is a first-order **method**, which means that the local error (error per step) is proportional to the square of the step size, and the global error (error at a given time) is proportional to the step size.

<u>Numerical Procedure</u>:

1. Start
2. Define function
3. Get the values of x0, y0, h and xn
   *Here x0 and y0 are the initial conditions
   h is the interval
   xn is the required value
4. n = (xn – x0)/h + 1
5. Start loop from i=1 to n
6. y = y0 + h*f(x0,y0)
   x = x + h
7. Print values of y0 and x0
8. Check if x <xn
   If yes, assign x0 = x and y0 = y
   If no, goto 9.
9. End loop i
10. Stop

<u>Examples</u>: Solve the Initial Value Problems

i. dy/dx=6−2(y/x), y(3)=1

ii. dy/dx=yln y/x, y(2)=e

iii. dy/dx=(y-x)/(y+x) , y(0)=1, Find y for x=0.1.

Lab 9 –

Aim: Numerical Solution of initial value problem usingRungaKutta method

Brief Theory:C program for RungeKutta 4 method is designed to find out the numerical solution of a first order differential equation. It is a kind of initial value problem in which initial conditions are known, i.e the values of x_0 and y_0 are known, and the values of y at different values x is to be found out.

Numerical Procedure:

1. Define f(x,y)
2. Enter the value of xo,y0, xn, h
3. do
   {
   k1= h.f(x0, y0)
   k2 = h.f(x0+h/2,y0+k1/2)
   k3 = h.f(x0 + h/2, y0 + k2/2)
   k4 = h.f(x0 + h, y0 + k3)
   k = (k1 + 2.k2 + 2.k3 + k4)/6
   y1 = y0 + k
   print x0, y0
   y0 = y1
   x0 =x0 +h
   }while(x0<xn)
4. stop

Examples:

i. Using RK method of order four find y at x=1.1 and 1.2 by solving dy/dx=x^2+y^2, y(1)=2.3

ii. Using RK method of order four find y at x=0.1 for dy/dx=x-y^2, y(0)=1.

iii. Find an approximate value of y for x=0.8,given that y=0.41 when x=0.4 anddy/dx= $\sqrt{(x+y)}$.

Lab 10 –

Aim: Numerical Solution for heat equation by suitable method

Brief Theory: The working principle of solution of heat equation in C is based on a rectangular mesh in a x-t plane (i.e. space-time plane) with the spacing h along x direction and k along t direction or. Simply, a mesh point (x,t) is denoted as (ih,jk). The calculations are based on one dimensional heat equation which is given as:

$du/dt = c^2(d^2u/dx^2)$

where $c^2 = k/s\rho$ is the diffusivity of a substance,
k= coefficient of conductivity of material,
ρ=density of the material, and
s= specific heat capacity

Numerical Procedure:

1. As the program is executed first of all it asks for value of square of c, value of u(0, t) and u(8, t) .
2. Using the input value the C program calculates the value of alpha.
3. Then, the iteration stats to calculate the value of u at different space and time i.e. u(x,t) using following iteration formula:

   $U_{i,j+1} = ½(u_{i-1, j} + u_{i+1,j})$

4. Finally the the program gives the result as console output as rectangular mess.

Examples:

i. Find a solution to the following partial differential equation that will also satisfy the boundary conditions. ∂u/∂t=k ∂2u/∂x2

u(x,0)=f(x), u(0,t)=0, u(L,t)=0

ii. ∂u/∂t=k ∂2u/∂x2

u(x,0)=f(x), ∂u/∂x(0,t)=0, ∂u/∂x(L,t)=0

iii. Solve the equation ∂u/∂t=k ∂2u/∂x2 subject to the condition u(x,0)=sin$p$ x, $0 £ x £ 1$, u(0,t)=u(1,t)=0 using Crank Nicolson method. Carry out computations for two levels taking h=1/3, k=1/36

Lab 11 –

Aim: Numerical Solution of wave equation by suitable method

Brief Theory: The Wave Equation is the simplest example of hyperbolic differential equation which is defined by following equation:

$d^2u/dt^2 = c^2 * d^2u/dt^2$

The C program for solution of wave equation presented here uses the following boundary conditions to solve the problems:

u(x,0)=f(x)

u_1(x,0)=φ(x)

u(0,t)=ψ_1(t)

u(1,t) = ψ_2(t)

for 0≤t≤T

<u>Numerical Procedure</u>:

1. First of all, the program asks for the value of square of c.
2. Then, the user has to input value of initial and boundary conditions i.e, u(0,t) and u(5,t). In this program, u(0,t) and u(5,t) are initial and boundary conditions respectively. They can be changed depending upon the nature of problem and given criteria for solution.
3. After inputting these parameters, the program displays the output on screen.

<u>Examples</u>:

i.Solve ∂2*u*/∂ *t*2=16 ∂2*u*/∂ *x*2, 0<x<2 *p*

u(0,t) =0=u(2 *p* ,t)

u(x,0) =2-x/2 *p* , ∂u(x,0)/ ∂t=1.

ii. $u_{tt} = 4u_{xx}$ for x I [0, L] and t I [0,T]
with boundary conditions
u(0, t) = 0, u(L, t) = 0.

iii. Evaluate the pivotal values of the equation $u_{tt} - 16u_{xx}$, taking $Dx \text{º } 1$ upto t=1.25. The boundary conditions are u(0,t)=u(5,t)=0, u_t (x,0)=0, u(x,0)= $x^2(5- x)$.

Lab 12 –

<u>Aim</u>: Numerical Solution of Laplace equation in two variables by suitable method

<u>Brief Theory</u>:$d^2u/dx^2 + d^2u/dy^2 = 0$
The program below for **Solution of Laplace equation in C** language is based on the finite difference approximations to derivatives in which the xy-plane is divided into a network of rectangular of sides Δx=h and Δy=k by drawing a set of lines.
x=ih,i=0,1,2,…
y=jk, j=0, 1, 2….
The points of intersection of these families of lines are called mesh points, lattice points or grid points.
<u>Numerical Procedure</u>:

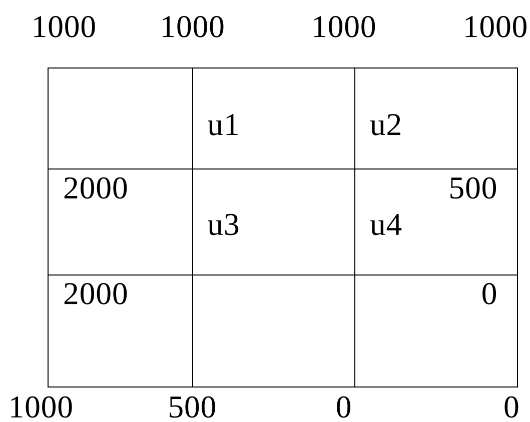
1. First of all the program asks for the boundary conditions. The user has to input the value ‘u’ at various mesh points.
2. After inputting the boundary condition, the program asks for the value of maximum allowed error and the maximum number iteration. The number of iteration depends on the degree of accuracy.

3. The initial values of iteration are already fixed in the source code. The program executes and there are two conditions to control the number of iteration; maximum allowed error and number of maximum number of iteration. If the desired accuracy is not met with the entered number of iteration the program gives the error message.

Examples:

i. Solve the Laplace's equation,
$\partial 2u/\partial x2 + \partial 2u/\partial y2 = 0$, for $u(x,y)$ defined withinthe domain of $0 \le x \le 1$ and $0 \le y \le 1$, given the boundary conditions
(I) $u(x, 0) = 1$ (II) $u(x,1) = 2$ (III) $u(0,y) = 1$ (IV) $u(1,y) = 2$.

ii. Given the values of u(x,y) on the boundary of the square in the fig, evaluate u(x,y) satisfying the Laplace equation $\nabla^2 u = 0$ at the pivotal points:



iii. Solve the Laplace equation $u_{xx} + u_{yy} = 0$ in the domain of the following figure by Jacobi's method.