NUMERICAL ANALYSIS

MATLAB Practicals (Autumn 2020)

B.E. III Semester

Thapar Institute of Engineering & Technology

Patiala

Name:................

Roll No:................

Group:..................

Instructor:................

Contents

Experiment 1 : Bisection Method

1. **Algorithm of Intermediate Value Theorem:** To determine all the subintervals $[a, b]$ of $[-N, N]$ that containing the roots of $f(x) = 0$.
   Input: function $f(x)$, and the values of h, N
   for $i = -N : h : N$
   if $f(i) * f(i + h) < 0$ then $a = i$ and $b = i + h$
   end if end i

2. **Algorithm of Bisection method:** To determine a root of $f(x) = 0$ that is accurate within a specified tolerance value ϵ, given values a and b such that $f(a)\ f(b) < 0$.
   Define $c = \dfrac{a + b}{2}$.
   If $f(a)\ f(c) < 0$, then set $b = c$, otherwise $a = c$.
   End if.
   Until $|a - b| \leq \epsilon$ (tolerance value).
   Print root as c.
   Stopping Criteria: Since this is an iterative method, we must determine some stopping criteria that will allow the iteration to stop. Criterion $|f(c_k)|$ very small can be misleading since it is possible to have $|f(c_k)|$ very small, even if c_k is not close to the root.
   The interval length after N iterations is $\dfrac{b - a}{2^N}$. So, to obtain an accuracy of ϵ, we must have
   $$N \geq \frac{\log(b - a) - \log \epsilon}{\log 2}.$$

3. Students are required to write both the programs (IVT and Bisection) and implement it on the following examples.
   (i) Use bisection method in computing of $\sqrt{29}$ with $\epsilon = 0.001$, $N = 10, h = 1$
   (ii) The smallest positive root of $\cos x = 1/2 + \sin x$ with $\epsilon = 0.001$ in $[0.001, 1], h = 0.0001$
   (iii) Determine the number of iterations necessary to solve $f(x) = x^3 + 4x^2 - 10 = 0$ with accuracy 10^{-3} using $a = 1$ and $b = 2$ and hence find the root with desired accuracy.
   sol.

Experiment 2 : Newton's and Secant Methods

---

1. Write an algorithm for Newton's and Secant method to compute the roots of the given non-linear equations.
   Algorithm for Newton's method: To find a solution to $f(x) = 0$, given an initial approximation x_0.
   Input: Initial approximation x_0, tolerance value ϵ, maximum number of iterations N.
   Output: Approximate solution x_1 or message of failure.
   Step 1: Set $i = 1$.
   Step 2: While $i \leq N$ do Steps 3 to 6.
   Step 3: Set $x_1 = x_0 - \dfrac{f(x_0)}{df(x_0)}$. (Compute x_i.)
   Step 4: If $|x_1 - x_0| \leq \epsilon$ or $\dfrac{|x_1 - x_0|}{|x_1|} < \epsilon$ then OUTPUT x_1; (The procedure was successful.) STOP.
   Step 5: Set $i = i + 1$.
   Step 6: Set x_0 =x_1. (Update x_0.)
   Step 7: Output ('The method failed after Niterations, N=', N); (The procedure was unsuccessful.) STOP.
   Algorithm for secant method:
   1. Give inputs and take two initial guesses x_0 and x_1.
   2. Start iterations
   $$x_2 = x_1 - \frac{x_1 - x_0}{f_1 - f_0} f_1.$$
   3. If
   $$|x_2 - x_1| < \epsilon \text{ or } \frac{|x_2 - x_1|}{|x_2|} < \epsilon$$
   then stop and print the root.
   4. Repeat the iterations (step 2). Also check if the number of iterations has exceeded the maximum number of iterations.

2. Students are required to write the program and compare the results by both the methods for the following examples. $\epsilon = 0.00001$.
   (i) Compute $\sqrt{17}$.
   (ii) The smallest positive root of $f(x) = \cos x = 1/2 + \sin x$.
   (iii) The root of $\exp(-x)(x^2 + 5x + 2) + 1 = 0$. Take initial guess -2.0 and -1.0.
   (iii) Find a non-zero solution of $x = 2 \sin x$. (apply IVT to find an initial guess.)

3. Solve the equation $4x^2 - e^x - e^{-x} = 0$ which has two positive solutions x_1 and x_2. Use Newton's method to approximate the solution to within 10^{-5} with the following values of x_0:
   $$x_0 = -10, -5, -3, 0, 1, 3, 5, 10.$$
   Sol.

Experiment 3 : Fixed-point iteration method

---

1. **Algorithm of fixed-point iterations:** To find a solution to $x = g(x)$ given an initial approximation x_0.
   Input: Initial approximation x_0, tolerance value ϵ, maximum number of iterations N.
   Output: Approximate solution α or message of failure.
   Step 1: Set $i = 1$.
   Step 2: While $i \leq N$ do Steps 3 to 6.
   Step 3: Set $x_1 = g(x_0)$. (Compute x_i.)
   Step 4: If $|x_1 - x_0| \leq \epsilon$ or $\dfrac{|x_1 - x_0|}{|x_1|} < \epsilon$ then OUTPUT x_1; (The procedure was successful.) STOP.
   Step 5: Set $i = i + 1$.
   Step 6: Set $x_0 = x_1$. (Update x_0.)
   Step 7: Print the output and STOP.

2. The equation $f(x) = x^3 + 4x^2 - 10 = 0$ has a unique root in $[1, 2]$. There are many ways to change the equation to the fixed-point form $x = g(x)$ using simple algebraic manipulation. Let g_1, g_2, g_3, g_4, g_5 are iteration functions obtained by the given function, then check which of the following iteration functions will converge to the fixed point? (tolerance $\epsilon = 10^{-3}$)
   (a) $g_1(x) = x - x^3 - 4x^2 + 10$
   (b) $g_2(x) = \sqrt{\frac{10}{x} - 4x}$
   (c) $g_3(x) = 0.5\sqrt{10 - x^3}$
   (d) $g_4(x) = \sqrt{\dfrac{10}{4 + x}}$
   (e) $g_5(x) = x - \dfrac{x^3 + 4x^2 - 10}{3x^2 + 8x}$.

3. Find smallest and second smallest positive roots of the equation $\tan x = 4x$, with an accuracy of 10^{-3} using fixed-point iterations.

4. Use a fixed-point iteration method to determine a solution accurate to within 10^{-2} for $2 \sin \pi x + x = 0$ on $[1, 2]$. Use initial guess $x_0 = 1$.

   Sol.

Experiment 4 : LU factorization and Gauss Elimination

---

1. (a) Write an algorithm for LU Factorization using Gauss elimination method.
   Algorithm for LU Factorization: Input: number of unknowns n; coefficient matrix $A = [a_{ij}], l_{ii} = 1$, where $1 \leq i \leq n$ and $1 \leq j \leq n, .$
   Output: the entries $l_{ij}, 1 \leq j \leq i, 1 \leq i \leq n$ of L and the entries, $u_{ij}, i \leq j \leq n$, $1 \leq i \leq n$ of U.
   Step 1: For $i = 1, \cdots, n-1$ do Steps 2-4. (Elimination process.)
   Step 2: Let p be the smallest integer with $i \leq p \leq n$ and $a_{pi} \neq 0$.
   If no integer p can be found
   then OUTPUT ('no solution');
   STOP.
   Step 3: If $p \neq i$ then perform $(E_p) \leftrightarrow (E_i)$.
   Step 4: For $j = i+1, \cdots, n$ do Steps 5 and 6.
   Step 5: Set $m_{ji} = a_{ji}/a_{ii}$, $l_{ji} = m_{ji}$.
   Step 6: Perform $(E_j) \to (E_j - m_{ji}E_i)$;
   step 7: $U = A$

2. Apply Gauss elimination method to augmented matrix $[A : b]$ and hence apply back substitution on the obtained system.
   Algorithm for back substitution:

   Step 1 If $a_{nn} = 0$ then OUTPUT ('no unique solution exists');
   STOP.
   Step 2: Set $x_n = a_{n,n+1}/a_{nn}$.(Start backward substitution.)
   Step 3: For $i = n-1, \cdots, 1$ set $x_i = [a_{i,n+1} - \sum\limits_{j=i+1}^{n} a_{ij}x_j]/a_{ii}$.
   Step 4: OUTPUT $(x_1, \cdots, x_n)$; (Procedure completed successfully.)
   STOP.

3. Factorize coefficient matrix of the following system of equations using the LU Factorization Algorithm and hence apply back substitution to find the solution the system:
(a)

$$\begin{aligned} 10x + 8y - 3z + u &= 16 \\ 2x + 10y + z - 4u &= 9 \\ 3x - 4y + 10z + u &= 10 \\ 2x + 2y - 3z + 10u &= 11 \end{aligned}$$

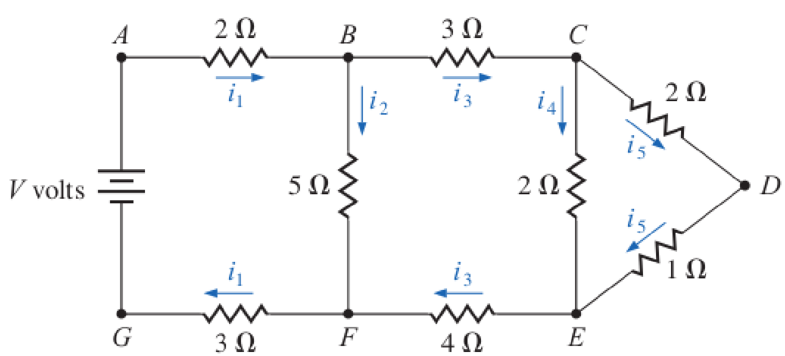
(b)

$$\begin{aligned} \pi x_1 + \sqrt{2}x_2 - x_3 + x_4 &= 0 \\ e x_1 - x_2 + x_3 + 2x_4 &= 1 \\ x_1 + x_2 - \sqrt{3}x_3 + x_4 &= 2 \\ -x_1 - x_2 + x_3 - \sqrt{5}x_4 &= 3. \end{aligned}$$

4. Kirchhoff's laws of electrical circuits state that both the net flow of current through each junction and the net voltage drop around each closed loop of a circuit are zero. Suppose that a potential of V volts is applied between the points A and G in the circuit and that i_1, i_2, i_3, i_4, and i_5 represent current flow as shown in the diagram. Using G as a reference point, Kirchhoff's laws imply that the currents satisfy the following system of linear equations:

$$\begin{aligned} 5i_1 + 5i_2 &= V, \\ i_3 - i_4 - i_5 &= 0, \\ 2i_4 - 3i_5 &= 0, \\ i_1 - i_2 - i_3 &= 0, \\ 5i_2 - 7i_3 - 2i_4 &= 0. \end{aligned}$$

By taking $V = 5.5$, solve the system.



Sol.

Experiment 5 : Gauss-Seidel and Successive-Over-Relaxation

1. **Algorithm:** Given a system of linear equations:

$$Ax = b.$$

In order to obtain the solution of the above system of equations by Gauss-Seidel method, we use the following algorithm:

1 Input matrix $A = [a_{ij}], b, x0$, tolerance TOL, maximum number of iterations

2 Set $k = 1$

3 while $(k \leq N)$ do step 4-7

4 For $i = 1, 2, \cdots, n$

$$x_i = \frac{1}{a_{ii}} \left[-\sum_{j=1}^{i-1} (a_{ij} x_j) - \sum_{j=i+1}^{n} (a_{ij} x0_j) + b_i \right]$$

5 If $||x - x0|| < TOL$, then OUTPUT $(x_1, x_2, \cdots, x_n)$
STOP

6 $k = k + 1$

7 For $i = 1, 2, \cdots, n$
Set $x0_i = x_i$

8 OUTPUT $(x_1, x_2, \cdots, x_n)$
STOP.

2. Write an algorithm for Successive-Over-Relaxation (SOR) method.

3. Solve this system of equations by Gauss-Seidel starting with the initial vector $[0, 0, 0]$ and tolerance 10^{-3} :

$$\begin{aligned} 4.63x_1 - 1.21x_2 + 3.22x_3 &= 2.22 \\ -3.07x_1 + 5.48x_2 + 2.11x_3 &= -3.17 \\ 1.26x_1 + 3.11x_2 + 4.57x_3 &= 5.11. \end{aligned}$$

   Sol.

4. Use the SOR method with $\omega = 1.2$ to solve the linear system with an initial vector $[0, 0, 0, 0]$ a tolerance 10^{-3} in the $\|.\|_\infty$ norm.

$$\begin{aligned} 4x_1 + x_2 - x_3 + x_4 &= -2 \\ x_1 + 4x_2 - x_3 - x_4 &= -1 \\ -x_1 - x_2 + 5x_3 + x_4 &= 0 \\ x_1 - x_2 + x_3 + 3x_4 &= 1. \end{aligned}$$

   Sol.

Experiment 6 : Power Method and Lagrange interpolation

1. **Algorithm for Power method:**
   (a) Start
   (b) Define matrix A and initial guess x
   (c) Calculate $y = Ax$
   (d) Find the largest element in magnitude of matrix y and assign it to K.
   (e) Calculate fresh value $x = (1/K) * y$
   (f) If $|K(n) - K(n-1)| >$ error, goto step c.
   (g) Stop

2. (a) Determine the largest eigenvalue and the corresponding eigenvector of the matrix

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 20 & 1 \\ 0 & 1 & 4 \end{bmatrix}$$

using the power method. Use $x^0 = [1, 1, 1]^T$ and $\epsilon = 10^{-3}$.
(b) Find the smallest eigenvalue and the corresponding eigenvector of the following matrix by inverse power method.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 3 & 3 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

Use $x^0 = [1, 1, 0, 1]^T$ and $\epsilon = 10^{-3}$.

Solution.

3. **Algorithm for Lagrange interpolation:** Given a set of function values:

x	x_1	x_2	$\cdots$	x_n
$f(x)$	$f(x_1)$	$f(x_2)$	$\cdots$	$f(x_n)$

To approximate the value of a function $f(x)$ at $x = p$ using Lagrange's interpolating polynomial $P_n(x)$ of degree n, given by

$$P_n(x) = l_1(x)f(x_1) + l_2(x)f(x_2) + \ldots + l_n(x)f(x_n)$$

where $l_i(x) = \prod_{j=1;i\neq j}^{n} \frac{(p - x_j)}{(x_i - x_j)}$.

We write the following algorithm by taking n points and thus we will obtain a polynomial of degree $\leq n - 1$:

- Input: The degree of the polynomial, the values $x(i)$ and $f(i)$, $i = 1, 2, \ldots, n$, and the point of interpolation p.
- Calculate the Lagrange's fundamental polynomials $l_i(x)$ using the following loop:
  `for i=1 to n`
  $l(i) = 1.0$
  `for j=1 to n`
  `if` $j \neq i$
  $l(i) = \dfrac{p - x(j)}{x(i) - x(j)} l(i)$
  `end j`
  `end i`
- Calculate the approximate value of the function at $x = p$ using the following loop:
  `sum=0.0`
  `for i=1 to n`
  `sum = sum +` $l(i) * f(i)$
  `end i`
- Print `sum`.

4. Use Lagrange's interpolation formula to approximate the value of $f(0.43)$, given that

x	0	0.25	0.5	0.75
$f(x)$	1	1.64872	2.71828	4.48169

Sol.

5. A census of the population of the United States is taken every 10 years. The following table lists the population, in thousands of people, from 1950 to 2000, and the data are also represented in the figure.

Year	1950	1960	1970	1980	1990	2000
Population(in thousands)	151326	179323	203302	226542	249633	281422

Use Lagrange interpolation to approximate the population in the years 1965, 1975, and 1995.

Sol.

Experiment 7 : Newton's divided difference Interpolation

1. **Algorithm** Write the algorithm for Newton's divided difference interpolation and then apply for the given examples on next page.

2. The following data represents the function $f(x) = e^x$.

x	1	1.5	2.0	2.5
$f(x)$	2.7183	4.4817	7.3891	12.1825

Estimate the value of $f(2.25)$ using the Newton's divided difference interpolation. Compare with the exact value.

Sol.

3. Approximate $f(0.43)$ by using Newton's divided difference interpolation, construct interpolating polynomials for the following data.

$$f(0) = 1, f(0.25) = 1.64872, f(0.5) = 2.71828, f(0.75) = 4.4816.$$

Sol.

Experiment 8 : Numerical Quadrature

1. **Algorithm (Composite trapezoidal rule):**
   Step 1 : Inputs: function $f(x)$; end points a and b; and N number of subintervals.
   Step 2 : Set $h = (b - a)/N$.
   Step 3 : Set sum = 0
   Step 4 : For $i = 1$ to $N - 1$
   Step 5 : Set $x = a + h * i$
   Step 6 : Set sum = sum+$2 * f(x)$
   end
   Step 7 : Set sum = sum+$f(a) + f(b)$
   Step 8 : Set ans = sum$*(h/2)$
   End

2. **Algorithm (Composite Simpson's rule):**
   Step 1 : Inputs: function $f(x)$; end points a and b; and N number of subintervals (even).
   Step 2 : Set $h = (b - a)/N$.
   Step 3 : Set sum = 0
   Step 4 : For $i = 1$ to $N - 1$
   Step 5 : Set $x = a + h * i$
   Step 6 : If rem$(i, 2) == 0$
   sum = sum+$2 * f(x)$
   else
   sum = sum+$4 * f(x)$
   end
   Step 7 : Set sum = sum+$f(a) + f(b)$
   Step 8 : Set ans = sum$*(h/3)$
   End

3. Approximate the following integrals using the composite trapezoidal and Simpson rule by taking different subintervals (e.g. 4, 6, 10, 20). a. $I = \int\limits_{-0.25}^{0.25} (\cos x)^2 \; dx$ b. $\int\limits_{e}^{e+1} \frac{1}{x \ln x} \; dx$. c. $\int_{-1}^{1} e^{-x^2} \cos x \; dx$.
   Sol.

4. The length of the curve represented by a function $y = f(x)$ on an interval $[a, b]$ is given by the integral

$$I = \int_a^b \sqrt{1 + [f'(x)]^2}\, dx.$$

Use the trapezoidal rule and Simpson's rule with 4 and 8 subintervals compute the length of the curve $y = \tan^{-1}(1 + x^2),\ 0 \leq x \leq 2$.
Sol.

Experiment 9: Solution of Initial Value Problem

1. **Algorithm (Modified Euler's method):**

2. **Algorithm (Runge-Kutta fourth-order method):**

3. Solve the following differential equations by the modified Euler's method and Runge-Kutta fourth-order method:
(a) $y' = -y + 2\cos t,\ y(0) = 1$.
(b) $y' = \sqrt{x+y}, y(0) = 0.8$ Compute solution in the interval $[0, 1]$ with mesh length 0.2.
Sol.