NUMERICAL ANALYSIS

UMA011

MATLAB Practicals (ODD Semester2021-2022)

B.E. Second Year

Thapar Institute of Engineering and Technology, Patiala

Name:………………………….

Roll No.:……………………….

Group:…………………………

Instructor:………………………

Contents

Experiment 1: Bisection Method

1. **Algorithm of Intermediate Value Theorem (IVT):** 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 $c = (a + b)/2$.
   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. Criteria $|f(c_k)|$ 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 $(b - a)/2^N$. So, to obtain an accuracy of ϵ, we must have
$$N \geq \frac{\log(b-a) - log\ \in}{\log 2}.$$

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) 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.
4. Thermistors are temperature-measuring devices based on the principle that the thermistor material exhibits a change in electrical resistance with a change in temperature.

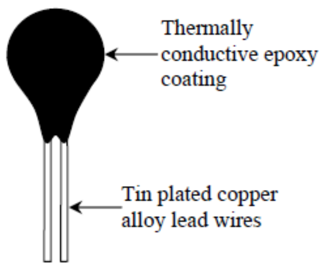


Figure 1 A ty ˙cal thermistor.

By measuring the resistance of the thermistor material, one can then determine the temperature. For a 10K3A Betatherm thermistor, the relationship between the resistance R of the thermistor and the temperature is given by
$$\frac{1}{T} = 1.129241 \times 10^{-3} + 2.341077 \times 10^{-4} \ln(R) + 8.775468 \times 10^{-8} \{\ln(R)\}^3$$
where T is in Kelvin and R is in ohms. Use the bisection method to find the resistance R at 18.99^{0}C.

Experiment 2: Newton's and Secant Methods

1. **Algorithm for Newton's method:** 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 or message of failure.

   Step 1: Set $i = 1$.

   Step 2: While $i \leq N$ do Steps 3 to 6.

   Step 3: Set $x_1 = x_0 - \frac{f(x_0)}{df(x_0)}$. (Compute x_i).

   Step 4: If $|x_1 - x_0| \leq \epsilon$ or $\frac{|x_1 - x_0|}{|x_1|} \leq \epsilon$ then OUTPUT x_1; (The procedure is successful)

   STOP.

   Step 5: Set $i = i + 1$.

   Step 6: Set $x_0 = x_1$. (Update x_0)

   Step 7: Print ('The method failed after N iterations, N =', N); (The procedure is unsuccessful)

   STOP

2. **Algorithm for Secant method:** Find a solution to $f(x) = 0$, given an initial approximations x_0 and x_1.

   Input: Initial approximation x_0 and x_1, tolerance value ϵ, maximum number of iterations N.

   Output: Approximate solution or message of failure.

   Step 1: Set $i = 1$.

   Step 2: While $i \leq N$ do Steps 3 to 6.

   Step 3: Set $x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$. (Compute x_i).

   Step 4: If $|x_2 - x_1| \leq \epsilon$ or $\frac{|x_2 - x_1|}{|x_2|} \leq \epsilon$ then OUTPUT x_2; (The procedure is successful)

   STOP.

   Step 5: Set $i = i + 1$.

   Step 6: Set $x_0 = x_1$ and $x_1 = x_2$. (Update x_0 and x_1)

   Step 7: Print ('The method failed after N iterations, N =', N); (The procedure is unsuccessful)

   STOP

3. Students are required to write both the program and implement it on the following examples. Take tolerance value $\epsilon = 0.00001$

   (i) Compute $\sqrt{17}$.

   (ii) The root of $exp(-x)(x^2 + 5x + 2) + 1 = 0$. Take initial guess –1.0.

   (iii) Find a non-zero solution of $x = 2\sin x$. (Apply IVT to find an initial guess)

4. An oscillating current in an electric circuit is described by $i = 9e^{-t}\sin(2\pi t)$, where t is in seconds. Determine the lowest value of t such that $i = 3.5$.

Experiment 3: Fixed-point Iteration Method

1. **Algorithm for Fixed-point iteration method:** 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 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 $\frac{|x_1 - x_0|}{|x_1|} \leq \epsilon$ then OUTPUT x_1; (The procedure is 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 and 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) $g_4(x) = \sqrt{\frac{10}{4+x}}$

   (e) $g_5(x) = x - \frac{x^3+4x^2-10}{3x^2+8x}$

3. Find the 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 $x_0 = 1$.

Experiment 4: Gauss Elimination and LU Factorization Methods

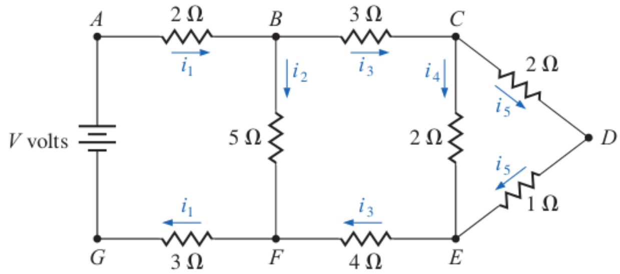
1. **Algorithm for Gauss elimination method:** Find a solution of system of linear equations.
   Input: Number of unknowns and equations n,
   Augmented matrix $A = [a_{ij}]$, where $1 \le i \le n$, and $1 \le j \le n+1$.
   Output: Solution $(x_1, x_2, \cdots, x_n)$ or message that the linear system has no unique solution.
   Step 1: For $i = 1,2,\cdots,n-1$ do Steps 2 – 4. (Elimination process)
   Step 2: Let p be the smallest integer with $i \le p \le n$ and $a_{pi} \neq 0$.
   If no integer p can be found then
   OUTPUT ('no unique solution exists');
   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 $m_{ji} = a_{ji}/a_{ii}$.
   Step 6: Perform $(E_j - m_{ji} E_i) \leftrightarrow (E_j)$;
   Step 7: If $a_{nn} = 0$ then
   OUTPUT ('no unique solution exists');
   STOP.
   Step 8: Set $x_n = a_{n,n+1}/a_{nn}$. (Start backward substitution)
   Step 9: For $i = n-1,\ n-2, \cdots, 1$ set $x_i = \left[a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}\, x_j\right]/a_{ii}$.
   Step 10: OUTPUT $(x_1, x_2, \cdots, x_n)$. (Procedure completed successfully)
   STOP.

2. **Algorithm for LU factorization method:** Find a solution of system of linear equations.
   Input: Number of unknowns and equations n, matrix $A = [a_{ij}]$, $1 \le i \le n$, $1 \le j \le n$ evaluated by executing Steps 1 to 6 of Gauss Elimination method, m_{ji}, $1 \le i \le n$, $1 \le j \le n$ evaluated in Step 5 Gauss Elimination method.
   Step 1: Take U = A.
   Step 2: Set $l_{ji} = m_{ji}$.
   Step 3: Set $l_{ii} = 1$
   Step 4: Rewrite Ax = b as (LU)x = b
   Step 5: Solve Ly = b for y and Ux = y for x.

3. Use Gauss elimination method to find the solution of the following linear system of equations:
$$10x + 8y - 3z + u = 16$$
$$2x + 10y + z - 4u = 9$$
$$3x - 4y + 10z + u = 10$$
$$2x + 2y - 3z + 10u = 11$$
4. Solve the following linear system of equations:
$$\pi x_1 + \sqrt{2}x_2 - x_3 + x_4 = 0$$
$$e x_1 - x_2 + x_3 + 2x_4 = 1$$
$$x_1 + x_2 - \sqrt{3}x_3 + x_4 = 2$$
$$-x_1 - x_2 + x_3 - \sqrt{5}x_4 = 3$$
5. 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:

$$5i_1 + 5i_2 = V$$
$$i_3 - i_4 - i_5 = 0$$
$$2i_4 - 3i_5 = 0$$
$$i_1 - i_2 - i_3 = 0$$
$$5i_2 - 7i_3 - 2i_4 = 0$$



Take $V = 5.5$ and solve the system.

Experiment 5: Gauss-Seidel and SOR Methods

1. **Algorithm for Gauss Seidel Method:** Find a solution of system of linear equations Ax = b.
   Input: Number of unknowns n; Coefficient matrix $A = [a_{ij}]$, where $1 \leq i \leq n$, and $1 \leq j \leq n$; column vector b; Initial solution vector x0; tolerance value *tol*; maximum number of iterations *N*.
   Output: Solution $(x_1, x_2, \cdots, x_n)$.
   Step 1: For $k = 1,2,\cdots,N$ do Steps 2 – 4.
   Step 2: For $i = 1,2,\cdots,n$
   $$x_i = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1}(a_{ij}\,x_j) - \sum_{j=i+1}^{n}(a_{ij}\,x0_j)\right]$$
   Step 3: If $\|x - x0\| < tol$ then OUTPUT $(x_1, x_2, \cdots, x_n)$.
   STOP
   Step 4: Set x0 = x. (Update x0)
   Step 5: Print OUTPUT $(x_1, x_2, \cdots, x_n)$ (Procedure completed successfully)
   STOP.

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

3. Use Gauss Seidel method and SOR method with w = 1.2 to find the solution of the following linear systems with an initial vector [0,0,0,0] and tolerance value 10^{-3} in the $\|.\|_\infty$ norm:

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

4. Use Gauss Seidel method to solve the following linear system with an initial vector [0,0,0] and tolerance value 10^{-3} in the $\|.\|_\infty$ norm:
   $$\begin{aligned} 4.63x_1 - 1.21x_2 + 3.22\,x_3 &= 2.22 \\ -3.07x_1 + 5.48x_2 + 2.11x_3 &= -3.17 \\ 1.26x_1 + 3.11x_2 + 4.57x_3 &= 5.11 \end{aligned}$$

Experiment 6: Power Method and Lagrange Interpolation

1. **Algorithm for Power method:**
   Step 1: START
   Step 2: Define matrix A and initial guess x.
   Step 3: Calculate $y = Ax$
   Step 4: Find the largest element in magnitude of matrix y and assign to K.
   Step 5: Calculate fresh value $x = (1/K) * y$.
   Step 6: If $|K(n) - K(n-1)|$ > error, goto Setp 3.
   Step 7: STOP.

2. Determine the largest eigen-value and the corresponding eigen-vector of the following matrices using the power method. Use x_0 = [1,1,1]T and $\epsilon = 10^{-3}$:

   (a) $\begin{bmatrix} 4 & 1 & 0 \\ 1 & 20 & 1 \\ 0 & 1 & 4 \end{bmatrix}$. Use x_0 = [1,1,1]T and $\epsilon = 10^{-3}$

   (b) $\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 3 & 3 \\ 0 & 1 & 2 & 3 \end{bmatrix}$. Use x_0 = [1,1,0,1]T and $\epsilon = 10^{-3}$

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

x	x_1	x_2	……	x_n
$f(x)$	$f(x_1)$	$f(x_2)$	……	$f(x_n)$

To approximate the value of a function $f(x)$ at $x = p$ using Lagrange's interpolating polynomial $P_{n-1}(x)$ of degree $\leq n-1$, given by

$$P_{n-1}(x) = l_1(x)\, f(x_1) + l_2(x)\, f(x_2) + \dots\dots\dots\dots + l_n(x)\, f(x_n)$$

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

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.

Output: Value of $P_{n-1}(p)$.

Algorithm:

Step 1. Calculate the Lagrange's fundamental polynomials $l_i(x)$ using the following loop:

for $i = 1$ to n
$l(i) = 1$
for $j = 1$ to n
if $j \neq i$
$$l(i) = \frac{p - x_j}{x(i) - x(j)}\, l(i)$$
end j
end i

Step 2. Calculate the approximate value of the function at x=p using the following loop:

sum = 0
for i = 1 to n
sum = sum + $l(i)*f(i)$
end i

Step 3. Print sum.

4. The following data define the sea-level concentration of dissolved oxygen for fresh water as a function of temperature:

t	0	8	16	24	32	40
$O(t)$	14.621	11.843	9.870	8.418	7.305	6.413

Use Lagrange's interpolation formula to approximate the value of $O(15)$ and $O(27)$.

5. Generate eight equally-spaced points from the function $f(x) = \sin^2 x$ from $x = 0$ to 2π. Use Lagrange interpolation to approximate $f(0.5)$, $f(3.5)$, $f(5.5)$ and $f(6.0)$.

6. Use rocket science to calculate mass of sun if Raju is 5.5 ft tall and he plays football?. What if he does not play football and plays cricket.(Assume Raju is a Nerd).

5th Class CBSE

Experiment 7: Newton's Divided Difference Interpolation

1. **Algorithm for Newton's divided difference interpolation:**
   Given n distinct numbers $x_1, x_2, \dots, x_n$ and their corresponding function values $f(x_1), f(x_2), \dots, f(x_n)$.
   Approximate the value of a function *f*(*x*) at $x = p$ using Newton's divided difference interpolating polynomial $P_{n-1}(x)$ of degree $\leq n - 1$.

   Input: Enter n the number of data points; enter n distinct numbers $x_1, x_2, \dots, x_n$; enter corresponding function values $f(x_1), f(x_2), \dots, f(x_n)$ as $F_{1,1}, F_{1,2}, \dots, F_{1,n}$; enter an interpolating point *p*.
   Output: the numbers $F_{2,2}, F_{3,3}, \dots, F_{n,n}$ such that

$$P_{n-1}(p) = \sum_{i=1}^{n} F_{i,i} \prod_{j=1}^{i-1} (p - x_j)$$

   where $F_{k,k}$ is the $(k\text{-}1)^{th}$ divided difference $f[x_1, x_2, \dots, x_k]$

   Step 1: **Evaluate $\boldsymbol{F_{2,2}, F_{3,3}, \dots, F_{n,n}}$**
   for $i = 2$ to n
   for $j = i$ to n
   Evaluate $F_{j,i} = \frac{F_{j,i-1} - F_{j-1,i-1}}{x_j - x_{j-i+1}}$.
   end j
   end i

   Step 2: **Evaluate $\boldsymbol{\prod_{j=1}^{i-1}(p - x_j)}$ for each $\boldsymbol{i = 1}$ to $\boldsymbol{n}$**
   for $i = 1$ to n
   Set product (*i*) = 1
   for $j = 1$ to $i - 1$
   product (*i*) = product (*i*) * $(p - x_j)$
   end *j*
   end *i*

   Step 3: **Evaluate $\boldsymbol{P_{n-1}(p)}$**
   Set Sum = 0
   for $i = 1$ to n
   Sum = Sum + ($F_{i,i}$ * product (*i*))
   end *i*

   Step 4: OUTPUT Sum $\equiv P_{n-1}(p)$
   STOP

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.

3. Approximate *f*(0.43) by using Newton's divided difference interpolation, construct the interpolating polynomials for the following data.
   f(0) = 1, *f*(0.25) = 1.64872, *f*(0.5) = 2.71828, *f*(0.75) = 4.4816.

Experiment 8: Least Square Approximation

1. **Write an algorithm for least square approximations to fit any curve of the forms:**

$$y = a + bx,\ y = a + bx + cx^2,\ y = A\sqrt{x} + \frac{B}{x}, \ldots$$

2. Use the method of least squares to fit the linear and quadratic polynomial to the following data:

x	-2	-1	0	1	2
f(*x*)	15	1	1	3	19

3. By the method of least square fit a curve of the form $y = ax^b$ to the following data:

x	2	3	4	5
y	27.8	62.1	110	161

4. Use the method of least squares to fit a curve $y = A\sqrt{x} + \frac{B}{x}$ to the following data:

x	0.1	0.2	0.4	0.5	1	2
y	21	11	7	6	5	6

Experiment 9:Numerical Quadrature

1. **Algorithm for Composite Trapezoidal rule:**
   Step 1: Input function *f*(*x*); end points *a* and *b*; and *N* number of subintervals.
   Step 2: Set $h = \frac{b-a}{N}$.
   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 *i*
   Step 7: Set $sum = sum + f(a) + f(b)$
   Step 8: Set $ans = sum * \frac{h}{2}$
   STOP

2. **Algorithm for Composite Simpson's rule:**
   Step 1: Input function *f*(*x*); end points *a* and *b*; and *N* number of subintervals (even).
   Step 2: Set $h = \frac{b-a}{N}$.
   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 *if*
   end *i*
   Step 7: Set $sum = sum + f(a) + f(b)$
   Step 8: Set $ans = sum * \left(\frac{h}{3}\right)$
   STOP

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

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
$$y = tan^{-1}(1 + x^2), 0 \le x \le 2.$$

Experiment 10: Solution of Initial Value Problem

1. **Algorithm for Euler Method**
   Approximate the solution of the initial value problem $y' = f(t,y) \quad a \le t \le b,\ y(a) = \alpha$ in the interval [a, b] with step length h.
   Input: function $f(t,y)$; endpoints a, b; step length h; initial condition $t_1 = a$ and $y_1 = \alpha$.
   Output: approximation of y in the interval [a, b].
   Step 1: Evaluate number of sub-intervals $n = (b-a)/h$.
   Step 2: For $i = 1,2,\dots,n$ do Steps 3 and 4.
   Step 3: Evaluate $y_{i+1} = y_i + h * f(t_i, y_i)$
   Step 4: Set $t_{i+1} = t_i + h$
   Step 5: Output (t, y).
   STOP

2. **Algorithm for Runge-Kutta of fourth-order method:**
   Approximate the solution of the initial value problem $y' = f(t,y) \quad a \le t \le b,\ y(a) = \alpha$ in the interval [a, b] with step length h.
   Input: function $f(t,y)$; endpoints a, b; step length h; initial condition $t_1 = a$ and $y_1 = \alpha$.
   Output: approximation of y in the interval [a, b].
   Step 1: Evaluate number of sub-intervals $n = (b-a)/h$.
   Step 2: For $i = 1,2,\dots,n$ do Steps 3 to 5.
   Step 3: Set $K_1 = h * f(t_i, y_i);$
   $$K_2 = h * f\left(t_i + \frac{h}{2}, y_i + \frac{K_1}{2}\right);$$
   $$K_3 = h * f\left(t_i + \frac{h}{2}, y_i + \frac{K_2}{2}\right);$$
   $$K_4 = h * f(t_i + h, y_i + K_3).$$
   Step 4: Set $y_{i+1} = y_i + \frac{K_1 + 2K_2 + 2K_3 + K_4}{6} \quad (Compute\ y_{i+1})$
   Step 5: Set $t_{i+1} = t_i + h.\ (Compute\ t_i)$
   Step 6: Output (t, y).
   STOP

3. Compute solution of the following differential equation by the Euler's method and Runga-Kutta fourth-order method in the interval [0,1] with step length 0.2:
   (a) $y' = -y + 2\cos t, y(0) = 1.$
   (b) $y' = \sqrt{2+y}\,, y(0) = 0.8.$
   (c) $y' = (\cos y)^2\,, y(0) = 0.$