Practical – 1

Write a program in Scilab to find the roots of the equation

x^3 - 9x - 1=0 using Bisection method correct upto 4 decimal places.

Code :

```
//Find the root of the equation x^3 -9x -1 using bisection method
clc
clear

// Define funtion or equation
function y=f(x)
   y = x^3-9*x-1
endfunction

// Find initial values
for i = 0:5
   if f(i) < 0 & f(i+1)>0 then
      a = i
      b = i+1
   end
end
printf("\n Initial Values are:")
printf("\n a = %d,\n b = %d",a,b)
printf("\n By Bisection Method")
printf("\n x = (a+b)/2 \n")
printf("\n Iteration    a       b       x       f(x)")

// Iteration calculations start
for i = 1:50
   x = (a+b)/2
   printf("\n %d          %.4f   %.4f  %.4f  %.4f",i,a,b,x,f(x))
   if f(x) > 0 then
      a = a
      b = x
   elseif f(x) < 0 then
      a = x
      b = b
   else
      a = x
```

```
    b = x
  end

  if abs(f(x)) < 0.0001 then
    break
  end
end
printf("\n")
printf("\n Root of the equation x^3-9*x- 1 is %.4f", x)
```

Practical 2

Write a program in Scilab to find the roots of the equation $x^2e^{-x}=1$ using

Secant method correct upto 4 decimal places.

Code :

```
//Find the root of the equation x^3 + x^2 + x -1
// using Secants method
clc
clear
// Define funtion or equation
function y=f(x)
  y = (x^2*exp(-x))-1
endfunction

for i = -5:5
  if f(i) > 0 & f(i+1) < 0 then
    a = i
    b = i+1
  end
end

disp("By Secants Method")
printf("\n x = b - ((b-a)/(f(b)-f(a)))*f(b)")
printf("\n Iteration  a  b  f(a)  f(b)  x  f(x)")

for i = 1 : 50
  x = b - ((b-a)/(f(b)-f(a)))*f(b)
  printf("\n %d  %.4f  %.4f  %.4f  %.4f  %.4f  %.4f",i,a,b,f(a),f(b),x,f(x))
  a = b
  b = x
```

```
   if abs(f(x)) < 0.0001 then
      break
   end
end

printf("\n Root of the equation x^3 - 4x - 9 is %.4f",x)
```

Practical – 3

Write a program in scilab to find the roots of the equation x3-x-1=0 using Regula-Falsi method correct upto 4 decimal places.

Code

```
clc
clear
// Define funtion or equation
function y=f(x)
   y = x^3 - x - 1
endfunction

for i = -5:5
   if f(i) < 0 & f(i+1) > 0 then
      a = i
      b = i+1
   end
end
disp("By False Position Method")
disp("x = (a*f(b)-b*f(a))/(f(b)-f(a))")
printf("\n Iteration a  b  f(a)  f(b)  x  f(x)")
for i = 1:50
   x = (a*f(b)-b*f(a))/(f(b)-f(a))
   printf("\n %d %.4f %.4f %.4f %.4f %.4f %.4f",i,a,b,f(a),f(b),x,f(x))
   if f(x) < 0 then
      a = x
      b = b
   elseif f(x) > 0 then
      a = a
      b= x
   else
```

```
    a = x
    b = x
  end
  if abs(f(x)) < 0.0001 then
    break
  end
end
printf("\n\n Root of the equation x^3 - x - 1 is %.4f",x)
```

Practical – 4

Write a program in Scilab to find the roots of the equation

$x^2 + 3x – 5 = 0$ using Newton Raphson method correct upto 4 decimal places.

Code :

```
clc
clear
// Define funtion or equation
function y=f(x)
  y = x^2 + 3*x - 5
endfunction

function dy=df(x)
  dy = 2*x+3
endfunction

for i = 0:5
  if f(i) < 0 & f(i+1) > 0 then
    a = i
    b = i +1
  end
end

disp(" By using Newton Raphson method")
printf("\n  x = x0 - f(x0)/df(x0)")
printf("\n Iteration   x  f(x)")
x0 = a
for i = 1 : 50
  x = x0 - f(x0)/df(x0)
  printf("\n %d  %.4f   %.4f",i,x,f(x))
```

```
   x0 = x

   if abs(f(x)) < 0.0001 then
      break
   end
end
printf("\n Root of the equation x^3 - 4*x - 9 by Newton raphson method is
%.4f",x)
```

Practical – 5

Write a Scilab program for numerical integration of y=x^3 in the

limits 0 to 1 using trapezoidal rule

code :

```
clc
clear
function y=f(x)
   y = x^3
endfunction

a = 1
b = 2
h = 0.1
x = a:h:b
n = (b-a)/h

printf("\n  x     y")
for i = 1 : n+1
   printf("\n %.4f    %.4f",x(i),f(x(i)))
end

sum = 0
for i = 1:n+1
   if i == 1 | i == n then
      sum = sum + f(x(i))
   else
      sum = sum + 2 * f(x(i))
   end
end
```

```
sum = h/2*sum
printf("\n\n By Trapezoidal Rule")
printf("\n Integration of f(x) is = %0.4f \n",sum)
```

Practical – 6

Write a Scilab program for numerical integration of 4+2sinx in the limits 0 to π using Simpson's 3/8th rule.

Code

```
clc
clear
function y=f(x)
  y = 4 + 2*sin(x)
endfunction

a = 0
b = %pi
h = %pi/6
x = a:h:b
n = (b-a)/h

printf("\n  x    y")
for i = 1 : n+1
  printf("\n %.4f   %.4f",x(i),f(x(i)))
end
sum = 0
for i = 1:n+1
  if i == 1 | i == n then
    sum = sum + f(x(i))
  elseif modulo(i+1,3) == 0
    sum = sum + 2 * f(x(i))
  else
    sum = sum + 3 * f(x(i))
  end
end
sum = 3*h/8*sum
printf("\n\n By Simpson 3/8 Rule")
printf("\n Integration of 4 + 2*sin(x) in limit 0 to pi is = %0.4f \n",sum)
```

Practical – 7

Write a program in Scilab for numerical integration of 3x2 +2x in the limits 1 to 3 using Simpson's 1/3rd rule.

Code :

```
clc
clear

function y=f(x)
   y = 3*x^2 + 2*x
endfunction

a = 1
b = 3
h = 0.2
x = a:h:b
n = (b-a)/h

printf("\n  x     y")
for i = 1 : n+1
   printf("\n %.4f    %.4f",x(i),f(x(i)))
end
sum = 0
for i = 1:n+1
   if i == 1 | i == n then
      sum = sum + f(x(i))
   elseif modulo(i+1,2) == 0
      sum = sum + 2 * f(x(i))
   else
      sum = sum + 4 * f(x(i))
   end
end
sum = h/3*sum
printf("\n By Simpson 1/3 Rule")
printf("\n Integration of 3*x^2 + 2*x with limit 1 and 3 is = %0.4f \n",sum)
```

Practical – 8

Write a Scilab program to find the value of y(0.1) and y(0.3) where dy=(1-y)dx , y(0)=0 using Euler's method.

CODE :

```
clc
clear

function dy=f(x, y)
   dy = 1-y
endfunction

x0 = 0
y0 = 0
h = 0.1
printf("\n By Eulers Method")
printf("\n y1 = y0 + h*f(x0,y0)")

for i = 1 : 3
   x1 = x0 + h
   y1 = y0 + h*f(x0,y0)
   printf("\n f(%.1f) = %.4f",x1,y1)
   x0 = x1
   y0 = y1
end
```

Practical – 9

Write a Scilab program to find the value of y(0.3) where

dy = (y – 2x)dx , y(0)=1 using Modified Euler's method.

Code :

```
clc
clear

function dy=f(x, y)
   dy = y - 2*x
endfunction

x0 = 0
y0 = 1
h = 0.3
printf("\n By Eulers Method")
printf("\n y1 = y0 + h*f(x0,y0)")
x1 = x0 + h
```

```
y1 = y0 + h*f(x0,y0)
printf("\n f(%.2f) = %.4f",x1,y1)
printf("\n By Eulers Modification Method")
printf("\n y1(n+1) = y0 + h/2*(f(x0,y0) + f(x1,y1n))")
y10 = y1
y11 = y0 + h/2*(f(x0,y0) + f(x1,y10))
printf("\n y11 = %.4f",y11)
y12 = y0 + h/2*(f(x0,y0) + f(x1,y11))
printf("\n y12 = %.4f",y12)
y13 = y0 + h/2*(f(x0,y0) + f(x1,y12))
printf("\n y13 = %.4f",y13)
y14 = y0 + h/2*(f(x0,y0) + f(x1,y13))
printf("\n y14 = %.4f",y14)
y15 = y0 + h/2*(f(x0,y0) + f(x1,y14))
printf("\n f(%.2f) = %.4f",x1,y15)
```

Practical 10

Write a Scilab program to solve differential equation x+y at x=0.2 using

Range-Kutta 4th order method. Y (0) =1; Take step size=0.2.

Code :

```
clc
clear

function dy=f(x, y)
   dy = x + y
endfunction
x0 = 0
y0 = 1
h = 0.2
printf("\n By Runge-Kutta 4th order")
x1 = x0 + h
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
printf("\n f(%.1f) = %.4f",x1,y1)
```

Practical – 11

Write a Scilab program for Linear regression to fit a straight line for the following
values:
X 1 2 3 4 5 6 7
Y 0.5 2.5 2.0 4.0 3.5 6.0 5.5

Code

```
clc
clear
x = [1 2 3 4 5 6 7]
y = [0.5 2.5 2 4 3.5 6 5.5]
n = length(x)
R = [sum(y); sum(x.*y)]
L = [n sum(x); sum(x) sum(x.^2)]
A = L\R
a = A(1)
b = A(2)
printf("\n Equation of regression for the following data is")
printf("\n y = %.4f + %.4f * x",a,b)
```

Practical – 12

Write a Scilab program to find the value of Second degree regression coefficient from
following data:
X 1 2 3 4 5 6 7 8
Y 3 7 10 12 14 17 20 24

Code :

```
clc
clear
x = [1 2 3 4 5 6 7 8]
y = [3 7 10 12 14 17 20 24]
N = length(x)
L = [N sum(x) sum(x.^2); sum(x) sum(x.^2) sum(x.^3) ;  sum(x.^2) sum(x.^3) sum(x.^4) ]
R = [sum(y) ; sum(x.*y) ; sum((x.^2).*y)]
A = L\R
printf("\n Equation of regression of Second degree parabola is :")
printf("\n y = %.4f + %.4f x + %.4f x^2",A(1),A(2),A(3))
```

Practical 13

A senior citizen receives on an average 2.5 telephone calls during his afternoon nap period between 14:00-14:05. Write a Scilab code to find the probability that on a certain day he receives a) No telephone calls b) Exactly 4 calls during the same period (Poisson distribution)

Code :

```
clc
clear

m = 2.5
function prob=pois(m, k)
   prob = exp(-m)*m^k/factorial(k)
endfunction

printf("\n By Poissons Distribution")
printf("\n P(X=x) = e^-m*m^x/x! ")
printf("No telephone calls")
k = 0
prob1 = pois(m,k)
printf("\n Probability of getting No telephone calls")
printf("\n P(X=0) = %.4f",prob1)
printf("\n Exactly 4 calls during the same period")
k = 4
prob1 = pois(m,k)
printf("\n Probability of getting Exactly 4 calls during the same period")
printf("\n P(X=4) = %.4f",prob1)
```

Practical – 14

In a factory, the probability that a maker is skilled is 0.4. Write a Scilab code to find the probability that out of five workers

1) None is skilled

2) Exactly one is skilled

3) At least one is skilled (Binomial distribution)

Code :

```
clc
clear
function comb=C(n, r)
   comb = factorial(n)/(factorial(r)*factorial(n-r))
endfunction
n = 5
p = 0.4
q = 1-p
printf("\n By Binomial Distribution")
printf("\n P(X=x) = nCx * p^x * q^(n-x)")
printf("\n probability that out of five worker None is skilled")
x=0
prob = C(n,x) * p^x * q^(n-x)
printf("\n P(X = 0) = %.4f",prob)
printf("\n probability that out of five worker Exactly one is skilled")
x=1
prob = C(n,x) * p^x * q^(n-x)
printf("\n P(X = 1) = %.4f",prob)
printf("\n probability that out of five worker At least one is skilled")
x=0
prob = 1 -(C(n,x) * p^x * q^(n-x))
printf("\n P(X >= 1) = %.4f",prob)
printf("\n Second Method")
printf("\n probability that out of five worker At least one is skilled")
x = 1
prob_sum = 0
for i = x:n
   prob = C(n,i) * p^i * q^(n-i)
   prob_sum = prob_sum + prob
end
printf("\n P(X >= 1) = %.4f",prob_sum)
```