**MATHLAB(MSc 2017-18)**

**Govt. Saadat College**

**Part-A: Mathematica**

**01.(a) Integration using Simpson 3/8 rules**

**--------------------------------------------------------------------------**

**f[x\_]:=Exp[x\*x];**

**a=0;**

**b=1;**

**ev=N[Integrate[f[x],{x,a,b}]];**

**n=12;**

**h=(b-a)/n;**

**s=f[a]+f[b];**

**For[i=1,i<n,i++,If[Mod[i,3]==0,s=s+2\*f[a+i\*h],s=s+3\*f[a+i\*h]]];**

**mh=N[(3\*h/8)\*s];**

**Print["---------------- Result ----------------"];**

**Print["Actual Value = ",ev]**

**Print["Using Simpson's 3/8 =",mh]**

**Print["Error = ",ev-mh]**

1. **(b) Integration using Weddles’s rules**

----------------------------------------------------------------------------------------------------------------------

**f[x\_]:= 1/(1+x\*x);**

**a = 0;**

**b = 6;**

**n = 12;**

**h = (b-a)/n;**

**ev = N[Integrate[f[x],{x,a,b}]];**

**y = {f[0]};**

**For[i=1,i< 13, i++ , AppendTo[y,f[a+i\*h]]]**

**w = (3\*h/10)\*( (Part[y,1]+Part[y,13])**

**+5\*( Part[y,2]**

**+Part[y,6]**

**+Part[y,8]**

**+Part[y,12])**

**+ 2\*Part[y,7]**

**+(Part[y,3]**

**+ Part[y,5]**

**+Part[y,9]**

**+Part[y,11])**

**+ 6\*(Part[y,4]**

**+Part[y,10]));**

**Print["-------- Output ----------"]**

**Print["Actual Value = ",ev];**

**Print["Using Weddles's Rule = ", N[w]];**

**Print["Error = ", ev-w]**

**02.(i) Solving Initial Value problems.**

**----------------------------------------------------------------**

**----------------------------------------------------------------**

**02.(ii)**

**----------------------------------------------------------------**

**----------------------------------------------------------------**

**02.(iii)**

**---------------------------------------------------------------**

**f=(x\*x-y[x]\*y[x]) -x\*y[x] \*y'[x]**

**DSolve[{f==0, y[1]==1},y[x],x]**

**---------------------------------------------------------------**

**03.(a) Solving System of linear equations.(Gauss Jacobi)**

**---------------------------------------------------------------------**

**Print["Using Gauss Jacobi"];**

**xx = (85-6\*y+z)/27;**

**yy = (72-6\*x-28z)/15;**

**zz = (110-x-y)/54;**

**xs = 0.001;**

**ys = 0.002;**

**zs = 0.003;**

**For[i=1,i<16,i++,{**

**aa = (85-6\*ys+zs)/27;**

**bb = (72-6\*xs-2\*zs)/15;**

**cc = (110-xs-ys)/54;**

**Print[i," "," ",N[xs]," ",N[ys]," ",N[zs]] ;**

**xs = aa;**

**ys = bb;**

**zs = cc;**

**}];**

**Print["Actual Result"];**

**Solve[{27\*x+6\*y-z==85, 6\*x+15\*y+2\*z==72,x+y+54\*z==110},{x,y,z}] //N**

**--------------------------------------------------------------------**

**3.(b) Solve system of linear equation using matrix**

**---------------------------------------------------------------------**

**Print["Actual Solution"];**

**Solve[{x+2y+3z ==-4, 2x+4y+5z==-7,3x+5y+6z==-10},{x,y,z}]**

**A ={{1,2,3},{2,4,5},{3,5,6}};**

**B = {-4,-7,-10} ;**

**Print["Solution using Matrix Inversion"];**

**X = Inverse[A].B**

**---------------------------------------------------------------------**

**3.(c) Solve System of Linear equation using Gauss Jordan Method**

**-------------------------------------------------------------------**

**Print["Actual Solution"];**

**Solve[{x+2y+z ==8, 2x+3y+4z==20,4x+3y+2z==16},{x,y,z}]**

**A = {{1,2,1,8},{2,3,4,20},{4,3,2,16}};**

**Print["Reduced Matrix"];**

**RowReduce[A] //MatrixForm**

**Print["Result using Jordan Elimination method"];**

**RowReduce[A][[All,4]] // MatrixForm**

**04.(a) Complex Analysis – Testing Analytic function**

**----------------------------------------------------------------**

**u[x\_,y\_]=x\*x\*x+x\*y\*y;**

**v[x\_,y\_]=y\*y\*y+x\*x\*y;**

**ux=D[u[x,y],x];**

**uy=D[u[x,y],y];**

**vx=D[v[x,y],x];**

**vy=D[v[x,y],y];**

**If[ux===vy&&vx===-uy,Print[u[x,y]+Iv[x,y],"is Analytic"],Print["w=",u[x,y],"+",I\*v[x,y],"is not Analytic"]];**

**--------------------------------------------------------------------------**

**04.(b) Harmonic Function Testing**

**--------------------------------------------------------------------**

**f[x\_,y\_]= (1/2)\*Log[x\*x+y\*y];**

**fx = D[f[x,y],x];**

**fxx= D[fx,x];**

**fy = D[f[x,y],y];**

**fyy =D[fy,y];**

**If[Simplify[fxx+fyy] === 0 , Print["Harmonic"],Print["Not Harmonic"]];**

**fx = D[f[x,y],x] /. {x-> z , y-> 0};**

**fy = D[f[x,y],y] /. {x-> z , y-> 0};**

**m = Integrate[fx-I\*fy,z];**

**v = Im[m /. z-> x+I\*y] //ComplexExpand**

**--------------------------------------------------------------------------**

**5.(a) Application of newton’s Law of Cooling**

**-------------------------------------------------------------------**

**diff = (T-20);**

**k =- Integrate[1/diff, {T,70,40}]/Integrate[1,{t,0,3}];**

**E1 = Integrate[1/diff,{T,70,T}];**

**E2 = -k\*Integrate[1,{t,0,6}];**

**s= Solve[E1 == E2 , T];**

**Print["Required temperature = ", s[[All,1,2]] ," Fahrenheit "];**

**-------------------------------------------------------------------**

**5.(b) Solving ODE with NDSolve and DSolve**

**-------------------------------------------------------------------**

**E1 = DSolve[{y'[x]==1+0.5\*y[x]\*y[x], y[0]==1},y[x],x];**

**s1[x\_] = E1[[1,1,2]];**

**E2 = NDSolve[{y'[x]==1+0.5\*y[x]\*y[x], y[0]==1},y[x],{x,0,1}];**

**s2[x\_]= E2[[1,1,2]];**

**data = Table[{x,s1[x],s2[x]},{x,0,1,.1}];**

**TableForm[data,TableHeadings->{None,{"x","Analytic","Numerical"}}]**

**Plot[{s1[x],s2[x]},{x,0,1}]**

**-------------------------------------------------------------------------**

**Part-B Fortran**

**08. Find real root of the equation using Bisection method**

**Solution:**

**----------------------------------------------------------------**

**f(x) = x\*x\*x – 2.0\*x -5.0**

**print\*,"Enter the value of a and b : "**

**10 read(5,\*) a,b**

**f1 = f(a)**

**r = f(b)**

**if( r\*f1 .GE. 0.0) goto 10**

**print\*," N A B"**

**n = 1**

**q = 0.1e-4**

**20 c = (a+b)/2**

**g = f(c)**

**if( g .EQ. 0.0) then**

**b = c**

**else**

**a = c**

**f1 = g**

**end if**

**if(abs(b-a) .LT. q) goto 35**

**write(6,30) n,a,b**

**30 format(3x , i2 , 2x , 2f15.6)**

**n = n+1**

**goto 20**

**35 c=(a+b)/2**

**write(6,50) c**

**50 format(3x, "The root is x =" , f15.6)**

**end**

**----------------------------------------------------------------Sample Input: 2 3**

**----------------------------------------------------------------**

**10. Find the value of using Trapezoidal rules**

**Solution:**

**---------------------------------------------------------------**

**integer i**

**real h, sum , x , f , a , b**

**print\*,"Enter the value of a and b"**

**read\*, a , b**

**n = 60**

**h = (b-a)/real(n)**

**sum = 0.5\*(f(a)+f(b))**

**do i = 1 , n-1**

**x = a+i\*h**

**sum = sum + f(x)**

**enddo**

**sum = h\*sum**

**print\*,"Value of the Integration = ", sum**

**end**

**function f(x)**

**f = exp(x/2.0)**

**return**

**end function**

**----------------------------------------------------------------Sample Input: 1 2**

**----------------------------------------------------------------**

**12. Integration using simpson 3/8 rules**

**Solution:**

**---------------------------------------------------------------**

**real function f(x)**

**real x**

**f = 1.0 - exp(-x/2.0)**

**return**

**end function**

**real y(7)**

**real a,b,sum,n,init**

**print\*,"Enter lower and upper limit: "**

**read\*,a,b**

**init = a**

**n = 6.0**

**h = (b-a)/n**

**do i=1,7**

**y(i) = f(init)**

**init = init + h**

**enddo**

**sum = h/3.0\*((y(1)+y(7)) + 4\*(y(2)+y(4)+y(6)) + 2\*(y(3)+y(5)))**

**print\*,"Value of the Integration = ",sum**

**end**

**----------------------------------------------------------------Sample Input: 1 2**

**----------------------------------------------------------------**

**15. Determining binomial coefficient nCr using function sub program**

**Solution:**

**---------------------------------------------------------------**

**integer n,r**

**22 print\*,"Enter the value of n and r"**

**read\*,n,r**

**if(n .LE. r) then**

**print\*,"n must be greater than r"**

**goto 22**

**endif**

**nr = n-r**

**ib = ifact(n)/(ifact(r)\*ifact(nr))**

**print\*,"value of nCr = ",ib**

**end**

**function ifact(k)**

**isum = 1;**

**do i = 1,k**

**isum = isum \* i**

**enddo**

**ifact = isum**

**return**

**end**

**----------------------------------------------------------------Sample Input: 5 2**

**----------------------------------------------------------------**

**16. Matrix Multiplication C=AB Where order of A = 3x4 and B = 4x5**

**Solution:**

**---------------------------------------------------------------**

**integer p**

**parameter(m=3,n=4,p=5)**

**dimension a(m,n), b(n,p) , c(m,p)**

**print\*,"Enter the Matrix A: "**

**read\*, ((a(i,j), j=1,n), i=1,m)**

**print\*,"Enter the Matrix B: "**

**read\*, ((b(i,j), j=1,p), i=1,n)**

**do i=1,m**

**do j=1,p**

**sum = 0.0**

**do k=1,n**

**sum = sum + a(i,k)\*b(k,j)**

**c(i,j) = sum**

**enddo**

**enddo**

**enddo**

**print\*," Product of A and B Matrix"**

**print 30, ((c(i,j),j=1,p),i=1,m)**

**30 format(2x3(2x,F8.2))**

**end**

**----------------------------------------------------------------Sample Input: Matrix A: 1 2 3 2 1 3 5 3 1 4 2 3**

**Matrix B: 2 1 3 5 8 6 5 4 2 1 4 5 6 7 3 4 5 6 3 4**

**----------------------------------------------------------------**

**END**