CHAPTER V

CHAPTER - V

COMPUTER IPLEMENTATION OF SOME METHODS :

5.1 INTODUCTION :

The advent of computers has changed the complete out look in many fields of engineering and science. Today many reaserch peoples active in applied mathematics, physics and engineering are finding use of large-scale digital computers in the solution of their problems. Also now a days computers are used widely to solve the equations of electrical networks, bending of beems, stability of aircraft, vibration theory and others.

5.2 PROGRAMS :

In this chapter computer programs of following methods of solving initial value differential equations are given in Pascal.

(1) R-K second order method, (2) R-K classical fourth order method, (3) Thired ordee Adams-Bashforth method (4) Second order differential equation using R - K method (5) Adams - Bashforth-Moulton predictor method.

Alweys R-K methods are generally considered to be the most efficient one-step methods. If accuracy requirement is low, then low order methods are used. But for high accuracy high order methods are used. In variable order R-K scheme required larger storage and their higher overheads than fixed order schemes. But they are worthy of consideration.

The Adams-Moulton method has given better results, but it is not as accurate as the fourth-order R-K method. For a given order R-K methods are slightly more accurate than predictor-corrector methods. But, in R-K method requires more function evalutions as the order is icreased, a P-C method needs a fixed number of function evalutions which is independent of the order.

If the R.H.S. of differential equation is independent of transcendental functions, then the accuacy of classical R-K is sufficiently greater to make it more efficient than the Adams method. The Adams method requires only two evaluations of the right hand side of the differential equation containing transcendental functions such as exponential and trignometric functions which are relatively costly to evaluate.

5.2 Programs:

```
{Program-1 for solving initial  value  ordinary  differential
equation using second order R-K method.X1,Y1: Initial values,
x[n+1]:Final value,h:Step size}
(*Program of second order R-K method*)
PROGRAM RUNGE2;
uses crt;
     VAR
       i,n: integer;
         x: array [1..20] of real;
         y: array [1..20] of real;
         h,k1,k2: real;
     FUNCTION f(x,y:real):real;
     BEGIN
         f:= x+y;
     End;
 BEGIN
     clrscr;
     writeln('Enter number of intervals');
     readln(n);
     writeln('Enter first and  last  data  points  of  entire
intervals');
     readln(x[1],x[n+1]);
     h:= (x[n+1]-x[1])/n; (*to calculate stepsize*)
     writeln{'Stepsize:=     ',h:12:8};
     writeln;
     For i:= 2 to n do
       x[i]:= x[i-1] + h;
    writeln('Give initial value of y[1] at x[1]');
```

```
    readln(y[1]);
    For i:= 2 to (n+1) do
    BEGIN
      k1:= h*f(x[i-1],y[i-1]);
      k2:= h*f(x[i-1]+h,y[i-1]+k1);
      y[i]:= y[i-1]+0.5*(k1+k2);
    End;
    writeln('      x[i]                    y[i]');
    For i:= 1 to (n+1) do
    BEGIN
        writeln(x[i]:12:8,'     ',y[i]);
    End;
    readln;
End.
```

```
{Program-2 for solving  initial  value  ordinary  differential
equation using Fourth order CLASICAL  R-K  method.X[1],X[N+1]
initial & final points,h:step size.}
PROGRAM RK4;
Uses crt;
      VAR
        i,n: integer;
          x: array[1..20] of real;
          y: array[1..20] of real;
          h,k1,k2,k3,k4: real;
     FUNCTION f(x,y:real):real;
     Begin
       f:= x+y;
     End;
 Begin
     clrscr;
     writeln('Enter the number of intervals');
     readln(n);
     writeln('Enter the first and last data points of  entire
intervals');
     readln(x[1],x[n+1]);
     h:= (x[n+1]-x[1])/n;{To calculate stepsize}
     writeln('Stepsize:=   ',h:12:6);
     writeln;
     For i:= 2 to n do
       x[i]:= x[i-1] + h;
       writeln('Give initial value of y[1] at x[1]');
       readln (y[1]);
```

```
 For i:= 2 to (n+1) do
 Begin
    k1:= h*f(x[i-1],y[i-1]);
    k2:= h*f(x[i-1]+0.5*h,y[i-1]+0.5*k1);
    k3:= h*f(x[i-1]+0.5*h,y[i-1]+0.5*k2);
    k4:= h*f(x[i-1]+h,y[i-1]+k3);
    y[i]:= y[i-1]+(1/6)*(k1+2*K2+2*K3+K4)
End;
writeln('      x[i]        y[i]');
For i:= 1 to (n+1) do
Begin
   writeln(x[i]:12:6,'     ',y[i]:12:6);
End;
readln;
End.
```

(Program-3 using third order Adams-Bashforth method for solving initial value differential eqution.x[1], x[n+1] are initial points of interval The initial values are y1,y2,y3.}

```
PROGRAM ADAMS3;
uses crt;
    VAR
      i,n: integer;
        x: array [1..20] of real;
        y: array [1..20] of real;
        k1,k2, k3,h: real;
    FUNCTION f(x,y:real):real;
    BEGIN
       f:=-y*y; (*Exact solution:= 1/(x+1)*)
       End;
BEGIN
    clrscr;
    writeln('Enter number of intervls');
    readln(n);
    writeln('Enter first and  last  data  points  of  entire
intervals');
    readln(x[1],x[n+1]);
    h:= (x[n+1]-x[1])/n;
    writeln('Stepsize:=      ',h:12:6);
    writeln;
    For i:=2 to n do
      x[i]:= x[i-1] + h;
    writeln('Enter the initial values y1,y2,y3');
    read(y[1],y[2],y[3]);
```

```
For i:= 4 to (n+1) do
BEGIN
k1:= f(x[i-1],y[i-1]);
k2:= f(x[i-2],y[i-2]);
k3:= f(x[i-3],y[i-3]);
y[i]:= y[i-1]+(23*k1-16*k2+5*k3)*h/12
End;
writeln('     x[i]      Y[i]');
For i:= 1 to (n+1) do
Begin
writeln(x[i]:12:6,'     ',y[i]:12:6);
End;
readln;
End.
```

```
(Program-4 for solving initial value ordinery differential
equations using Adam-Bashforth-Moulton-P-C-method. ehere
first four values are obtained from R-K methods or exact
solution.t1,t[n+1] are first & last data points, y1,y2,y3,y4
are initial values.}
Program Abmpcr;
uses crt;
   var
    i,n,k:integer;
    t,y:array[1..100] of real;
    p,h,h2,k1,k2,k3,k4,k5:real;
  Function f(t,y:real):real;
    begin
        f:=t+y;
    end;
Begin
   clrscr;
   writeln('Enter number of intervals');
   readln(n);
   writeln('Enter first and last data points of entier
intervals');
   readln(t[1],t[n+1]);
   h:=(t[n+1]-t[1])/n;
   writeln('Stepsize=    ',h:12:7);
   writeln;
   For i:= 2 to n do
      t[i]:=t[i-1]+h;
      writeln('Enter the initial values y1,y2,y3,y4');
```

```
(Program-5 for higher order differential equations using  R-K
method.here we used Adaptive method.  x[1],  x[n+1]  are  end
points of interval.dydx =  dy/dx.y[1]  &  dydx[1]are  initial
values.}
Program RK2;
uses crt ;
     VAR
       i,n: integer ;
       x : array [1..20] of real;
       y : array [1..20] of real;
       dydx : array [1..20] of real;
       h, k1, k2 : real;
 FUNCTION f(x, y: real): real;
 Begin
     f:= (1+x*x)*y;
End;
 BEGIN
 clrscr;
 writeln ( 'ENTER NUMBER OF INTERVALS ');
 readln (n);
 writeln  ('Enter  first  and  last  data  points  of  entire
intervals ');
 readln (x[1],x[n+1]);
 h:=(x[n+1]-x[1])/n;(*To calculate stepsize*)
 writeln('Stepsize:=       ',h:12:4);      writeln;
 For i:=2 to n do
   x[i]:=x[i-1] + h;
 writeln('Give initial value of y[1],dydx[1]');
```

```
    read (y[1],y[2],y[3],y[4]);
    h2:=h/24;
  For i:= 5 to n do
  Begin
   k1:=f(t[i-1],y[i-1]);
   k2:=f(t[i-2],y[i-2]);
   k3:=f(t[i-3],y[i-3]);
   k4:=f(t[i-4],y[i-4]);
   p:=y[i]+h2*(-9*k1+37*k2-59*k3+55*k4);
   t[i+1]:=t[1]+h*(i+1);
   k5:=f(t[i+1],p);
   y[i+1]:=y[i]+h2*(k2-5*k3+19*k4+9*k5);
   k1:=k2;
   k2:=k3;
   k3:=k4;
   k4:=f(t[i+1],y[i+1]);
   writeln('i':4,'t[i]':14,'y[i]':10)
  end;
  For k:=1 to (n+1) do
   writeln('    ':4,t[k]:12:7,y[k]:12:7);
   readln;
end.
```

```
readln(y[1],dydx[1]);
For i:=2 to (n+1) do
Begin
   k1:= f(x[i-1],y[i-1])*h*h/2;
   k2:=
f(x[i-1]+h*2/3,y[i-1]+(2*h*dydx[i-1]*1/3+4*k1*1/9))*h*h/2;
   y[i]:= y[i-1]+h*dydx[i-1]+0.5*(k1+k2);
   dydx[i]:= dydx[i-1]+0.5*(k1+3*k2)/h
   End;
   writeln('    x[i]     y[i]');
   For i:=1 to (n+1) do
   Begin
     writeln(x[i]:12:4,'        ',Y[i]:12:4);
     End;
     readln;
 End.
```

5.3 Out-Put:

ASD4.PASsecond order R-K method The solution of differential equation y´ = x+y , y(o)=1

THE NUMBER OF INTERVALS ARE : 10

THE FIRST & LAST DATA POINTS : 0.0000000000E+00 1.0000000000E+00

STEPSIZE IS : 0.10000000

THE INITIAL VALUE Y[1] : 1.0000000000E+00

X[i]	Y[i]
0.00000000	1.0000000000E+00
0.10000000	1.1100000000E+00
0.20000000	1.2420500000E+00
0.30000000	1.3984652500E+00
0.40000000	1.5818041012E+00
0.50000000	1.7948935319E+00
0.60000000	2.0408573527E+00
0.70000000	2.3231473748E+00
0.80000000	2.6455778491E+00
0.90000000	3.0123635233E+00
1.00000000	3.4281616932E+00

ASD3.PASThe out put of the solution of diff. Eq.y´=x+y, y(0)=1 usingfourth order classical R-K method.

THE NUMBER OF INTERVALS ARE : 10

THE FIRST & LAST DATA POINTS ARE : 0.0000000000E+00 1.0000000000E+00

STEPSIZE IS : 0.100000

THE INITIAL VALUE Y[1] IS : 1.0000000000E+00

x[i]	y[i]
0.000000	1.000000
0.100000	1.110342
0.200000	1.242805
0.300000	1.399717
0.400000	1.583648
0.500000	1.797441
0.600000	2.044236
0.700000	2.327503
0.800000	2.651079
0.900000	3.019203
1.000000	3.436559

ASD1.PASthe solution using thired order Adams- Bashforth method.Eq. is y´=x+y, y(0)=0.

THE NUMBER OF INTERVALS ARE : 10

THE FIRST AND LAST DATA POINTS ARE : 0.0000000000E+00 1.0000000000E+00

STEPSIZE IS : 0.1000

THE INITIAL VALUES Y[1] , Y[2] & Y[3] ARE : 1.0000000000E+00 1.1000000000E+00 1.2000000000E+00

x[i]	y[i]
0.3000	1.3500
0.4000	1.5296
0.5000	1.7378
0.6000	1.9781
0.7000	2.2543
0.8000	2.5700
0.9000	2.9295
1.0000	3.3372

AD3.PASThe solution of diff. Eq. y´=x+y, y(0)=1, using Adoms-Bashforth_moultan-p-c formula.

THE NUMBER OF INTERVALS ARE : 10

THE FIRST & LAST DATA POINTS ARE : 0.0000000000E+00 1.0000000000E+00

STEPSIZE IS : 0.1000000

THE INITIAL VALUE OF Y1 , Y2 , Y3 & Y4 ARE : 1.0000000000E+00 1.1000000000E+00 1.2000000000E+00 1.3620000000E+00

i	T[i]	Y[i]
1	0.0000000	1.0000000
2	0.1000000	1.1000000
3	0.2000000	1.2000000
4	0.3000000	1.3620000
5	0.4000000	1.4477878
6	0.6000000	1.5512124
7	0.7000000	1.6694869
8	0.8000000	1.8116328
9	0.9000000	1.9665459
10	1.0000000	2.1473284

AASD1.PASThe out put of program of the solution of second order differenial Eq.y,=t+y, y(0)=1.

THE NUMBER OF INTERVALS ARE : 10

THE FIRST & LAST DATA POINTS : 0.0000000000E+00 1.0000000000E+00

STEPSIZE IS : 0.10000000

THE INITIAL VALUES Y[1], dydx[1] ARE : 0.0000000000E+00 1.0000000000E+00

X[i]	Y[i]
0.0000	0.0000
0.1000	0.1002
0.2000	0.2014
0.3000	0.3046
0.4000	0.4113
0.5000	0.5227
0.6000	0.6407
0.7000	0.7675
0.8000	0.9057
0.9000	1.0589
1.0000	1.2313

The analytical solution of differential equation y´ = x+y is y = 2*exp(x)-x-1.the calculated values of y at x=0.1 to x=1 with h = 0.1 are

x	y(x)
0.0	1.0000000
0.1	1.1103418
0.2	1.1428055
0.3	1.3997176
0.4	1.5836494
0.5	1.7974425
0.6	2.0442376
0.7	2.3275054
0.8	2.6510818
0.9	3.0192062
1.0	4.3365637

5.4 CONCLUSION

We discussed in above the differential equations y´=t+y, with initial condition y(0) = 1. From above programs we conclude that Fourth-oredr classical R-K method is the best one for solving ordinary differential equations. From programs (.) and (.) it is clear that the higher order methods obtained better accuracy for the same computation effert and that the gain in accuracy for the additional effert tends to diminish after a point. Thus higher order R-K techniques are always the methods of preference.

If the range of integration of problem is very large to involve large number of steps , then it may be necessary and appropriate to use more accurate technique such as classical Fourth order R-K method and Fourth order Adams method. In these cases , it´s useful to estimate the truncation error for each step as a guide to selecting the best step size. If the truncation error are extremely small , it may be wise to increase the step size to save computer time. The R-K methods are simple to program and convenient to use but may be less efficient than multistep methods and R-K methods are used to find initial values when the multistep methods are used.